

Applying Direct Preference Optimization to Improve GPT-2 Sonnet Generation

Stanford CS224N Default Project

Aadhav Prabu

Department of Computer Science
Stanford University
aprabu2@stanford.edu

Abstract

With language models being trained on a broad range of knowledge and skills, it is difficult to achieve precise control of their behavior. Direct Preference Optimization (DPO) offers a stable, efficient, and computationally lightweight approach for enhancing the quality of LM generations, outperforming more complex methods such as Reinforcement Learning from Human Feedback (RLHF). The goal of this research is to investigate whether implementing DPO on a baseline GPT-2 model, specifically for the sonnet generation task, significantly improves its alignment with human-preferred responses compared to standard supervised fine-tuning. This research follows the DPO methodology outlined in the original paper, adapting it to our specific use case. To construct effective negative samples for our DPO datasets, we explored two approaches for sample creation, leveraging the 131 provided training samples. The first approach employed rule-based corruption strategies, including artificially lengthening or shortening sonnets, swapping quatrains between different sonnets, and introducing sentence-level word swaps. The second approach generated negative samples using outputs from a baseline SFT model, which attained an average CHRF score of 41.208. Our results indicate that the rule-based corruption method outperformed the baseline SFT-generated negatives, achieving a CHRF score of 41.863 on the test set, demonstrating the effectiveness of incorporating preference-based optimization. Ultimately, enabling language models to adapt dynamically to context and individual user preferences is essential for creating systems that are both safer and more responsive to real-world needs and applications.

1 Key Information to include

- Mentor: Sarah Chen
- External Collaborators (if you have any): N/A
- Sharing project:N/A
- Late Days (if you are using any):

Team member	Late days remaining
Aadhav Prabu	1
Total pooled	1
Team size	1
Late days applied for the report	1

2 Introduction

Large language models (LLMs) are trained on a wide variety of tasks, gaining linguistic competence through next-token prediction. However, this training objective for models is indifferent to what humans actually want from a model’s outputs [1]. Bridging this gap between statistical accuracy and genuine alignment with human preferences is one of the central challenges in modern NLP. The alignment problem is important for both practical and scientific reasons. LLMs that generate fluent but unreliable, biased, or stylistically incoherent outputs are limited in real-world deployment, particularly in fields such as creative writing [2]. Scientifically, the question of how to encode human preferences into a generative model’s parameters without access to a ground-truth reward signal remains an active area of research.

The standard approach to human alignment relies on RLHF, which requires fitting a separate reward function on human-ranked outputs and then optimizing the language model against it using reinforcement learning (RL) [3]. This pipeline is computationally intensive, introduces training instability by adding an additional RL model, and requires thousands of manually annotated human preferences. Unlike RLHF, DPO uses a much simpler objective, which allows for a more stable learning process since significant hyperparameter tuning is not required, high variance in policy gradients is avoided, and computational costs are significantly reduced [4]. Despite these changes, DPO can match or exceed RLHF performance on tasks such as Reddit forum summarization [4]. Ultimately, it is a more efficient method for aligning the model with human preferences without substantial deviation.

This project addresses the alignment problem in two complementary ways. First, by replicating a minimalist GPT-2 architecture across three tasks: binary sentiment classification, paraphrase detection, and open-ended sonnet generation, and by investigating whether DPO can be used to improve the quality of generated sonnets that align with the original Shakespearean language. Specifically, we present two methods for generating the DPO dataset: manual corruption of the sonnet training set and outputs from a baseline model. By optimizing the model for human preferences, the "desired responses and behavior from its very wide knowledge [can build] AI systems that are safe, performant, and controllable" [4]. As large language models’ training data continue to grow in size and complexity, it is important to maintain and improve approaches like DPO to keep models aligned for safer systems.

3 Related Work

Aligning language model outputs with human preferences has become a central research problem as LLMs scale, motivated by the broader problem of LLM hallucination. A study by OpenAI provides evidence that the autoregressive training objective is inherently indifferent to factual accuracy, producing models that prioritize fluency over faithfulness [5]. Specifically, in creative generation tasks, hallucination takes a structurally different form, violating meter, rhyme scheme, or line count. Direct Preference Optimization provides a substantially simpler alignment method than other approaches like RLHF, reparameterizing the reward as an implicit function of the policy and reducing alignment to a single supervised training stage on preference pairs [4]. DPO has been validated primarily on models with 6B or more parameters and thousands of human-labeled pairs; its behavior in low-resource creative generation settings is not well characterized. Subsequent work has also identified important failure modes: another study documented the challenges of reward over-optimization [6]. Additionally, several studies find a consistent performance gap between offline DPO and online methods that resample rejected outputs from the evolving policy [7]; a gap that is especially pronounced when chosen and rejected samples are close in distribution.

Specifically, for DPO, its central challenge lies in constructing informative preference pairs without expensive human annotation. Rejection sampling methods score multiple model outputs using a trained reward model and treat the top and bottom-scoring samples as the chosen and rejected pairs, producing on-policy rejected samples that are close in distribution to the chosen outputs [8]. Similarly, Anthropic introduced Constitutional AI, which uses model self-critique to generate preference data without human annotation, relying on automated scoring [9]. These approaches are widely used in the DPO literature and in practice, motivated by the intuition that on-policy rejected samples reflect realistic model failure modes rather than artificial corruptions. However, a study by ServiceNow shows that the informativeness of the preference signal depends on the margin between chosen and rejected quality: when the distributional gap is too narrow, the DPO loss provides insufficient gradient to separate the distributions [10]. Given the small dataset size in our project, we plan to implement

both approaches in our DPO application, exploring the practical trade-off between preference signal quality and training stability when data is scarce.

4 Approach

4.1 Supervised Fine-Tuning

A baseline GPT-2 model was implemented according to the default project guidelines [11]. The model follows a standard decoder-only transformer architecture trained with a causal language modeling objective, used as the foundation for all three tasks of sentiment classification, paraphrase detection, and sonnet generation.

4.2 Paraphrase Detection Task

The model takes a pair of sentences or questions as input and predicts a binary paraphrase label as a “close-style” task [11]. The primary modification required to run at scale was upgrading from an L4 GPU in Google Cloud to an H100 GPU in Google Colab to accommodate the larger Quora Question Pairs dataset; all other hyperparameters were left the same, with the exception of the learning rate and batch size, which were updated to exceed the provided baseline as defined in section 5.3.

4.3 Sonnet Generation Task

The model is prompted with the first three lines of a held-out Shakespearean sonnet and trained to generate the remaining 11 lines. No modifications to the base fine-tuning procedure were required to reach the SFT baseline of ChrF 41.208. This score serves as the reference point against which all of the custom DPO configurations are compared in Section 5.4.

4.3.1 Direct Preference Optimization

DPO was implemented as proposed in its original paper, without using any external libraries for sonnet generation. Unlike RLHF, which trains a separate reward function and uses reinforcement learning to align the model, DPO uses a much simpler objective, as seen in the equation [4]. This simplification allows for a more stable learning process while achieving performance similar to or better than that of approaches like RLHF. I expect DPO to have a similar effect on the sonnet generation task in this project.

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

4.3.2 Constructing DPO Preference Pairs

A key design decision in DPO is to generate rejected samples that the model can effectively distinguish, thereby achieving better performance or a higher ChrF score in the sonnet generation task. As the quality and realism of these rejected samples directly impact the model’s ability to learn nuanced preferences, I explored multiple negative sampling strategies aiming to balance diversity and alignment with real-world model outputs. Each preference pair consisted of the prompt (the first three lines of the sonnet), the chosen (the next 11 lines of the actual sonnet), and the rejected (the next 11 lines, corrupted or model-generated) components.

4.3.3 Rule-based Corruption

The first approach employed rule-based methods to systematically deconstruct the 131 sonnets in the training set. These manual corruption strategies enabled precise control over the types of errors introduced, such as disruptions to meter, rhyme schemes, or line counts. The primary advantage of this approach lies in its interpretability. By explicitly defining the types of corruption, I ensured that the negative samples directly targeted weaknesses the model encountered in the baseline SFT model. This contrasts with model-generated negatives, which can be more arbitrary in negative outputs.

Three corruption types were implemented, with each type applied to all training sonnets, resulting in a total training set size of 393.

- **Meter corruption:** Words within a sonnet line are randomly shuffled, breaking iambic pentameter while preserving vocabulary.
- **Rhyme scheme corruption:** The ABAB rhyme pattern is converted to AABB by swapping alternating lines within each quatrain, disrupting the Shakespearean form.
- **Line-count corruption:** The sonnet length is modified by adding or removing lines ($[\pm 2, \pm 10]$ lines inclusive), violating the fixed 14-line structure.

4.3.4 Model-Generated Rejected Samples

The strategy utilizes a baseline SFT model to generate rejected samples. Similar to the evaluation phase, I provided the first three lines of each sonnet from the training dataset to the SFT model for generating an output. Each training sonnet produced three variations, resulting in a total of 393 sonnets—matching the size of the rule-based rejection sample generation method. Because this approach better imitates the SFT model’s outputs than manual corruption methods, it may yield a more informative DPO learning signal. Appendix A provides samples generated through this approach.

5 Experiments

5.1 Data

I used the datasets provided in the default project: Stanford Sentiment Treebank (SST) and the CFIMDB dataset for the sentiment classification task, the Quora Dataset for the paraphrase detection task, and the Sonnet Dataset for the sonnet generation task. The negative samples used for DPO training were generated according to the methods described in the approach section.

5.2 Evaluation method

Following the default project guidelines, I used the evaluation metrics defined in the default project outline. This includes using accuracy for the binary sentiment classification and paraphrase detection tasks, and the CHRF score for the sonnet generation task. Applying my DPO extension to the sonnet generation task, I expect it to improve upon the CHRF score of the fine-tuned GPT-2 model as it aligns more with the human reference output.

5.3 Experimental details

For the sentiment classification task, I ran my experiments on a g2-standard-4 (4 vCPUs, 16 GB Memory virtual machine (VM) in Google Cloud, configured with the NVIDIA L4 GPU. The training time for each epoch was around 5 minutes. However, this configuration was not suitable for the paraphrase detection task, as the Quora dataset was significantly larger than the sentiment classification datasets. For this task, I trained the model in Google Colab on an NVIDIA H100 GPU. Training time for this task averages around 7 minutes per training epoch with a batch size of 64. For the sonnet task, the Google Cloud VM described above was sufficient. The SFT training took an estimated 7 seconds per epoch, while DPO training took 49 seconds per epoch, each with a batch size of 8. Below are the hyperparameters that were fine-tuned for each task.

	Paraphrase	Sonnet	DPO (Rule-based)	DPO (Model Output)
Hardware	NVIDIA H100	NVIDIA L4	NVIDIA L4	NVIDIA L4
Environment	Colab	Google VM	Google VM	Google VM
GPT-2 Size	Medium	Medium	Medium	Medium
Epochs	10	15	10	3
Learning Rate	1e-5	2e-5	1e-6	1e-6
Batch Size	64	8	8	8
Temperature	-	1.2	1.2	0.9
Top p	-	0.9	0.9	0.85
DPO Sample Size	-	-	393	393
DPO β	-	-	0.3	1.0

Table 1: Model and environment configuration for all tasks

5.4 Results

My approach is evaluated across three tasks: sentiment classification on SST, paraphrase detection on QQP, and sonnet generation. For the two classification tasks, results are reported against the provided baselines. For sonnet generation, we report ChrF scores across all DPO configurations relative to the SFT baseline.

5.5 Sentiment Classification

The sentiment classification results are broadly in line with expectations. For SST, fine-tuning only the last linear layer achieves 0.485, a modest improvement over the baseline of 0.462. Fine-tuning the full model on SST slightly underperforms the baseline (0.505 vs. 0.513), which is somewhat surprising but not unexpected given SST’s small size, as full fine-tuning on limited data can risk overfitting. For CFIMDB, both approaches match or closely approach the baseline, with last-layer fine-tuning performing comparably (0.865 vs. 0.861) and full fine-tuning marginally below (0.971 vs. 0.976). Overall, these results suggest that the GPT-2 backbone already encodes sentiment-relevant features, and the marginal gap between the scores and the baselines reflects optimization.

	Baseline Dev Score	Dev Score
Last Linear Layer for SST (Sentiment)	0.462	0.485
Full Model for SST (Sentiment)	0.513	0.505
Last Linear Layer for CFIMDB (Sentiment)	0.861	0.865
Full Model for CFIMDB	0.976	0.971

Table 2: Score Comparison for Sentiment Classification Tasks

5.6 Paraphrase Detection

The paraphrase detection results exceed the provided baseline on both dev and test sets, achieving 0.892 vs. 0.882 on dev and 0.859 vs. 0.856 on test. This is slightly better than expected, given that minimal hyperparameter modifications were made beyond switching to an H100 GPU for scale. The consistent improvement across both splits suggests this performance increase is not due to overfitting to the dev set.

	Baseline Dev Score	Dev Score	Baseline Test Score	Test Score
Accuracy	0.882	0.892	0.856	0.859

Table 3: Results for the paraphrase detection task

5.7 Sonnet Generation

The sonnet generation results reveal a clear ordering across the three configurations. The SFT baseline slightly exceeds the provided dev baseline (41.816 vs. 41.414), confirming the fine-tuning procedure is sound before any preference optimization is applied. Rule-based DPO then improves further on

the dev set, reaching 42.679, a gain of approximately 1.3 ChrF points over the SFT model and the strongest result across all configurations. On the test set, it also outperformed the baseline model with a ChrF score of 41.863 compared to 41.208.

These results are both somewhat surprising and consistent with the failure modes identified in our analysis. The improvement from rule-based DPO was not expected, given the shallow nature of the corruption, as line counts and structural perturbations do not reflect the nuanced quality differences humans would actually prefer in sonnets. The degradation in model-generated DPO indicates that the model-generated samples provide insufficient signal to improve the policy and may even actively harm it. The gap between the two DPO configurations, roughly 1.3 ChrF points on dev, provides empirical evidence that preference data construction strategy matters more than theoretical motivation in low-resource settings.

	Baseline Dev Score	Dev Score	Baseline Test Score	Test Score
SFT Model - No Extension	41.414	41.816	41.208	-
Rule-Based DPO	41.414	42.679	41.208	41.863
Model-Generated DPO	41.414	41.337	41.208	-

Table 4: CHRF scores for the sonnet generation task

6 Analysis

6.1 Paraphrase Detection Task Performance

The minimalist GPT-2 architecture successfully improves upon both baselines without significant hyperparameter tuning. For paraphrase detection, the larger Quora dataset required upgrading from the L4 GPU used for sentiment classification to an H100 in Google Colab; only the learning rate was slightly increased from $1e-5$ to $2e-5$. We did not apply DPO in this scenario because the binary classification labels are inherently limited to the space of yes, no, with “human preferred” labels providing no change.

6.2 Sonnet Generation: Analyzing Various DPO Sampling Strategies

Contrary to the prevailing assumption in the DPO literature that on-policy rejected samples produce a more informative preference signal than rule-based corruptions, we find the opposite holds in our constrained setting. Rule-based corruptions produce rejected samples that are structurally distinct from chosen samples in a consistent, predictable way. While this makes the preference signal weaker, the model can learn to avoid explicitly defined malformed outputs by focusing on deeper stylistic preferences. By contrast, the model-generated rejected samples are much closer in distribution to chosen samples. This narrow margin between chosen and rejected log-probabilities means the DPO loss must push the policy farther apart, increasing the risk of reward collapse and policy destabilization. With only 393 training pairs, there is insufficient signal to reliably achieve this separation. This finding suggests that in data-scarce settings, the stability of rule-based corruptions may outweigh the theoretical benefits of using on-policy rejected samples.

Furthermore, the training metrics indicate reward collapse for the model-generated approach. After just 4 DPO training epochs, the model achieved an accuracy of 1.0 and a loss of 0.012. The KL penalty partially slowed this process, but it did not prevent the model from overfitting. In contrast, the corruption-based approach achieved an accuracy of 0.88 and a loss of 0.08 after 10 epochs. This suggests the corruption-based approach provided a stronger learning signal for the model and prevented overfitting. Training the model on a larger, more diverse training set could potentially eliminate these issues.

Several limitations constrain the interpretation of these results. Most fundamentally, 393 preference pairs are far below the scale at which DPO has been shown to work reliably, as the fixed training set of 154 sonnets imposes a hard ceiling on how many valid pairs can be constructed. Model scale is a further limitation, as the original DPO paper evaluates on models with 6B or more parameters, and GPT-2 Medium’s limited capacity constrains the extent to which preference optimization can reshape its output distribution. Due to cost and memory constraints, larger models, such as GPT-2 Large, could not be used.

7 Conclusion

This project implemented and evaluated Direct Preference Optimization (DPO) for Shakespeare sonnet generation on a fine-tuned GPT-2 model. The SFT baseline matched benchmarks for both sentiment classification and paraphrase detection with minimal hyperparameter tuning. For sonnet generation, the SFT baseline achieved a ChrF of 41.23, and the corruption-based DPO configuration improved upon this score with hyperparameter fine-tuning, providing marginal gains. The relative performance across corruption strategies follows an inverse relationship with theoretical motivation: more traditional rejection strategies cause greater training instability than artificial corruptions. Future work involves implementing methods, such as Iterative DPO [12], in which the model generates new rejected samples and retrain iteratively, potentially improving alignment with out-of-distribution samples across multiple rounds, offsetting data scarcity. Additionally, we were limited to using the GPT2-medium model during training. Applying a larger model, such as GPT2-large can help improve the training process and capture more nuanced differences between the chosen and negative samples.

References

- [1] R. Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D. Hardy, and Thomas L. Griffiths. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences of the United States of America*, 121, 2024.
- [2] Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen Ahmed. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50:1097–1179, 2023.
- [3] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, 30, 2017.
- [4] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023.
- [5] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022.
- [6] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.
- [7] Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. Unpacking DPO and PPO: Disentangling best practices for learning from preference feedback. *Advances in Neural Information Processing Systems*, 37:36602–36633, 2024.
- [8] Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kai Shum, and Tong Zhang. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023.
- [9] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022.

- [10] Pulkit Pattnaik, Rishabh Maheshwary, Kelechi Ogueji, Vikas Yadav, and Sathwik Tejaswi Madhusudhan. Curry-dpo: Enhancing alignment using curriculum learning ranked preferences. *ArXiv*, abs/2403.07230, 2024.
- [11] CS 224N Course Staff. CS 224N default final project: Build GPT-2. https://web.stanford.edu/class/cs224n/project/DFP_Instructions.pdf, 2024. Stanford University.
- [12] Richard Yuanzhe Pang, Weizhe Yuan, He He, Kyunghyun Cho, Sainbayar Sukhbaatar, and Jason E Weston. Iterative reasoning preference optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

A Appendix A

Who will believe my verse in time to come
 If it were filled with your most high deserts? Though yet,
 heaven knows, it is but as a tomb, A name, when your ashes are buried.
 Yet for your remembrance I dare say, That you were your self, and that you were buried,
 The tomb of your youth, in your love's bosom,
 In the bosom of your best virtue. To this boast I add, that from you I
 All this rest shall come, to be your memorial.

Shall I compare thee to a summer's day? Thou art more lovely and more temperate.
 Rough winds do shake the darling buds of May, And they smell more bitter than flowers do when dry,
 Than sweet thy odor to be affords, Which must be new-sown or that which grew.
 That sweet odor is sweet to those who love it, For it grows in the bosoms of thine eyes,
 And these bosoms their sweetest buds are. So it is with sweet perfumes,
 which with their smell Take root in the breasts of sweet women and men.

Devouring Time, blunt thou the lion's paws And make the Earth devour her own sweet brood;
 Pluck the keen teeth from the fierce tiger's jaws, And break his jaws open, and grind
 The living steel of the earth's living things. Let the living wood, with a single wail,
 Shall bear witness that Time did his best To carve the Earth to thy love,
 And made thee a queen of the world, Of wond'rous beauty and wealth.
 But for thee, Time's hand is no better than a gree'n knife, And Time's steel is but the dull knife's
 edge.

A woman's face with Nature's own hand painted Hast thou, the master mistress of my passion;
 A woman's gentle heart, but not acquainted With this art of love's alteration;
 A woman's sweet complexion, but not acquainted With this perfumed hue;
 A woman's complexion, but not acquainted With this complexion, which is painted in mine eye.
 To see a woman's face is to feel her in thee; This feeling is the most precious treasure of mine,
 For a woman's face to me is as a flower doth live, And, though it be dead, it lives in thee.

So is it not with me as with that muse Stirred by a painted beauty to his verse,
 Who heaven itself for ornament doth use, And with the fairest gems in his eyes doth store?
 In what wonder is it with this beauty That heaven itself doth draw so fine a line
 Of earth and sky, of air and water, of all kinds, Which brass, brass, and precious stones do all tie
 By their mutual parts to his composition? O, what wonder is it with this wonder
 When heaven itself for ornament doth use Such wondrous works of composition
 as that thou mayst see, That he that hath such works can never praise But to his muse,
 whom nature calls his own!