

GPT2 Optimization with PEFT and DPO

Stanford CS224N Default Project

Kiran Sun

Department of Computer Science
Stanford University
skxran@stanford.edu

Abstract

This project implements the core components of a Generative Pretrained Transformer (GPT-2) from scratch and fine-tunes it for downstream sentiment analysis, paraphrase detection, and sonnet generation tasks. While unsupervised language models learn significant knowledge during pre-training, adapting a model with 700m+ parameters to tasks with limited data, such as a 143-poem sonnet corpus, runs a severe risk of overfitting. To address this, we scaled our architecture to gpt2-large using Parameter-Efficient Fine-Tuning (PEFT), specifically Low-Rank Adaptation (LoRA), to adapt deep semantic representations within a strict memory budget. Furthermore, we integrated Direct Preference Optimization (DPO) to improve sonnet generation without relying on unstable reinforcement learning pipelines. With some other optimizations and hyperparameter fine-tuning, we achieved a paraphrase detection test accuracy of 0.874 and a sonnet generation chrF score of 43.552, demonstrating that lightweight preference optimization and efficient fine-tuning can effectively align large language models on highly constrained downstream tasks.

1 Key Information to include

- Mentor: Fang Wu
- External Collaborators (if you have any): n/a
- Sharing project: n/a
- Late Days: 1

2 Introduction

While unsupervised language models learn great amounts of general knowledge during pre-training, they often require substantial post-training to succeed at highly specific downstream tasks. Adapting a generalized model to evaluate semantic equivalence (paraphrase detection) or follow strict poetic structures (sonnet generation) poses a fundamental representation bottleneck. Specifically, forcing a massively over-parameterized model to map complex language features into narrow tasks using limited data, such as our 143-poem sonnet corpus, typically leads to severe overfitting.

Traditional methods for addressing these constraints often fail or prove computationally prohibitive. Full-parameter supervised fine-tuning is memory-intensive and simply teaches the model to mimic training data without necessarily learning the underlying structural rules. Meanwhile, aligning models to complex human preferences or stylistic constraints typically relies on Reinforcement Learning from Human Feedback (RLHF), a notoriously unstable and expensive process requiring multiple phases of reward modeling and policy optimization.

To overcome these limitations, our approach integrates Low-Rank Adaptation (LoRA) and Direct Preference Optimization (DPO). First, we utilize LoRA to efficiently scale our architecture to

gpt2-large by freezing the base model and injecting small trainable rank-decomposition matrices. Second, we apply DPO to enforce the strict stylistic constraints of sonnet generation. DPO replaces the unstable RLHF pipeline with a simple binary cross-entropy objective optimized on synthetic pairwise preference data. Combined with bfloat16 mixed-precision training and linear warmup scheduling, this lightweight framework provides a robust solution for aligning large language models to strict structural and semantic constraints without the costs of traditional fine-tuning.

3 Related Work

Recent advancements in natural language processing have increasingly focused on adapting massive pre-trained models to specialized domains under strict computational constraints. To address the memory bottlenecks associated with full-parameter fine-tuning, Hu et al. introduced Low-Rank Adaptation (LoRA) [1], a parameter-efficient technique that freezes pre-trained weights and injects trainable rank-decomposition matrices. While LoRA successfully scales model capacity within a limited GPU budget, it fundamentally remains a supervised fine-tuning method. As such, it is prone to simply mimicking the surface-level statistics of training data, often failing to learn complex, rigid structures on its own.

To explicitly align model behavior with strict structural requirements, such as those required by the sonnet, the field has traditionally relied on Reinforcement Learning from Human Feedback (RLHF). However, to address the instability and high computational overhead of RLHF pipelines, Rafailov et al. proposed Direct Preference Optimization (DPO) [2]. DPO reformulates reward modeling into a supervised binary cross-entropy objective, allowing models to directly learn from paired preference data without requiring a separate reward model.

Our project applies a combination of these two distinct advancements. While DPO is predominantly used for broad conversational safety, and LoRA is typically used for simple domain adaptation, our work explores their combined efficiency in a highly constrained domain. By pairing the parameter efficiency of LoRA with the structural enforcement capabilities of DPO, our approach represents a promising next step: demonstrating a lightweight, highly accessible framework for aligning large language models to rigid poetic constraints (sonnet generation) and fine-grained semantic bounds (paraphrase detection), bypassing the need for massive compute clusters or complex reinforcement learning architectures.

4 Approach

4.1 Baselines

Our baseline models and evaluation procedures follow the specifications outlined in the CS 224N Default Final Project (DFP) instructions [3]. For sentiment analysis and paraphrase detection, the baselines consist of a standard GPT-2 implementation evaluated on accuracy. For the sonnet generation task, the baseline is a standard autoregressive fine-tuning of GPT-2, evaluated via the chrF metric to measure adherence to Shakespeare’s original language distribution. Please refer to the project handout for details.

4.2 Base Architecture and Training Enhancements

Our foundational architecture relies on a decoder-only Transformer using masked multi-head self-attention. For the initial sentiment analysis and paraphrase detection tasks, we implemented the core GPT-2 layers and the step function of the Adam optimizer utilizing Decoupled Weight Decay Regularization. To handle the cloze-style paraphrase task, we reformatted the inputs to prompt the model with a binary “yes” or “no” target token, shifting from linear classification to next-token prediction.

To improve upon the baseline training pipeline, we implemented several custom optimizations. We introduced a linear warmup scheduler to stabilize the gradients during early training batches, preventing the model from forgetting what it learned about the English language. Furthermore, we conducted extensive hyperparameter tuning and integrated an early halting mechanism based on validation loss deltas to prevent overfitting on the relatively small downstream datasets.

4.3 Parameter-Efficient Fine-Tuning (LoRA)

To scale our architecture to the gpt2-large model without exceeding memory constraints, we implemented Low-Rank Adaptation (LoRA), referencing the foundational paper by Hu et al.[1]. Parameter-efficient finetuning (PEFT) techniques like LoRA reduce the trainable parameter count relative to full finetuning by freezing most or all of the base model. This enables the finetuning of significantly larger models given a strict GPU memory budget.

Specifically, we inject trainable rank-decomposition matrices into the query and value projection layers of the Transformer attention blocks (and also the key layer, for sonnet generation). In a standard linear layer, the forward pass computes $h = W_0x$. LoRA freezes the pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$ and adds a trainable low-rank update $\Delta W = BA$, where $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$, with the rank $r \ll \min(d, k)$. This yields the modified forward pass:

$$h = W_0x + BAx$$

By only updating the low-rank matrices A and B , we drastically reduce the optimizer state and memory footprint while preserving the network’s deep semantic adaptability.

4.4 Direct Preference Optimization (DPO)

To improve the autoregressive sonnet generation, we implemented Direct Preference Optimization (DPO), based on the methodology proposed by Rafailov et al.[2]. DPO is an approach to aligning language models with human preferences through pairwise comparison data. Unlike RL-based alternative methods such as Reinforcement Learning from Human Feedback (RLHF), DPO provides a more stable training paradigm that operates in a supervised manner.

Fundamentally, DPO trains models using paired examples: for each input prompt, we have both a preferred output (y_w) and a less preferred alternative (y_l). To implement this, we constructed a dual-model architecture comprising a frozen reference model (π_{ref}) and a trainable policy model (π_θ). The model learns to assign higher probability to preferred outputs by optimizing the following objective:

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

where σ is the sigmoid function and β is a temperature parameter controlling the strength of the preference signal.

Because the provided sonnet dataset contains only positive canonical sonnets, we constructed the pairwise data manually through automatic generation. We generated the flawed, losing samples (y_l) by using a high generation temperature, injecting sufficient entropy to force the model into structural errors.

4.5 Inference Enhancements: Top-k and Repetition Penalties

During the inference stage for sonnet generation, we implemented Top- k sampling ($k = 50$) to truncate the long tail of the probability distribution, ensuring only high-probability, contextually relevant words were sampled.

Additionally, we coded a custom token repetition penalty into the logits prior to the softmax normalization. Autoregressive models, particularly when fine-tuned on small datasets, are highly prone to getting stuck in generative loops. As an example of what occurs without this penalty, our early generation attempts yielded the following degraded output:

My love is as a fever, longing still
For that which longer nurseth the disease,
Feeding on that which doth preserve the ill,
And wasting away that which doth preserve the well.
I have been as a fever, and yet I have been
Though I had been fed with more nourishment;
I have been as a fever, and yet I have been
Though I had been fed with less; and yet I have been

Though I had been fed with less, and yet I have been
 Though I had been fed with less, and yet I have been
 Though I had been fed with less, and yet I have been
 Though I had been fed with less.
 But love is as a fever, and yet lives,

By applying a multiplicative penalty factor to the logits of previously generated tokens, we successfully forced the model to explore a broader vocabulary and more strictly follow to structure of Shakespearean poetry.

5 Experiments

5.1 Data and Evaluation

- **Sentiment Analysis:** We used the Stanford Sentiment Treebank (SST) and the CFIMDB dataset, containing 11,855 and 2,434 movie reviews, respectively. Performance was evaluated using classification accuracy.
- **Paraphrase Detection:** We utilized a subset of the Quora dataset consisting of 400,000 question pairs. Performance was evaluated using classification accuracy.
- **Sonnet Generation:** The model was trained on a dataset of 143 Shakespearean sonnets. We evaluated the generated sonnets using the chrF metric, a character-level n-gram comparison against the target language distribution.

5.2 Experimental Details

All advanced experiments were conducted on an NVIDIA H100 GPU. To increase efficiency and prevent Out of Memory (OOM) crashes during the 4x forward passes required by DPO, we used mixed-precision `bf16` training and implemented gradient accumulation. In particular, using `bf16` effectively doubled our training speed, with little drawbacks. A linear warmup scheduler was applied to prevent forgetting what the model learned about the English language during the initial training steps. After optimizations, it took around 6 minutes per epoch for the paragraph detection task and around 2 minutes per round of training and generating sonnets for the sonnet task.

Beyond these optimizations, we conducted extensive hyperparameter tuning tailored to the specific representational demands of each downstream task. For sentiment analysis, we primarily explored the architectural configurations (last-linear-layer versus full-model fine-tuning) as specified in the project guidelines [3]. For paraphrase detection, we swept learning rates and LoRA capacities, finding that a larger batch size of 64 combined with a learning rate of $2e-4$ and a minimal LoRA rank ($r = 8$) yielded the best validation accuracy. To prevent overfitting, we included an early stopping mechanism that successfully halted training at epoch 6 (out of a maximum of 20) when the change in loss was sufficiently small. Finally, sonnet generation required the most rigorous tuning due to its complex generative constraints. During the Supervised Fine-Tuning (SFT) phase, we experimented with LoRA ranks ($r \in \{8, 16, 32, 64, 128, 256\}$), identifying $r = 128$ with a dropout of 0.1 as the optimal capacity. We also tuned the warmup step ratio (testing 0%, 10%, 20%, and 30%), settling on 20% to best stabilize early gradients, alongside a learning rate of $1e-4$ and an inference temperature of 0.85 over 8 epochs. For the DPO phase, tuning the parameter to generate the “rejected” samples was important; we gradually decreased the generation temperature from 1.5 down to 0.9 to ensure the negative samples were sufficiently flawed.

5.3 Results

Sentiment Analysis Classification: Our baseline GPT-2 implementation performed competitively on the sentiment classification tasks.

Model Configuration	Dev Accuracy	Test Accuracy
Baselines		
SST Last-Linear	0.462	-
SST Full Model	0.513	-
CFIMDB Last-Linear	0.861	-
CFIMDB Full Model	0.976	-
Our Implementation		
SST Last-Linear	0.470	0.467
SST Full Model	0.495	0.530
CFIMDB Last-Linear	0.865	-
CFIMDB Full Model	0.979	-

Table 1: Sentiment Analysis Accuracy Compared to Baselines

Cloze-Style Paraphrase Detection: For paraphrase detection, we used a batch size of 64 and a learning rate of $2e-4$. Applying LoRA with a rank of $r = 8$ yielded the best performance. We implemented early stopping, halting at epoch 6 (out of 20) when the loss delta dropped below 0.001.

Model	Dev Accuracy	Test Accuracy
Baseline (No LoRA/Scheduler)	0.862	-
<code>gpt2-large</code> + LoRA ($r = 8$)	0.897	0.874

Table 2: Paraphrase Detection Accuracy

Sonnet Generation: For supervised fine-tuning, the best configuration used `gpt2-large`, a learning rate of $1e-4$, and a generation temperature of 0.85 for 8 epochs. LoRA was configured with a rank of $r = 128$, dropout of 0.1, and 20% warmup steps. Table 3 illustrates a sample of the hyperparameter search space.

Configuration	LoRA Rank	LR	Warmup	Epochs	Temp	Dev chrF
Baseline	-	$1e-5$	0%	10	1.2	41.699
Sample Configuration	$r = 64$	$1e-4$	10%	8	0.70	41.750
Sample Configuration	$r = 64$	$2e-4$	10%	6	0.85	41.151
Sample Configuration	$r = 64$	$1e-4$	20%	8	0.85	42.533
Sample Configuration	$r = 256$	$1e-4$	20%	8	0.85	42.754
Optimal	$r = 128$	$1e-4$	20%	8	0.85	43.095
DPO Fine-Tuned	$r = 128$	$1e-4$	20%	8	0.85	43.372

Table 3: Sonnet Generation Performance (chrF)

Applying DPO over the optimal SFT model yielded a final Test chrF of **43.552**.

6 Analysis

Note: this includes both quantitative and qualitative discussion of results.

Sentiment Analysis: Our sentiment classification results were largely in line with our expectations, performing competitively against the expected baselines provided in the project guidelines. As anticipated, the full model fine-tuning significantly outperformed the last-linear-layer approach across both datasets. For instance, on the CFIMDB dataset, the full model achieved a near-perfect dev accuracy of 0.979. While our SST full model dev accuracy (0.495) was slightly lower than the theoretical 0.513 baseline, it generalized exceptionally well to the test set (0.530). This confirms our hypothesis that updating deeper semantic representations is necessary for complex, granular sentiment classification, and that lower overall accuracies on SST are expected due to its more nuanced, 5-class labeling compared to CFIMDB’s binary labels.

Paraphrase Detection: The paraphrase detection results were about as expected. By scaling the architecture to gpt2-large and integrating LoRA, we achieved a dev accuracy of 0.897 and a test accuracy of 0.874, surpassing the gpt2-small baseline of 0.862. Interestingly, we found that a very small LoRA rank ($r = 8$) was optimal. This tells us that evaluating fine-grained notions of semantic meaning in a cloze-style formulation does not require massive re-parameterization of the model’s attention layers. Furthermore, we increased the learning rate from the default $1e-5$ to $2e-4$. Because LoRA updates only a small fraction of the model’s total parameters, a proportionally higher learning rate is more appropriate and necessary to ensure these low-rank matrices converge effectively. Because the pre-trained model already possesses a strong grasp of the English language, introducing this higher learning rate alongside a linear warmup scheduler and minimal rank adaptations was enough to achieve high accuracy without causing the model to forget what it had previously learned.

Sonnet Generation: The final sonnet generation results significantly exceeded our expectations, resulting in a test chrF score of 43.552. We initially thought that autoregressive generation on a small dataset of only 143 training sonnets would lead to severe overfitting, and that applying DPO might fail due to the lack of explicitly labeled negative samples in the corpus. However, our hyperparameter sweep revealed that the fundamental difficulty of this generative task required a much larger LoRA rank ($r = 128$) compared to paraphrase detection, providing the necessary capacity to learn complex poetic structures. Requiring a higher LoRA rank is likely because generating poetry is a much more complicated task than binary classification, which can get by with a much smaller rank.

To successfully implement DPO, we relied heavily on the synthetic generation of "losing" pairwise data by dynamically decreasing the sampling temperature from 1.5 down to 0.9. A high initial temperature of 1.5 was necessary to force the model into making large errors and deviations in the beginning. If these rejected samples were generated at a standard temperature, the model would penalize itself for writing valid poetry, lowering its overall language capabilities. We also utilized a DPO penalty parameter of $\beta = 0.1$ to tightly constrain the policy model to the reference model, which prevented the aggressive structural updates from destroying standard English grammar.

Interestingly, the baseline model performed relatively well, and even after implementing complex optimizations like LoRA, the model didn’t necessarily perform better immediately and required extensive hyperparameter tuning. This suggests that such optimizations are highly sensitive to parameters, and without extensive testing of hyperparameter possibilities, may be outperformed by even the baseline.

Ultimately, the consistent increase in chrF score from the SFT baseline (41.750) to the optimal SFT (43.095) and finally the DPO fine-tuned model (43.372 dev / 43.552 test) validates our approach. The tight alignment between our final dev and test scores confirms that combining Parameter-Efficient Fine-Tuning (PEFT) with preference-based alignment can successfully enforce strict structural constraints and generalize stylistic rules without memorizing validation noise, even in severely data-limited tasks.

7 Conclusion

In this project, we successfully implemented a GPT-2 architecture and demonstrated the impact of modern optimization strategies on constrained downstream tasks. By using Parameter-Efficient Fine-Tuning, we scaled our generative capacity to a model with hundreds of millions of parameters while avoiding the severe overfitting typically associated with small datasets. Furthermore, constructing a synthetic pairwise dataset to execute Direct Preference Optimization was highly effective for sonnet generation, validating that preference-based alignment can enforce strict stylistic rules better than standard supervised fine-tuning. Future work could explore generalizing DPO out-of-distribution or applying quantization-aware fine-tuning to further improve model results on these downstream tasks.

References

- [1] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.
- [2] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2023.

- [3] Stanford CS224N Staff. Cs 224n: Default final project: Build gpt-2. Course Handout, Stanford University, 2026. Provided for the Winter 2026 iteration of CS224N: Natural Language Processing with Deep Learning.