

Efficient Fine-tuning of GPT-2 for Paraphrase Detection and Sonnet Generation

Stanford CS224N Default Project

Jonathan You

Department of Computer Science
Stanford University
youj@stanford.edu

Abstract

Paraphrase detection and sonnet generation are fundamental natural language processing tasks that test a model’s capacity for semantic understanding and constrained creative text generation, respectively. We fine-tune GPT-2 on both tasks and systematically investigate extensions including Low-Rank Adaptation (LoRA), data augmentation, Minimum Bayes Risk (MBR) decoding, and scaling to larger model variants. For paraphrase detection on the Quora Question Pairs dataset, full fine-tuning of GPT-2 small achieves a development accuracy of 0.896, surpassing the staff baseline of 0.882, with a test leaderboard accuracy of 0.872 (vs. a test baseline of 0.856). For sonnet generation, our best configuration—GPT-2 large with LoRA, MBR decoding ($n=10$), data augmentation, and temperature 1.2—achieves a development ChrF score of 43.605 and a test leaderboard score of 43.051. Our analysis reveals that LoRA’s regularization benefit is dataset-size-dependent: it matches full fine-tuning on large datasets but is critical for preventing overfitting on small ones. We also find that MBR decoding provides a meaningful and consistent improvement for generation quality.

1 Key Information to include

- Mentor: Nevin George
- External Collaborators (if you have any): No
- Sharing project: No

2 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across a wide range of NLP tasks when adapted through fine-tuning. However, full fine-tuning of all model parameters is computationally expensive and can lead to overfitting, particularly on small datasets. Parameter-efficient fine-tuning methods such as Low-Rank Adaptation (LoRA) [1] address the computational cost by introducing a small number of trainable parameters while freezing the rest, but the interaction between parameter efficiency, model scale, and dataset size remains underexplored in practice. This project investigates these trade-offs in the context of two complementary tasks: paraphrase detection and sonnet generation.

Paraphrase detection requires the model to determine whether two questions express the same semantic intent, demanding robust sentence-level understanding. The Quora Question Pairs dataset provides over 141K training examples, making it a relatively data-rich setting. Sonnet generation, by contrast, requires producing creative text that adheres to the structural and stylistic conventions of Shakespearean sonnets—fourteen lines of iambic pentameter with a specific rhyme scheme—testing the model’s capacity for constrained language generation on a much smaller training corpus.

Together, these tasks provide a useful testbed for studying how parameter-efficient fine-tuning, decoding strategies, and model scaling interact with dataset characteristics.

Our baseline approach fine-tunes all parameters of GPT-2 small on both tasks. We then explore several extensions: (1) Low-Rank Adaptation (LoRA), which introduces trainable low-rank decompositions into the attention layers while freezing the remaining parameters; (2) scaling from GPT-2 small (124M parameters) to GPT-2 large (774M); (3) data augmentation via a sliding window over sonnet lines to expand the small training set; and (4) Minimum Bayes Risk (MBR) decoding for sonnet generation. We also conduct a systematic grid search over temperature and learning rate for the generation task.

Our experiments yield three main findings. First, LoRA’s effectiveness as a regularizer is closely tied to dataset size: it performs comparably to full fine-tuning on the large paraphrase dataset but is essential for preventing overfitting when scaling to GPT-2 large on the small sonnet dataset. Second, higher sampling temperatures (1.1–1.2) consistently improve sonnet generation quality under ChrF evaluation, suggesting that diversity is rewarded in this setting. Third, MBR decoding provides a simple, orthogonal improvement of approximately 1 ChrF point by selecting consensus candidates from multiple samples. Beyond these headline results, our grid search over temperature and learning rate reveals an interaction between the two hyperparameters, and our qualitative analysis identifies recurring failure modes—such as near-miss semantic pairs in paraphrase detection and structural breakdowns in sonnet generation—that point to concrete directions for future improvement.

3 Related Work

Parameter-efficient fine-tuning. The cost of full fine-tuning has motivated a family of parameter-efficient methods. Houlsby et al. [2] proposed adapter layers—small bottleneck modules inserted between transformer layers—that are trained while the original parameters remain frozen. Li and Liang [3] introduced prefix tuning, which prepends learnable continuous vectors to the key and value sequences in each attention layer, steering model behavior without modifying the underlying weights. Hu et al. [1] proposed LoRA, which decomposes weight updates into low-rank matrices, achieving comparable performance to full fine-tuning on several benchmarks while training only a fraction of the parameters. We adopt LoRA for its simplicity and its natural interpretation as a constraint on the rank of the update, which we hypothesize provides implicit regularization proportional to the gap between model capacity and dataset size.

Paraphrase detection. Identifying whether two sentences convey the same meaning is a well-studied NLP task with applications in duplicate question detection, information retrieval, and data deduplication. Early neural approaches used siamese networks with shared encoders [4], while more recent work has leveraged pretrained transformers. Devlin et al. [5] demonstrated that BERT with cross-encoding achieves strong results on paraphrase benchmarks, including the Quora Question Pairs dataset. Cross-encoding, where both sentences are concatenated and processed jointly, allows rich token-level interaction between the two inputs, which is critical for capturing subtle semantic distinctions. Our approach differs in that we use a generative model (GPT-2) rather than an encoder-only model, framing paraphrase detection as a cloze-style next-token prediction task. This autoregressive formulation is less natural for classification—since GPT-2 processes tokens left-to-right and lacks bidirectional attention—but allows us to use a single model architecture for both the discriminative and generative tasks in this project, providing a useful lens on the versatility of decoder-only language models.

Decoding strategies for text generation. The choice of decoding strategy significantly impacts generation quality. Greedy and beam search decoding tend to produce repetitive, degenerate text from neural language models [6]. Nucleus (top- p) sampling [6] addresses this by truncating the probability distribution to the smallest set of tokens whose cumulative probability exceeds a threshold p , and sampling from this set. Temperature scaling provides a complementary control by sharpening or flattening the distribution before sampling. MBR decoding [7] takes a different approach: rather than modifying the sampling distribution, it generates multiple candidates and selects the one that maximizes expected utility under a chosen metric. MBR has shown strong results in machine translation, and we apply it here to sonnet generation with ChrF as the utility function.

ChrF as an evaluation metric. Popović [8] proposed ChrF, which computes the F-score over character n -grams between hypothesis and reference texts. Unlike word-level metrics such as BLEU,

ChrF captures sub-word morphological similarities and is robust to minor lexical variations, making it appropriate for evaluating poetic text.

4 Approach

4.1 Paraphrase Detection

We formulate paraphrase detection as a cloze-style classification task. Given two questions q_1 and q_2 , we construct the prompt:

q_1 [SEP] q_2 Are these questions asking the same thing?

The model generates a probability distribution over the next token, and we compare the probabilities assigned to the tokens “yes” and “no,” selecting the label with the higher probability. This formulation converts a discriminative task into a generative one, leveraging GPT-2’s language modeling capabilities for classification.

Full fine-tuning. Our baseline fine-tunes all parameters of GPT-2 small (124M parameters) on the Quora Question Pairs training set. All transformer layers, embedding layers, and the language modeling head are updated during training.

LoRA. Following Hu et al. [1], we apply low-rank decompositions to the query and value projection matrices in each self-attention layer. Concretely, for a pretrained weight matrix $W_0 \in \mathbb{R}^{d \times k}$, LoRA parameterizes the update as:

$$W = W_0 + \Delta W = W_0 + BA \quad (1)$$

where $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$ with rank $r \ll \min(d, k)$. During training, W_0 is frozen and only B and A are updated. This reduces the number of trainable parameters per adapted layer from $d \times k$ to $r \times (d + k)$. For inference, the product BA can be merged into W_0 , introducing no additional latency.

4.2 Sonnet Generation

For sonnet generation, the model is fine-tuned to produce Shakespearean sonnets given a starting prompt. The input consists of a sonnet prefix, and the model is trained to continue the text in the style and structure of a Shakespearean sonnet. We explore several strategies to improve generation quality.

Model scaling. We scale from GPT-2 small (124M parameters) to GPT-2 large (774M). Both are fine-tuned with LoRA to manage the increased risk of overfitting that comes with larger models on a small dataset. Larger models have greater capacity to capture the stylistic and structural patterns of sonnets—including meter, rhyme, and thematic coherence—but this capacity is a double-edged sword when training data is limited.

Data augmentation via sliding window. The sonnet training set contains only 143 sonnets, making overfitting a significant concern. To increase the effective training set size, we apply a sliding window augmentation: for each sonnet, we create multiple training examples by shifting the starting line. Specifically, we generate three versions of each sonnet by starting from line 0 (the full 14-line sonnet), line 1 (lines 1–13), and line 2 (lines 2–13). This triples the training set from 143 to 429 examples. The partial sonnets expose the model to a greater variety of continuations and reduce memorization of fixed starting patterns.

Temperature scaling. At generation time, we apply temperature τ to control the entropy of the output distribution. Given logits z_i for each token i in the vocabulary, the sampling probability is:

$$p_i = \frac{\exp(z_i/\tau)}{\sum_j \exp(z_j/\tau)} \quad (2)$$

When $\tau < 1$, the distribution becomes sharper, favoring high-probability tokens. When $\tau > 1$, the distribution flattens, increasing the probability of less likely tokens and producing more diverse outputs. We hypothesize that moderate increases in temperature ($\tau \approx 1.1$ – 1.2) are beneficial for creative generation, where diversity is desirable and the “correct” output is not unique.

Minimum Bayes Risk (MBR) decoding. Standard autoregressive decoding produces a single sample from the model’s output distribution, which can be high-variance. MBR decoding reduces this variance by generating n candidate outputs and selecting the one that maximizes expected utility across all candidates:

$$\hat{y} = \arg \max_{y_i} \frac{1}{n} \sum_{j=1}^n u(y_i, y_j) \quad (3)$$

where $u(y_i, y_j)$ is a utility function measuring the similarity between candidates y_i and y_j . We use ChrF as the utility function and set $n = 10$. Intuitively, MBR selects the candidate that is most “central” among the samples, acting as a consensus filter that suppresses outliers. The computational cost scales quadratically in n (since all pairwise utilities must be computed), but with $n = 10$ this remains tractable.

5 Experiments

5.1 Data

Paraphrase detection. We use the Quora Question Pairs dataset, which contains pairs of questions labeled with binary indicators of semantic equivalence. The dataset is split into 141,506 training examples, 20,215 development examples, and 40,431 test examples. The dataset exhibits moderate class imbalance, with approximately 37% of pairs labeled as paraphrases.

Sonnet generation. We use the provided sonnet dataset, which contains 143 training sonnets (out of 154 total Shakespearean sonnets, with 12 held out for testing). With our sliding window augmentation (Section 4.2), the effective training set is expanded to 429 examples. This dataset is substantially smaller than the paraphrase dataset—by nearly three orders of magnitude—which has important implications for regularization and overfitting, as discussed in our analysis (Section 6).

5.2 Evaluation Method

For paraphrase detection, we report accuracy: the fraction of examples for which the model correctly predicts the paraphrase label on the development set, as well as the test leaderboard score. For sonnet generation, we report ChrF (Character n-gram F-score) [8], which computes the F-score over character n-grams between generated and reference sonnets. ChrF is well-suited for evaluating poetic text because it captures sub-word similarities, is less sensitive to exact word matching than token-level metrics like BLEU, and is robust to the creative lexical variation that characterizes poetry.

5.3 Experimental Details

All experiments were conducted on a single NVIDIA T4 GPU (16 GB VRAM) via Google Colab.

Paraphrase detection. We used a learning rate of 1×10^{-5} , a batch size of 8, and trained for 10 epochs with the AdamW optimizer. Each epoch required approximately 1.5 hours of training time with full fine-tuning. With LoRA, training time per epoch was substantially reduced, though the improvement on the T4 was primarily in memory usage rather than wall-clock time due to the relatively small model size.

Sonnet generation. For GPT-2 large with LoRA (rank 8), we conducted a grid search over three learning rates (2×10^{-6} , 5×10^{-6} , and 1×10^{-5}) and three temperatures (0.7, 0.9, 1.1), for a total of 9 configurations. We selected rank $r = 8$ following the recommendations in Hu et al. [1], who found that low ranks (4–8) suffice for adaptation on downstream tasks; exploring the sensitivity to rank is left for future work. We applied early stopping based on development ChrF, typically stopping around epoch 3–4 to prevent overfitting. For MBR decoding, we generated $n = 10$ candidates per prompt at the best temperature and selected the candidate with the highest average pairwise ChrF.

5.4 Results

Paraphrase detection. Table 1 summarizes our paraphrase detection results. Full fine-tuning achieves the best development accuracy of 0.896, with LoRA close behind at 0.893. Both sub-

stantially outperform the staff baseline of 0.882. On the test leaderboard, our best model achieves 0.872, compared to the staff baseline of 0.856.

Table 1: Paraphrase detection results.

Configuration	Dev Accuracy	Test Accuracy
Staff baseline	0.882	0.856
GPT-2 small, full fine-tuning	0.896	0.872
GPT-2 small + LoRA	0.893	—

Sonnet generation. Table 2 presents our sonnet generation results across all configurations. The best development ChrF of 43.605 is achieved by GPT-2 large with LoRA, MBR decoding ($n=10$), data augmentation, and temperature 1.2. Our test leaderboard score of 43.051 uses the same configuration as our best development model. We note that sliding window augmentation yields no measurable improvement on GPT-2 small (41.947 with and without), likely because the small model’s limited capacity is the binding constraint rather than dataset size. However, augmentation contributes to the best GPT-2 large configuration, where the model has sufficient capacity to benefit from additional training examples.

Table 2: Sonnet generation results (development set ChrF and test leaderboard).

Configuration	Dev ChrF
Staff baseline	41.414
GPT-2 small, no LoRA, lr=1e-5	41.963
GPT-2 small + LoRA, lr=1e-5	41.947
GPT-2 small + LoRA + sliding window aug.	41.947
GPT-2 large + LoRA, lr=1e-5, $\tau=1.2$	42.213
GPT-2 large + LoRA, lr=5e-6, $\tau=1.1$	42.818
GPT-2 large + LoRA, lr=1e-5, $\tau=1.2$, MBR=10, aug.	43.605
Test leaderboard (staff baseline: 41.208)	43.051

Grid search findings. Table 3 shows the results of our grid search over temperature and learning rate for GPT-2 large with LoRA (rank 8, no MBR). We observe two clear trends. First, higher temperatures consistently yield better ChrF scores: every learning rate improves monotonically from $\tau = 0.7$ to $\tau = 1.1$, with the gap between 0.7 and 1.1 ranging from 1.8 to 3.6 ChrF points. Second, the optimal learning rate shifts with temperature. At lower temperatures ($\tau = 0.7$ and 0.9), $lr = 1 \times 10^{-5}$ performs best, while at $\tau = 1.1$ the more conservative $lr = 5 \times 10^{-6}$ edges ahead (42.819 vs. 42.683). This interaction suggests that higher-entropy decoding pairs better with smaller weight updates, possibly because aggressive fine-tuning narrows the output distribution in ways that conflict with high-temperature sampling.

Table 3: Grid search results for GPT-2 large + LoRA, rank 8, no MBR (dev ChrF). Best result per temperature is **bolded**.

Learning Rate	$\tau=0.7$	$\tau=0.9$	$\tau=1.1$
2×10^{-6}	38.467	40.810	42.066
5×10^{-6}	40.010	41.237	42.819
1×10^{-5}	40.933	41.935	42.683

6 Analysis

6.1 LoRA as Dataset-Size-Dependent Regularization

Our most informative finding concerns the contrasting role of LoRA across the two tasks. On paraphrase detection, where the training set contains over 141K examples, LoRA (0.893) performs comparably to full fine-tuning (0.896)—a gap of only 0.003. With abundant data and a relatively small

model (124M parameters), overfitting risk is low regardless of whether all parameters are updated, and the small accuracy gap likely reflects the reduced expressiveness of the low-rank update space.

On sonnet generation, however, where the training set contains only 143 sonnets (429 with augmentation), the picture reverses. GPT-2 large (774M parameters) overfits severely under full fine-tuning, with development ChrF peaking around epoch 3–4 and then declining. LoRA prevents this by constraining updates to a low-rank subspace, enabling us to leverage GPT-2 large’s superior representational capacity without paying the overfitting penalty.

This pattern is consistent with LoRA’s theoretical motivation: the rank constraint acts as an implicit regularizer whose strength depends on the ratio of model capacity to dataset size. When data is plentiful, the constraint is unnecessary; when data is scarce, it becomes essential.

6.2 Temperature and Generation Diversity

Our grid search reveals a clear and consistent trend: higher sampling temperatures improve ChrF scores for sonnet generation. Temperature 1.1 outperforms 0.9, which outperforms 0.7, across all three learning rates tested. The improvement is substantial—the best $\tau = 1.1$ configuration (42.819) exceeds the best $\tau = 0.7$ configuration (40.933) by nearly 2 ChrF points.

This finding reflects the nature of both the task and the metric. Sonnets reward lexical diversity and creative word choices; lower temperatures cause the model to repeatedly select high-probability tokens, producing grammatically correct but stylistically flat text. ChrF, which operates at the character n-gram level, rewards diverse vocabulary choices that would be penalized by stricter word-level metrics. However, very high temperatures ($\tau \gg 1.2$) would eventually degrade coherence, and identifying the upper bound remains future work.

6.3 MBR Decoding as Post-Hoc Quality Improvement

MBR decoding with $n = 10$ candidates provides an approximately 1-point ChrF improvement, boosting the best configuration from ~ 42.8 (single sample) to 43.605 (MBR). This improvement is notable because MBR is fully orthogonal to other optimizations—it operates at decoding time, agnostic to training procedure and model architecture, and can be stacked on top of any configuration without interference.

MBR’s mechanism acts as a variance reduction technique: by selecting the candidate most similar to all others, it filters out low-quality outliers such as truncated or incoherent outputs. The consistent improvement suggests meaningful variance in generation quality across samples, with the central tendency being better than a random draw. The computational cost is modest—with $n = 10$, we compute $\binom{10}{2} = 45$ pairwise ChrF scores per example, which is inexpensive relative to generation itself.

6.4 Error Analysis

To better understand our model’s failure modes, we examined misclassified examples from the paraphrase detection development set. Our model produced 2,660 false positives (predicted paraphrase when the gold label was not) and 1,558 false negatives (predicted not paraphrase when the gold label was yes), for an overall error rate of 10.4%.

False positives: lexical overlap masking semantic differences. The model’s most common failure mode was predicting paraphrase for question pairs with high surface-level overlap but distinct meanings. For example:

Q1: “What is vaccination?”
Q2: “What is in a vaccine?”

These questions share the *vaccin-* morphological stem and nearly identical syntactic structure, but ask fundamentally different things: one requests a definition, the other asks about ingredients. Similarly, the model incorrectly labeled the following pair as paraphrases:

Q1: “Who will be the next President of the United States?”
Q2: “Who should be the next president, and why?”

Here, the distinction hinges on a single modal verb: *will be* asks for a prediction, while *should be* solicits an opinion. The cloze-style formulation, which relies on the model’s implicit next-token predictions rather than explicit sentence representations, appears to conflate lexical similarity with semantic equivalence.

False negatives: semantic equivalence despite surface variation. Conversely, the model failed to recognize paraphrases when the two questions expressed the same intent with substantially different wording:

Q1: “Why can’t desalination solve the drinking water problem for India?”

Q2: “Why can’t India set up many sea water desalination plants to solve water problems?”

Both questions ask why desalination is insufficient for India’s water crisis, but the second uses a much more specific framing. The left-to-right autoregressive processing of GPT-2 may make it difficult to recognize semantic equivalence when the shared meaning only becomes apparent after reading both questions in full. This contrasts with cross-encoder architectures like BERT [5], which attend bidirectionally over the concatenated input.

Sonnet generation failure modes. We examined generated sonnets from GPT-2 small (no LoRA, $\text{lr}=1\text{e-}5$) across training epochs to identify common structural failures.

Premature truncation. Many continuations produced far fewer than the expected lines. Given the prompt beginning “So, now I have confessed that he is thine,” the model at epoch 1 generated only: “Myself I’ll forfeit, so that other mine freedom;”—a single line that abandons the sonnet structure entirely.

Thematic incoherence. The model frequently injected incongruous references that broke the Shakespearean register. At epoch 0, a continuation included “Take and look above Israel & Blessed Israel’s mighty blood & Potestiar stone”; other outputs referenced Ex Machina, Alcibiades, and Paris—mixing biblical, classical, and modern references incoherently.

Broken rhyme and meter. Even at reasonable length, rhyme schemes frequently failed. At epoch 1, a continuation of “Be wise as thou art cruel” produced:

“My pleasure, le though compulsive wit develops,
remains only lieth thy love
In spending my mind in fooling and fussing,
which, at points left unappreciated, comfort pays for thou prd.”

The output lacks any consistent rhyme scheme and degrades into fragments. These failures were most severe at low temperatures and early epochs, and were substantially mitigated by scaling to GPT-2 large with LoRA, higher temperatures, and MBR decoding.

7 Conclusion

We fine-tuned GPT-2 for paraphrase detection and sonnet generation, achieving a development accuracy of 0.896 (test: 0.872) for paraphrase detection and a best development ChrF of 43.605 (test: 43.051) for sonnet generation, both exceeding the staff baselines. Notably, while the GPT-2 small baseline for sonnet generation only marginally exceeds the staff baseline (41.963 vs. 41.414), our extensions—model scaling, LoRA, temperature tuning, data augmentation, and MBR decoding—yield a cumulative improvement of over 1.6 ChrF points, demonstrating that the gains are driven by methodological choices rather than simply fine-tuning a larger model. Our systematic exploration of extensions revealed three main findings: (1) LoRA provides comparable performance to full fine-tuning on large datasets while serving as essential regularization on small ones, with its value increasing with the model-size-to-dataset-size ratio; (2) higher sampling temperatures consistently improve generation quality under ChrF evaluation, reflecting the metric’s tolerance for lexical diversity; and (3) MBR decoding offers a simple, orthogonal, post-hoc improvement for generation tasks by acting as a variance reduction mechanism.

The primary limitation of this work is computational: with access to only a single T4 GPU, we were unable to explore GPT-2 XL (1.5B parameters), conduct more extensive hyperparameter sweeps,

or test larger values of n for MBR decoding. Additionally, we did not explore nucleus sampling (top- p) in combination with temperature scaling, which could further improve generation quality. Future work could investigate the interaction between LoRA rank and dataset size more systematically, explore alternative utility functions for MBR decoding beyond ChrF, and examine whether the temperature–learning rate interaction we observed generalizes to other generation tasks and evaluation metrics.

References

- [1] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.
- [2] Neil Houlsby, Andrei Giber, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning (ICML)*, 2019.
- [3] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 2021.
- [4] Jonas Mueller and Aditya Thyagarajan. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.
- [6] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*, 2020.
- [7] Bryan Eikema and Wilker Aziz. Sampling-based minimum Bayes risk decoding for neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.
- [8] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, 2015.