

Cloze-Style Paraphrase Detection and Sonnet Generation with GPT-2: Exploring LoRA and Decoding Strategies

Stanford CS224N Default Project

Nick Fursa

Department of Computer Science
Stanford University
nfursa@stanford.edu

Abstract

We implement GPT-2 from scratch, load pretrained weights, and fine-tune it on sentiment analysis, cloze-style paraphrase detection, and sonnet generation. For paraphrase detection, we reformulate binary classification as next-token prediction, prompting the model to generate “yes” or “no” via a weight-tied output projection. Our best full fine-tuning configuration achieves 89.7% dev accuracy and 87.4% test accuracy on the Quora paraphrase dataset, exceeding the staff baseline of 85.6%. We implement Low-Rank Adaptation (LoRA) as a parameter-efficient alternative, finding that LoRA with rank 16 matches the staff baseline accuracy (88.3% vs. 88.2%) while training only 0.48% of parameters. For sonnet generation, we conduct a systematic sweep over decoding hyperparameters, finding that temperature 1.0 with nucleus sampling ($p = 0.95$) achieves a chrF score of 41.661 on dev and 41.628 on test, surpassing the staff baseline of 41.208. We provide thorough ablations across LoRA ranks and decoding configurations, with qualitative analysis of generated outputs.

1 Key Information

- Mentor: Ali Sartaz Khan
- External Collaborators: None
- Sharing project: No

2 Introduction

Large language models pretrained on next-token prediction acquire broad linguistic knowledge that transfers effectively to downstream tasks via fine-tuning [1]. GPT-2, a decoder-only Transformer trained on WebText, demonstrated that scaling model size and data yields emergent capabilities across diverse tasks without task-specific architectures [2]. In this project, we implement GPT-2 from scratch, verify correctness against HuggingFace weights, and fine-tune it on three tasks of increasing complexity.

A key design choice in our paraphrase detection system is the *cloze-style* formulation: rather than adding a classification head, we prompt the model with the question pair and predict the next token, where “yes” (BPE index 8505) indicates a paraphrase and “no” (BPE index 3919) indicates otherwise. This leverages GPT-2’s pretrained language modeling capability directly, framing classification as generation.

While full fine-tuning achieves strong results, it requires updating all 124M parameters, which is costly and limits scalability to larger models. We address this through **Low-Rank Adaptation**

(LoRA) [3], which injects small trainable rank decomposition matrices into the attention layers while freezing the base model. We conduct a systematic ablation across ranks $r \in \{2, 4, 8, 16\}$, revealing a smooth accuracy–efficiency tradeoff where rank 16 (0.48% of parameters) matches the staff full fine-tuning baseline.

For sonnet generation, we fine-tune GPT-2 as a language model on Shakespeare’s sonnets and investigate how decoding hyperparameters - temperature and nucleus sampling threshold - affect generation quality as measured by chrF [4]. We find that the choice of decoding parameters substantially impacts output quality, with moderate temperature ($\tau = 1.0$) outperforming both conservative ($\tau = 0.8$) and aggressive ($\tau = 1.2$) settings.

3 Related Work

Pretrained language models. The Transformer architecture [5] enabled large-scale pretraining of language models. GPT-1 [2] demonstrated that unsupervised pretraining followed by supervised fine-tuning improves performance on diverse NLP tasks. GPT-2 [1] scaled this approach, showing strong zero-shot performance. BERT [6] explored bidirectional pretraining but requires task-specific heads, whereas our cloze-style approach stays within the generative paradigm.

Parameter-efficient fine-tuning. Full fine-tuning becomes impractical as models grow. Hu et al. [3] proposed LoRA, which adds low-rank update matrices $\Delta W = BA$ to frozen pretrained weights, where $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times d}$ with $r \ll d$. This reduces trainable parameters by orders of magnitude while maintaining competitive performance. We apply LoRA to the query and value projections in GPT-2’s attention layers and ablate across four ranks.

Decoding strategies. Holtzman et al. [7] showed that maximization-based decoding (e.g., greedy, beam search) leads to degenerate, repetitive text, and proposed nucleus (top- p) sampling as an alternative. Temperature scaling adjusts the sharpness of the output distribution before sampling. We systematically vary both parameters and evaluate using chrF [4], a character-level n -gram F-score that correlates well with human judgments for generation tasks.

4 Approach

4.1 GPT-2 Implementation

We implement GPT-2 small (12 layers, 768 hidden dimensions, 12 attention heads, 124M parameters) from scratch, including causal multi-head self-attention with an upper-triangular mask, pre-norm Transformer layers (LayerNorm before each sub-layer), token and learned positional embeddings, and the AdamW optimizer [8]. Pretrained weights from HuggingFace are loaded and verified via a sanity check against the reference implementation.

4.2 Cloze-Style Paraphrase Detection

Each question pair (s_1, s_2) is formatted as:

Question 1: "{s1}" Question 2: "{s2}" Are these questions asking the same thing?

The model predicts the next token using GPT-2’s weight-tied output projection $h \cdot E^\top$, where E is the input embedding matrix, producing logits over the full 50,257-token BPE vocabulary. The model is trained with cross-entropy loss on the target token IDs (8505 for “yes”, 3919 for “no”).

4.3 Sonnet Generation

We fine-tune GPT-2 as a causal language model on 143 Shakespeare sonnets using next-token prediction loss. At inference, the model is conditioned on the first three lines of a held-out sonnet and generates the remaining lines autoregressively using top- p (nucleus) sampling with temperature scaling.

4.4 LoRA: Low-Rank Adaptation

We implement LoRA by injecting parallel low-rank paths into the query (W_Q) and value (W_V) projection layers of each attention head. For a pretrained weight matrix $W_0 \in \mathbb{R}^{d \times d}$, the modified forward pass computes:

$$h(x) = W_0 x + \frac{\alpha}{r} \cdot x A^\top B^\top \quad (1)$$

where $A \in \mathbb{R}^{r \times d}$ (initialized from $\mathcal{N}(0, 1/d)$) and $B \in \mathbb{R}^{r \times d}$ (initialized to zero), so the adapter output is zero at initialization. The base model parameters are frozen; only A and B are trained. We set the scaling factor $\alpha = r$ throughout, yielding an effective scale of 1.0.

4.5 Code Attribution

The starter code, including data loading, evaluation utilities, and training scaffolding, is provided by the CS224N course.¹ All model implementations (attention, GPT-2 layers, embeddings, optimizer, classifier, paraphrase forward pass, sonnet forward pass) and the LoRA module were implemented by us.

5 Experiments

5.1 Data

We use four datasets: (1) the Stanford Sentiment Treebank (SST) [9] with 8,544/1,101 train/dev examples and 5 sentiment classes; (2) CFIMDB with 1,701/245 train/dev examples and binary sentiment labels; (3) Quora Question Pairs with 141,506/20,215/40,431 train/dev/test examples for binary paraphrase detection; and (4) 143 Shakespeare sonnets for language model fine-tuning, with 12 held-out dev and 12 held-out test sonnets for generation evaluation.

5.2 Evaluation Methods

Sentiment analysis and paraphrase detection are evaluated by accuracy. Paraphrase predictions are the argmax over the full vocabulary logits, yielding token ID 8505 (“yes”) or 3919 (“no”). Sonnet generation is evaluated by chrF [4], a character-level n -gram F-score comparing generated continuations to the original Shakespeare sonnets.

5.3 Experimental Details

All experiments use GPT-2 small. For last-linear-layer sentiment, we train for 15 epochs with batch size 64, learning rate 10^{-3} , and dropout 0.3 on CPU. For full-model sentiment, we use learning rate 10^{-5} , dropout 0.1, and 15 epochs on GPU.

For paraphrase detection (full fine-tuning), we sweep learning rates $\{5 \times 10^{-6}, 10^{-5}, 2 \times 10^{-5}\}$ with batch size 16, no weight decay, and 5–10 epochs, selecting the best checkpoint by dev accuracy. For LoRA experiments, we train for 15 epochs with learning rate 10^{-4} , batch size 16, and sweep ranks $r \in \{2, 4, 8, 16\}$ with $\alpha = r$.

For sonnet generation, we train for 10 epochs with learning rate 10^{-5} and sweep decoding parameters: temperature $\tau \in \{0.8, 1.0, 1.2\}$ and $p = 0.95$. All GPU training uses a single NVIDIA L4 (23GB) on Google Cloud.

5.4 Results

Sentiment analysis. Table 1 confirms our GPT-2 implementation is correct, with results meeting or exceeding the project handout baselines.

Paraphrase detection. Table 2 shows our hyperparameter sweep results. The best configuration ($\text{lr}=2 \times 10^{-5}$, 8 epochs) achieves 89.7% dev accuracy and 87.4% test accuracy, both exceeding the staff baselines (88.2% dev, 85.6% test).

¹https://github.com/stanfordnlp/cs224n_gpt

Table 1: Sentiment analysis dev accuracy.

Task	Configuration	Dev Accuracy
SST (5-class)	Last-linear-layer	0.467
SST (5-class)	Full-model	0.513
CFIMDB (binary)	Last-linear-layer	0.873
CFIMDB (binary)	Full-model	0.984

Table 2: Paraphrase detection hyperparameter sweep (full fine-tuning).

Learning Rate	Epochs	Best Dev Acc	Best Epoch
5×10^{-6}	10	0.893	10
1×10^{-5}	5	0.893	5
2×10^{-5}	8	0.897	4

LoRA rank ablation. Table 3 presents results across four LoRA ranks compared to full fine-tuning. Accuracy increases monotonically with rank, and LoRA at $r = 16$ matches the staff full fine-tuning baseline while training fewer than 0.5% of the model’s parameters.

Table 3: LoRA rank ablation for paraphrase detection ($\text{lr}=10^{-4}$, 15 epochs, $\alpha = r$).

Rank (r)	Trainable Params	% of Total	Dev Accuracy
2	73,728	0.06%	0.859
4	147,456	0.12%	0.868
8	294,912	0.24%	0.874
16	589,824	0.48%	0.883
Full	124M	100%	0.897

Sonnet generation. Table 4 reports chrF scores across decoding configurations. Temperature $\tau = 1.0$ with $p = 0.95$ achieves the highest chrF on both dev and test, surpassing the staff baseline.

Leaderboard summary. Table 5 compares our best results to the staff baselines on both dev and test sets.

6 Analysis

6.1 LoRA: Parameter Efficiency vs. Accuracy

The LoRA rank ablation (Table 3) reveals a smooth, monotonically increasing relationship between rank and accuracy, with clear diminishing returns. Doubling the rank from 2 to 4 yields a 0.9 percentage point gain, while doubling from 8 to 16 yields the same 0.9 points but at $4\times$ the parameter cost. At $r = 16$, LoRA achieves 0.883 dev accuracy - matching the staff full fine-tuning baseline of 0.882 - with only 590K trainable parameters (0.48% of the model). The remaining 1.4 point gap to our best full fine-tuning result (0.897) suggests that while low-rank updates capture most task-relevant adaptations, some capacity is lost through the rank bottleneck.

This result has practical implications: LoRA enables fine-tuning on hardware with limited memory, makes it feasible to serve multiple task-specific adaptations from a single base model, and reduces the storage cost of checkpoints from 500MB to under 3MB.

6.2 Decoding Strategy Analysis

The decoding sweep reveals that the relationship between temperature and chrF is non-monotonic. Low temperature ($\tau = 0.8$) produces the most coherent individual lines but suffers from repetitive phrasing (e.g., “love’s love’s gain,” “mine heart’s fair and mine heart’s fair”), reducing character-level n -gram diversity and thus chrF. High temperature ($\tau = 1.2$) introduces greater lexical variety

Table 4: Sonnet generation chrF scores across decoding configurations.

Temperature (τ)	Top- p	Dev chrF	Test chrF
0.8	0.95	39.956	-
1.0	0.95	41.661	41.628
1.2	0.95	41.282	-
Staff baseline		41.414	41.208

Table 5: Test leaderboard results vs. staff baselines.

Task	Ours (Test)	Staff (Test)	Δ
Paraphrase (Accuracy)	0.874	0.856	+0.018
Sonnet (chrF)	41.628	41.208	+0.420

but also produces incoherent phrases (e.g., “hot sugar,” “hit-slaats rechips”) that diverge from Shakespearean vocabulary. The standard temperature ($\tau = 1.0$) balances these extremes, producing text that maintains thematic coherence while using sufficiently diverse vocabulary to achieve high character-level overlap with the original sonnets.

This finding aligns with Holtzman et al. [7], who observed that the probability distributions of human-generated text are neither peaked nor flat. The optimal $\tau = 1.0$ result suggests that GPT-2’s pretrained distribution, when fine-tuned on sonnets, already produces a well-calibrated distribution that does not benefit from further sharpening or flattening.

Qualitative examples comparing outputs across all three decoding configurations are provided in Appendix A.

6.3 Cloze-Style Formulation

The cloze-style approach achieves strong results without a task-specific classification head, which is notable because the model must select “yes” or “no” from a vocabulary of 50,257 tokens. The cross-entropy loss over the full vocabulary provides a richer training signal than binary cross-entropy, as the model also learns to suppress logits for semantically related tokens (e.g., “maybe,” “perhaps”). This may partially explain why the cloze-style approach exceeds simple baseline expectations despite not using a dedicated classifier.

7 Conclusion

We implemented GPT-2 from scratch and demonstrated its effectiveness on sentiment analysis, paraphrase detection, and sonnet generation. Our main contributions are:

1. A systematic LoRA rank ablation showing that 0.48% of parameters suffice to match full fine-tuning baselines for paraphrase detection, with a smooth accuracy–rank tradeoff.
2. A decoding parameter sweep revealing that standard temperature ($\tau = 1.0$) with nucleus sampling ($p = 0.95$) optimally balances coherence and diversity for sonnet generation, achieving chrF 41.628 on test (vs. staff baseline 41.208).
3. Test leaderboard results exceeding staff baselines on both tasks.

Limitations. Our LoRA experiments are limited to GPT-2 small; scaling to GPT-2 medium would test whether LoRA’s advantage grows with model size, as suggested by Hu et al. [3]. The sonnet decoding sweep covers only three temperature settings; a finer grid or learned temperature might yield further improvements. Our sonnet training uses no early stopping, which could reduce overfitting on the small 143-sonnet corpus.

Future work. Applying LoRA to sonnet generation, experimenting with GPT-2 medium, and exploring learned decoding parameters are natural next steps.

References

- [1] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.
- [2] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI blog*, 2018.
- [3] 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. *International Conference on Learning Representations*, 2022.
- [4] Maja Popović. chrF: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, 2015.
- [5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.
- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.
- [7] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020.
- [8] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.
- [9] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, 2013.

A Sonnet Generation Examples

Below we compare outputs from the same prompt (the first three lines of held-out Sonnet 0) under different decoding configurations. The prompt lines are shown in *italics*.

Prompt:

*Thine eyes I love, and they, as pitying me,
Knowing thy heart torment me with disdain,
Have put on black, and loving mourners be,*

$\tau = 0.8, p = 0.95$ (**chrF 39.956**):

The love of their love’s pleasure.
Yet they live for my weary and sad heart,
And all for my dearest love’s loss,
Which, if I should be slain, shall not prevent.
 How may my thoughts dwell in him,
 And render my love to him better;
 How, my weary weary heart, which I love so much,
 Shall you do no more than the hour of thy doom.

Note: Coherent but repetitive (“love” appears 5 times, “weary” repeated). Conservative sampling leads to lexical poverty.

$\tau = 1.0, p = 0.95$ (**chrF 41.661**):

Said: If thou art dead and my old love,
On earth, then my stripes shall be bound with shame;
Suppose the sun to leave the earth being next,
Which heaven so desirest, but beckons,
Not ill to my cries o’er the world?

Then dost thou return 't thou whether I say "I love thee ill?"
That 'tis better to fear the sun nor do I hold thee,
Which world, to what love should they owe?
So will thou belie and despair thee
To thy threatening part,
That never

Note: Greater variety and Shakespearean register. Uses diverse imagery (sun, earth, heaven, stripes) without degenerate repetition.

$\tau = 1.2, p = 0.95$ (**chrF 41.282**):
Above all their lofty detachment,
Who soon to tranquillise me, snuff out sorrow,
And filled me with so grand languor a voice.
Hearing thou groan defeat, wheled praise and love,
Savants fooluou, my green eyes sly-witted sits,
Inducing with fraud and burdens that well fly,
And pining poor's say my woes with hearsawdust winds,
Caceting preisance and let my common suspicions
Or lion vows get their way with wracking desert swift.
Yet cunning still still my bounds thou give me,
My

Note: More creative but less coherent. Produces non-words ("fooluou," "Caceting," "hearsawdust") and unusual constructions. Higher temperature increases lexical novelty at the cost of grammaticality.