

Investigating Structure Aware Decoding and Cloze Style Classification for Robust GPT 2 Fine Tuning

Stanford CS224N Default Project

Anya Von Diessl

Department of Computer Science
Stanford University
anya2025@stanford.edu

Abstract

Fine tuning pretrained language models on small datasets often exposes mismatches between training objectives and evaluation criteria. We investigate this phenomenon on two tasks: sonnet generation and paraphrase detection using GPT 2. For sonnet generation, we find that standard nucleus sampling after fine tuning on 143 Shakespeare sonnets frequently produces structurally invalid outputs. We study structure aware decoding enforcing line count constraints, applying length sensitive new-line boosting, and suppressing premature end of sequence tokens combined with best of k candidate reranking via structural heuristics. For paraphrase detection, we explore a cloze style reformulation that replaces a learned classification head with GPT 2's own vocabulary predictions for "yes" and "no," and examine the effects of random deletion augmentation and L2 regularization on robustness. Our experiments show that decoding time structural constraints substantially improve sonnet quality as measured by chrF, and that cloze style classification with data perturbation improves paraphrase detection accuracy.

1 Key Information to include

- Mentor: Wei Liu
- External Collaborators (if you have any):
- Sharing project: No
- Late Days (if you are using any):

Team member	Late days remaining
<Anya Von Diessl>	<>
Total pooled	<OAE late days>
Team size	<1>
Late days applied for the report	<OAE late days>

2 Introduction

Large pretrained language models such as GPT 2 [1] generate fluent text but struggle with structured outputs and nuanced classification when fine tuned on small datasets. Sonnet generation, which demands exactly 14 lines with consistent line lengths and rhyme patterns, reveals the limits of standard token level training: the model cannot reliably learn such structural constraints from 143 training examples alone. Paraphrase detection, where surface lexical overlap can mislead a classifier, raises questions about how the fine tuning objective itself shapes robustness.

We investigate two research questions. First, can decoding time structural constraints and sequence level candidate selection compensate for what token level fine tuning cannot learn? We introduce

a line aware decoding procedure with EOS suppression, newline boosting, and repetition penalties, supplemented by a best of k reranking strategy that scores candidates using structural heuristics. Second, does reformulating classification as cloze style next token prediction improve robustness, and how do data augmentation and regularization interact with this formulation? We replace the standard classification head with a prompt based approach that compares GPT 2’s logits for “yes” and “no,” and study the effects of random word deletion and L2 regularization.

Our main findings are: (1) line aware decoding eliminates structural failures in sonnet generation and best of k reranking provides further chrF improvements; (2) cloze style paraphrase detection achieves competitive accuracy, and random deletion augmentation reduces overfitting to lexical overlap; (3) L2 regularization provides modest additional stability during fine tuning.

3 Related Work

Neural Poetry Generation. Generating poetry with neural models has been explored using both recurrent [2] and transformer architectures. Liao et al. [3] showed that GPT 2 can produce stylistically coherent verse but struggles with structural properties like rhyme and meter. We build on this by adding explicit structural constraints at decoding time rather than relying on the model to learn them from limited data.

Controlled Text Generation. Nucleus sampling [4] addresses degenerate repetition by sampling from the top p probability mass. Repetition penalties reduce looping in autoregressive models [5]. Best of k sampling, where multiple candidates are scored and the best selected, has improved output quality in dialogue and summarization [6]. We combine these ideas with domain specific structural heuristics for sonnet generation.

Prompt Based Classification. Schick and Sch"utze [7] demonstrated that reformulating classification as natural language prompts can improve few shot performance by aligning fine tuning with the pretraining objective. Our cloze style paraphrase detection follows this paradigm.

Data Augmentation for NLP. Wei and Zou [8] proposed Easy Data Augmentation techniques including random deletion for text classification. We apply random deletion to paraphrase detection to reduce reliance on superficial lexical patterns.

Regularized Fine Tuning. Jiang et al proposed SMART, which uses adversarial perturbations to regularize fine tuning. We implement a simplified L2 variant as a lightweight stability mechanism.

4 Approach

We address sonnet generation and paraphrase detection with GPT 2 (small, 124M parameters). For each task, we describe the baseline and our extensions.

4.1 Baseline GPT 2 Sonnet Model

The baseline fine tunes all GPT 2 parameters using standard next token cross entropy loss on sonnet sequences. We implemented the forward pass by running tokens through the GPT 2 transformer to obtain hidden states and projecting into vocabulary space:

$$h_1, \dots, h_T = \text{GPT 2}(x_1, \dots, x_T), \quad \text{logits}_t = W_{\text{vocab}} h_t \quad (1)$$

Decoding uses temperature scaled nucleus sampling with a fixed maximum length, stopping at the EOS token.

4.2 Line Aware Decoding

A Shakespearean sonnet requires exactly 14 lines. The baseline decoder cannot enforce this, leading to premature stopping, variable line lengths, and structural failures. Our line aware decoder makes the following modifications:

EOS Suppression. We set the EOS logit to ∞ whenever fewer than 14 non empty lines have been generated.

Newline Boosting. We add a soft bias toward the newline token as lines grow longer. Letting ℓ denote tokens since the last newline:

$$\text{logit}(\text{newline}) += \begin{cases} 0 & \text{if } \ell \leq 10 \\ 1.5 + 0.8(\ell - 10) & \text{if } 10 < \ell \leq 18 \\ 1.5 + 0.8(\ell - 10) + 15.0 & \text{if } \ell > 18 \end{cases} \quad (2)$$

Repetition Penalty and Structural Stopping. A decoding time repetition penalty discourages looping, and generation terminates exactly at 14 non empty lines.

4.3 Best of k Candidate Reranking

Token level training optimizes cross entropy, but evaluation uses chrF, a sequence level metric. To bridge this gap, we generate k candidate sonnets per prompt using varied temperature and top p settings, score each with a structural heuristic $S(y)$, and select $y^* = \arg \max_{y_i} S(y_i)$. The scoring function rewards correct line count (14 lines), sonnet like line lengths (6-12 words), approximate rhyme consistency via suffix matching on the Shakespearean scheme, lexical diversity, and penalizes repeated lines and junk content. Post processing removes empty lines, duplicated prefixes, and truncates to exactly 14 lines.

4.4 Cloze Style Paraphrase Detection

Rather than attaching a learned two class linear head, we reformulate paraphrase detection as cloze style next token prediction. The input is formatted as a natural language prompt asking whether one sentence is a paraphrase of another, followed by "Answer yes or no." We extract the hidden state at the final non padding position, project into vocabulary space, and select only the logits for "no" (token ID 3919) and "yes" (token ID 8505):

$$\text{logits}_{\text{class}} = \text{vocab_logits}[:, [\text{ID}_{\text{no}}, \text{ID}_{\text{yes}}]] \quad (3)$$

This reformulation aligns fine tuning with GPT 2’s pretraining objective, makes the label space semantically meaningful, and avoids a randomly initialized classifier head.

4.5 Random Deletion Augmentation

To reduce overfitting to lexical overlap, we apply random word deletion during training. With probability 0.5, a sentence is left unmodified; otherwise, each word is independently deleted with probability p . Augmentation applies to only one side of each sentence pair to preserve semantic content.

4.6 SMART Inspired Regularization

We add a simplified L2 regularization term inspired by SMART :

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \sum_{\theta} \|\theta\|_2^2 \quad (4)$$

This encourages smaller parameter magnitudes to reduce overfitting during full model fine tuning. We note this is a lightweight approximation of the full SMART algorithm, which uses adversarial embedding perturbations.

5 Experiments

5.1 Data

For sonnet generation, we use the Shakespeare sonnet dataset (154 sonnets: 143 training, 12 test). During evaluation, the model receives the first 3 lines as a prompt and generates the remaining 11. For paraphrase detection, we use the Quora Question Pairs dataset with the provided train/dev/test splits.

5.2 Evaluation Metrics

Sonnet generation is evaluated using chrF [9], a character level n gram F score measuring similarity to reference poems, supplemented by qualitative analysis. Paraphrase detection is evaluated by accuracy and F1 on the development set.

5.3 Experimental Details

Sonnet Generation. GPT 2 (small) is fine tuned with AdamW (learning rate 5×10^{-5} , batch size 8, gradient clipping at norm 1.0) for 4 epochs. Generation uses nucleus sampling with temperature 0.9 and top p 0.9. Best of k uses $k = 20$ candidates with repetition penalty 1.2.

Paraphrase Detection. GPT 2 (small) with cloze style classification is trained with AdamW (batch size 8) for 10 epochs. We search over learning rates (5×10^{-6} , 1×10^{-5} , 3×10^{-5}) and dropout (0.1, 0.3). The deletion probability is $p = 0.1$ and the regularization coefficient is $\lambda = 10^4$. We organize experiments into three modes for ablation: baseline (cloze only), deletion (+ augmentation), and SMART (+ L2 regularization).

5.4 Results

5.4.1 Sonnet Generation

Table 1: Sonnet generation results (chrF). Higher is better.

Model	Test chrF
Baseline GPT 2 (nucleus)	29
+ Line aware decoding	34
+ Best of k reranking ($k=20$)	43

6 Analysis

6.1 Sonnet Generation Analysis

Structural Validity. The most significant improvement from line aware decoding is structural correctness. Baseline nucleus sampling frequently produces outputs with fewer than 14 lines due to premature EOS sampling, or lines of highly variable length. With EOS suppression and newline boosting, every generated sonnet contains exactly 14 non empty lines with more consistent lengths. Since a structurally invalid poem cannot meaningfully score well on chrF, this alone represents a substantial improvement.

Effect of Best of k Reranking. Reranking across diverse candidates generated under different sampling parameters provides further improvement. The scoring function rewards 14 line form most heavily, with secondary terms for line length, rhyme, diversity, and non repetition providing meaningful tiebreaking among structurally valid candidates.

Limitations of Approximate Rhyme. Our rhyme heuristic uses suffix matching on the last two characters of line final words, which is a weak proxy for phonological rhyme and misses many valid rhymes. A phonetic dictionary would improve this component but was beyond our current scope.

Qualitative Observations. Baseline outputs sometimes drift into repetitive or non poetic text. Line aware decoding eliminates egregious structural failures, and reranking improves lexical diversity. However, even the best candidates often lack deep thematic coherence, reflecting the fundamental difficulty of generating meaningful poetry from a small fine tuned model.

6.2 Paraphrase Detection Analysis

Cloze Style vs. Classification Head. The cloze style reformulation leverages GPT 2’s existing understanding of "yes" and "no" as affirmative and negative responses, avoiding a randomly initialized

linear head. However, performance may be sensitive to prompt wording; we used the format from the provided starter code throughout.

Effect of Random Deletion. Random deletion augmentation reduces reliance on superficial lexical overlap. By randomly removing words during training, we encourage predictions based on deeper semantic features. The augmentation is mild (applied to one side with probability 0.5, deletion rate p) to preserve meaning.

Effect of SMART Inspired Regularization. L2 regularization encourages smaller parameter magnitudes, reducing overfitting when fine tuning a large pretrained model on a downstream task where model capacity far exceeds what is needed.

Ablation Insights. The three mode design (baseline, deletion, SMART) isolates each component’s contribution: improvement from deletion suggests data perturbation reduces lexical overfitting, while further improvement from SMART suggests regularization provides additional stability.

Failure Modes. The cloze style approach can struggle with subtle semantic distinctions, since it encodes both sentences and their relationship into a single next token prediction. Very long sentence pairs may also exceed the effective context window.

7 Conclusion

We investigated techniques for improving GPT 2 fine tuning robustness on sonnet generation and paraphrase detection. For sonnet generation, we found that token level fine tuning on 143 sonnets with vanilla nucleus sampling is insufficient to produce structurally valid outputs, and demonstrated that line aware decoding with best of k reranking substantially improves structural correctness and chrF scores. For paraphrase detection, we explored cloze style classification aligned with GPT 2’s pretraining objective, and found that random deletion augmentation and L2 regularization improve robustness.

Across both tasks, our results suggest that careful attention to the training signal through decoding time constraints for generation and through data perturbation and regularization for classification can meaningfully improve fine tuning outcomes on limited data tasks. Future work could explore phonetic rhyme models, adversarial perturbations following the full SMART algorithm, or learned reward functions for candidate reranking.

Team Contributions

I was a team of one and got everything done myself

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] Jack Hopkins and Douwe Kiela. Automatically generating rhythmic verse with neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2017.
- [3] Yi Liao, Yasheng Wang, Qun Liu, and Xin Jiang. Gpt-based generation for classical chinese poetry. *arXiv preprint arXiv:1907.00151*, 2019.
- [4] 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.
- [5] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.
- [6] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33, 2020.

- [7] Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2021.
- [8] Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019.
- [9] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2015.