Task-Specific Optimizations for GPT-2

Stanford CS224N Default Project

Bodo Wirth
Department of Mathematics
Stanford University
bodow@stanford.edu

Thomas Deng
Department of Computer Science
Stanford University
tdeng23@stanford.edu

Abstract

Large language models excel at a wide range of NLP tasks, yet efficiently adapting them to specialized applications remains challenging. We investigate parameter-efficient fine-tuning via Low-Rank Adaptation (LoRA), to examine whether it can reduce training costs without sacrificing accuracy. We also explore transfer learning to determine if pretraining on semantically or stylistically relevant datasets improves downstream performance. For sonnet generation, we address the need for rigid formatting by introducing a line-by-line approach that enforces a 14-line structure, comparing its impact alongside top-k sampling and beam search. Our findings indicate that LoRA substantially lowers computational overhead while retaining strong results, whereas transfer learning yields limited quantitative gains, likely due to baseline saturation or inappropriate evaluation measures. Moreover, although strict line-by-line generation enforces structural consistency, maintaining more nuanced poetic elements such as rhyme remains difficult.

1 Key Information

- Mentor: None
- External Collaborators: None
- Sharing project: No

2 Introduction

Large language models are highly effective across a range of NLP tasks, but adapting them to specialized applications requires methods that balance performance, efficiency, and structural control. This paper explores approaches for improving paraphrase detection efficiency and accuracy. We assess whether LoRA provides a viable alternative to full fine-tuning while maintaining accuracy. For sonnet generation, we introduce a line-by-line structural enforcement method to ensure outputs conform to the 14-line format. Additionally, we analyze the impact of transfer learning and sampling strategies on model performance. By addressing these challenges, this work provides insights into fine-tuning efficiency and structure-aware generation, offering practical methods for adapting large language models to specialized tasks.

3 Related Work

Research on paraphrase detection has established foundational methods for identifying semantic similarity between text pairs. Chandra and Stefanus, in Experiments on Paraphrase Identification Using Quora Question Pairs Dataset [1], implemented a paraphrase detection model using a Bag of Words approach and a Count Vectorizer, achieving an accuracy of up to 97%. However, their method was based on BERT, rather than GPT-2, and did not explore more recent advancements in parameter-efficient fine-tuning such as PeFT (Parameter-Efficient Fine-Tuning) and LoRA (Low-Rank Adaptation).

For sonnet generation, Xiao Liting's work, William Wanna Shake Pear [2], employed a LSTM/RNN-based approach. While effective, this method lacks attention mechanisms and does not incorporate modern fine-tuning techniques or decoding strategies that could improve text fluency and structure, unlike our implementation.

One such modern decoding strategies that we implement is top-k sampling, a technique that improves the diversity and expressiveness of generated text by restricting token selection to the k most probable words [3].probability as the final output.

4 Approach

4.1 Baselines

Sentiment Analysis Baseline We used GPT-2 and performed finetuning on the full model and the last-linear layer with a batch size of 8 over 10 epochs. For the full model we used a learning rate of 1×10^{-5} with a hidden dropout probability of 0.5, while for the last-linear layer we used a learning rate of 1×10^{-3} with a hidden dropout probability of 0.3. The goal was to match the provided leaderboard baseline. Since this task was not a focus of our extensions, we do not report further results except to state that we matched what was required.

Paraphrase Detection Baseline We performed full fine-tuning of GPT-2 small and medium models for 5 epochs using the AdamW optimizer, a batch size of 8, and a learning rate of 1×10^{-5}. We opt for a lower number of epochs to allow for faster iteration during experimentation. Performance is evaluated based on dev accuracy.

Sonnet Generation Baseline We use GPT-2 small and large models, and performed full fine-tuning for 10 epochs using AdamW optimization, top-p sampling, and softmax temperature adjustments. No early stopping is implemented.

4.2 Parameter-Efficient Fine-Tuning (PEFT)

Given the computational cost of full fine-tuning, we apply Low-Rank Adaptation (LoRA) as a parameter-efficient alternative [4]. LoRA injects trainable low-rank matrices into transformer layers, enabling the model to learn task-specific information while preserving most pretrained weights. This reduces memory usage and speeds up training, making it particularly valuable for the large datasets used in paraphrase detection.

4.3 Transfer Learning

In an attempt to improve generalization to the test sets, we applied transfer learning by pretraining on related datasets before fine-tuning on our highly specific target tasks [5].

Paraphrase Detection We pretrain on the Stanford Natural Language Inference (SNLI) Corpus and the MultiNLI Corpus before fine-tuning [6] [7]. These datasets aim to improve the model's ability to recognize semantic similarity and entailment relationships, and thus build a better general understanding of the relationship between two sentences.

Sonnet Generation We use Amoretti – a series of sonnets that match the style and time of Shakespeare's – written by Edmund Spenser, as additional pretraining data [2]. These simply aim to increase the size of our training data and thus reduce the chance of the model from overfitting, allowing the model to learn more general rhyme schemes, vocabulary, and structure.

4.4 Sampling Strategies: Top-K and Beam Search

In response to extremely poor performance from Top-P sampling we explored two other strategies [3].

Top-K Sampling Top-K Sampling confines the model's token selection to the top K most probable tokens at each step, setting the rest to zero and redistributing the probabilities among the k. This approach reduces randomness by limiting the token pool, thereby avoiding unlikely or nonsensical words.

Beam Search Beam Search is a search algorithm that explores a graph by expanding the most promising nodes (tokens) within a set. It maintains multiple candidate sequences and selects the highest-probability output. By considering multiple hypotheses simultaneously, Beam Search aims to ensure that the generated sequence is globally optimal rather than being stuck in a local optimum.

4.5 Structural Enforcement (SE): Line-by-Line Generation

A key challenge in improving our CHRF scores for sonnet generation was ensuring adherence to the rigid structural format of Shakespeare's sonnets. We did this by employing an iterative line-by-line generation procedure that calls a generate_line function for each new line, designating the newline token as the stopping criterion. We repeated this process 14 times to ensure that exactly one line was generated per iteration. This method helped us achieve a higher CHRF score by enforcing more correct and consistent formatting across the sonnets.

4.6 Early Stopping

To prevent overfitting and improve model adaptability, we implemented early stopping based on validation performance. Training was halted when no improvement was observed in dev loss for a fixed number of consecutive epochs. This method helped reduce training times and avoid unnecessary parameter updates, particularly for sonnet generation, where overtraining led to memorization rather than generalization.

5 Experiments

5.1 Data

Paraphrase Detection We use the Quora Question Pairs dataset for paraphrase detection. Additionally, we experiment with transfer learning using the SNLI and MultiNLI datasets as pretraining sources as described in Section 4.3.

Sonnet Generation For sonnet generation, we use a dataset of 154 Shakespearean sonnets. In an aim to improve performance, we incorporate an extended dataset containing 89 additional sonnets from the Amoretti collection as described in Section 4.3.

5.2 Evaluation Method

Paraphrase Detection We evaluate models using accuracy on the dev and test sets. Dev accuracy was used to select the best-performing model for final test evaluation.

Sonnet Generation We use CHRF as our primary evaluation metric. It is rapidly calculable and aims to provide a basic measure of both lexical similarity and structural adherence in text generation [8].

5.3 Experimental Details

5.3.1 Paraphrase Detection

All paraphrase detection experiments use the AdamW optimizer and a batch size of 8 (except where noted). We made the following adjustments:

LoRA Fine-Tuning Applied LoRA with rank 8, dropout probability 0.1, alpha factor of 32, and a tuned learning rate of 1×10^{-4} after trials with various other parameters.

Transfer Learning Pretraining using SNLI and MultiNLI for 3 epochs, followed by either LoRA fine-tuning (with the same configuration as earlier) or full-model fine-tuning with a batch size of 10 to speed up training.

5.3.2 Sonnet Generation

All sonnet generation experiments use the AdamW optimizer for 10 epochs with a batch size of 2 and learning rate of 1×10^{-5} (except where noted). We made the following adjustments:

Top-K Sampling Implemented early stopping based on dev loss with a patience of 2. Used top-k sampling for output generation, with the default value set at 50.

Top-K Sampling with Grid Search Tuned learning rate, batch size, and patience parameters to improve dev loss. The optimal configuration for both models was a batch size of 4 and patience of 3, but the learning rate for the small model was best at 1^{-4}, while the large model had an optimal learning rate of 5×10^{-5}.

LoRA Fine-Tuning with Grid Search Applied LoRA with Top-K sampling with structural enforcements using the same configuration as described in paraphrase detection and again tuned learning rate, batch size, and patience parameters to improve dev loss. The optimal configuration for the small model was with a batch size of 2, patience of 2, and learning rate of 5×10^{-5}, while for the large model it was a batch size of 4, patience of 3, and learning rate of 1×10^{-4}.

Transfer Learning Pretrained on Amoretti sonnet dataset for 3 epochs before full model fine-tuning with top-k sampling and structural enforcements.

5.4 Results

5.4.1 Paraphrase Detection

Table 1: Paraphrase Detection Results for Small Models

Small Models	Dev Accuracy	Training Time (A100)
Baseline	**0.882**	4.5h
LoRA	0.879	**2.5h**
Transfer + LoRA	0.879	-
Transfer + Full FT	0.880	-

Table 2: Paraphrase Detection Results for Medium Models

Medium Models	Dev Accuracy	Training Time (A100)
Baseline	**0.904**	6h
LoRA	0.898	**4.5h**
Transfer + LoRA	0.899	-
Transfer + Full FT	0.895	-

Table 3: Paraphrase Detection Test Set Result (Best Dev Model Used)

Model	Test Accuracy
Medium (Baseline)	**0.900**

Naturally, the larger model consistently outperformed the smaller model, as it is able to capture more complex relationships between the sentence pairs and store more information within its model parameters. More interestingly, we see that LoRA fine-tuning dramatically reduced training time (by up to 45%) while maintaining strong accuracy, making it a viable alternative for resource-efficient training. On the other hand, it is surprising to note that transfer learning did not yield improvements to accuracy in any configuration. This suggests that the transfer data may not have generalized well enough to the task, or that the baseline models had already reached near-optimal performance, which is reinforced by seeing that the accuracy is close to the state of the art [1].

5.4.2 Sonnet Generation

Table 4: Sonnet Generation Results for Small Models

Small Models	Dev CHRF Score
Baseline	22.908
Top-K Sampling	19.305
SE + Top-K	28.263
SE + Beam Search	17.406
SE + Top-K + Grid Search	28.575
SE + LoRA + Grid Search	**28.824**
SE + Transfer Learning	26.723

Table 5: Sonnet Generation Results for Large Models

Large Models	Dev CHRF Score
Baseline	24.999
Top-K Sampling	19.889
SE + Top-K	**29.603**
SE + Beam Search	17.575
SE + Top-K + Grid Search	28.821
SE + LoRA + Grid Search	29.064
SE + Transfer Learning	27.902

Table 6: Sonnet Generation Test Set Result (Best Dev Model Used)

Test Set Model	Test CHRF Score
SE + Top-K (Large)	**43.046**

We once again note that across each task, the larger model outperformed the smaller one, which aligns with our expectation that it would be able to identify more nuanced features of the vocabulary or structure of Shakespeare's sonnets. Moreover, we see that while Top-K sampling on its own led to a decrease in model performance, when combined with the 14-line structural enforcement, it led to dramatic improvement in model performance, which indicates a potential issue in the evaluation metric that I will explore in more detail in the next section. This is in contrast to beam search sampling, which was the worst performing model and generated sonnets in which the same phrase was repeated 11 times[1]. Interestingly, the extent to which further changes affected performance differed for each model.

The larger models did not quantitatively benefit from either hyper parameter grid searches, LoRA, or transfer learning, though we note that the drop in performance due to LoRA was minimal and the training times were much faster. On the other hand, the smaller models saw improvement from the grid search and notably from the implementation of LoRA. This suggests somewhat unintuitively that the smaller GPT-2 is more prone to overfitting this small dataset than the larger one, and so benefited greater from training fewer parameters during finetuning.

[1] This is a common occurrence in beam search and though it would have likely been avoidable with further parameter tuning, we lacked the resources and compute for further exploration down this path [9].

6 Qualitative Analysis

For qualitative analysis, we focus on sonnet generation, as it provided greater variety and interpretability in model outputs.

6.1 CHRF Evaluation Metric

While CHRF scores provide a quantitative measure of text similarity, we wish to highlight notable deficiencies in its use as a metric in the quality of Shakespearean sonnet generation by examining two examples of the output of the large models (before structural enforcements).

Baseline (CHRF 24.99)

> My love is as a fever, longing still
> For that which longer nurseth the disease,
> Feeding on that which doth preserve the ill,
> madness's dear can sail creatures ineffective Educ 978 stopsreasonuttonppy inline
> And father gr Slater thing foreittered
> $ Frag Minediscicepsizing medicine,lake belongingpo Einstein mute. punish in thineifles;
> Aviation meltdown willfullydlarms rectangularquer example ventures
> Crown sits Clerkhoun intersections neglect Foo aunt/// Dup Junior May Sierra
> Calls to Spring hanging hidesPause somet Claw sterlingzhou.
> callaretz ignorance thanaley, some concentrations this deb board,
> paddingaces oral wills frequent vow. lives payslocal
> un Quebeceekshow lineleader from how deeply report juggling pearogenic goingess Vance TTCscoreedited Scores and I Words feather energy

Top-K (CHRF 19.889)

> My love is as a fever, longing still
> For that which longer nurseth the disease,
> Feeding on that which doth preserve the ill,
> And lives on no other than his own love.
> So thy sweet self, thy sweet breast,
> Thou art as the sun that no shadow doth cover,
> And thy love, like a jewel,
> Thou hast not so much the power of thy might
> But by a

While the baseline produces a completely incoherent and nonsensical output, the top-k implementation manages to generate a stylistically and semantically correct sonnet, failing only in its structure. Despite this clear difference in quality however, the baseline achieves a far higher CHRF score. This is likely because CHRF primarily measures character-level overlap rather than semantic coherence, making it insensitive to the presence of nonsensical or malformed text as long as it shares characters with the reference. The baseline's high score likely results from its inclusion of Shakespearean words or phrases, even if they are arranged incoherently, whereas the top-k implementation, despite producing a more appropriate output, suffers from lower lexical overlap. This highlights a fundamental limitation of CHRF in evaluating creative text generation.

6.2 Transfer Learning

Despite the transfer learning model not achieving the highest CHRF score, it likely produced the most faithful Shakespearean sonnets among the models evaluated. Below, we present an example output from the transfer learning model:

> O, from what power hast thou this powerful might
> With insufficiency my heart to sway?

To make me give the lie to my true sight,
If thou wouldst make me think I am some coward clown.
O, in your spite you do it spiteful spite;
In me thou maketh thy subject be my pride,
And mine own frailties thy crime confounds,
Or at best thy love being all, I am your slave.
O, therefore be as thou art, thou canst not mock me,
Nor mock me with my true heart’s shame, but dost offend
My true feelings, which are not offended by thee
When I am thy flatterer, thou being thy flatterer.
"What is your name?" I say, "O, ’tis my love, ’tis
"Rough, ’tis rougher than rough, ’tis ’cause you are."

Unlike the other models, it not only maintains a consistent archaic tone, but even achieves some rhyme in alternating lines, something neither the baseline nor the "best" top-k model was able to accomplish. The influence of transfer learning from Amoretti appears to have helped the model internalize the Shakespearean or early modern English poetic form, however, since CHRF does not capture these stylistic features or rhyme structure, this qualitative improvement is not reflected in its score.

7 Conclusion

To conclude, we explored task-specific optimizations for GPT-2 in paraphrase detection and sonnet generation, focusing on parameter-efficient fine-tuning, transfer learning, and structural enforcement. Our results demonstrate that LoRA effectively reduces computational costs while maintaining strong performance, making it a viable alternative to full fine-tuning. For sonnet generation, enforcing a strict 14-line structure significantly improved formatting adherence, though capturing deeper stylistic elements remained a challenge. While transfer learning failed to improve paraphrase detection accuracy, it introduced qualitative improvements in poetic style, but these gains were not reflected in CHRF scores, which underscores the limitations of current evaluation metrics.

A key limitation of our work is that structural enforcement alone does not ensure rhyme or meter, which are critical for accurate Shakespearean sonnet generation. Additionally, while LoRA improves efficiency, its trade-offs in final accuracy warrant further investigation. Future work could implement ReFT as an alternative method of parameter efficient finetuning and could focus on refining evaluation methods to better capture stylistic features, implementing explicit rhyme and meter constraints [10].

Team contributions

Bodo Wirth: Conducted baselines, implemented transfer learning for both models, and LoRA for paraphrase detection.
Thomas Deng: Extended sonnet generation to the two different sampling methods, obtained data sets for transfer learning, implemented sentiment analysis classifier, and conducted hyperparameter search.
All other work was shared between us (write-ups etc.).

References

[1] Andreas Chandra and Ruben Stefanus. Experiments on paraphrase identification using quora question pairs dataset. *arXiv preprint arXiv:2006.02648*, 2020.

[2] Liting Xiao. William-wanna-shake-pear. `https://github.com/litingxiao/William-wanna-shake-pear`, 2020.

[3] TiDB Team. Decoding methods compared: Top-k and other token selection techniques, 2024. Accessed: 2025-02-28.

[4] 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. *arXiv preprint arXiv:2106.09685*, 2021.

[5] Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. In Anoop Sarkar and Michael Strube, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[6] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[7] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference, 2018.

[8] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In Ondřej Bojar, Rajan Chatterjee, Christian Federmann, Barry Haddow, Chris Hokamp, Matthias Huck, Varvara Logacheva, and Pavel Pecina, editors, *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[9] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2020.

[10] Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. Reft: Representation finetuning for language models, 2024.