

Implementing and Fine-Tuning GPT2 for Sentiment Analysis, Paraphrase Detection, and Sonnet Generation

Stanford CS224N Default Project

Zhenghui Chen

Department of Computer Science
Stanford University
zhengh04@stanford.edu

Abstract

Large pretrained language models have shown strong capabilities across a wide range of natural language processing tasks. Full fine-tuning of large models is computationally expensive and prone to overfitting on small datasets, while task-specific architectures sacrifice generality. In this project, we specifically implement and fine-tune the GPT2 model, which is a decoder-only Transformer, and evaluate its ability to generalize across tasks like sentiment analysis, paraphrase detection, and sonnet generation. We first reproduced baseline results on sentiment classification benchmarks on the SST and CFIMBD datasets and explored last-layer and full-model finetuning to understand tradeoffs between efficiency and performance. Additionally, we extended the model to a cloze-style paraphrase detection task on the Quora dataset to evaluate if a prompt-based formulation can capture semantic similarity without adding task-specific classification layers. We then also extended the model to sonnet generation by training it on Shakespearean sonnets conditioned on the first three lines and evaluated its generated output by using the chrF metric. Our key contributions are two extensions on the paraphrase detection and sonnet generation tasks to improve their performance past the baseline. First, we applied Low-Rank Adaptation (LoRA) for the paraphrase detection task to reduce the number of trainable parameters, allowing us to train the gpt2-large model. We also introduced top-k sampling, temperature scaling, and early stopping to reduce incoherence and prevent overfitting on the small training set for the sonnet generation task. Through these, we achieved improvements on paraphrase detection, improving dev accuracy from 0.872 to 0.875 and test accuracy from 0.857 to 0.863. For sonnet generation, our extensions improved dev chrF from 40.815 to 40.983, though test chrF decreased from 41.935 to 40.389.

1 Key Information to include

- Mentor: Wei Liu
- External Collaborators (if you have any): No
- Sharing project: No
- Late Days (if you are using any): 1

Team member	Late days remaining
<Zhenghui Chen>	<1>
Total pooled	<1>
Team size	<1>
Late days applied for the report	<1>

2 Introduction

Models prior to GPT-2 for natural language processing relied primarily on supervised learning, where models were trained on task-specific labeled datasets for problems such as sentiment classification, translation, or answering questions. These methods relied heavily on human involvement and learned representations for a specific task, which limited their ability to generalize to other datasets. As a result, making progress across tasks often required making new architectures or collecting new labeled datasets for each problem. Radford et al. [1] challenged this paradigm by demonstrating that a decoder-only Transformer trained on massive amounts of unlabeled text with a next-token prediction objective could learn general-purpose language representations that transfer broadly across downstream tasks. This allowed for a single pretrained model to be adapted through finetuning for multiple tasks rather than making specific models for each task.

In this project, we implement GPT-2 from scratch and fine-tune it across three tasks: sentiment analysis, paraphrase detection, and sonnet generation. For sentiment analysis on the SST and CFIMDB datasets, we compare last-layer and full-model fine-tuning to understand the efficiency-performance tradeoff. For paraphrase detection on the Quora Question Pairs dataset, we adopt a cloze-style formulation, prompting the model to generate "yes" or "no" in response to whether two sentences are paraphrases. For sonnet generation, we fine-tune GPT-2 on Shakespearean sonnets and evaluate generated output using the chrF character-level n-gram metric. Beyond reproducing these baselines, we introduce two key extensions for the paraphrase detection and sonnet generation tasks. For paraphrase detection, we apply Low-Rank Adaptation [2], a parameter-efficient fine-tuning method that freezes the pretrained backbone and injects trainable low-rank adapter matrices into the attention layers. This reduces the trainable parameters, allowing us to fine-tune the larger gpt2-large model, avoiding much of the associated memory cost. Our hope with using a larger model was that the increased capacity of gpt2-large would allow it to capture more nuanced semantic relationships between sentence pairs, leading to better detection performance. This allowed us to improve the dev accuracy from 0.872 to 0.875 and test accuracy from 0.857 to 0.863. For sonnet generation, we improved the generation pipeline with top-k sampling and temperature scaling to reduce incoherence in generated text and introduced early stopping to prevent overfitting on the small training set, which improved the chrF score for the dev set from 40.815 to 40.983 but reduced the score for the test set from 41.935 to 40.389.

Our results demonstrate that parameter-efficient fine-tuning via LoRA provides consistent improvements on paraphrase detection by enabling a larger model without overfitting, while sonnet generation improvements highlight the sensitivity of small-data generation tasks to training dynamics and decoding strategy. Together, these findings underscore both the promise and the challenges of adapting general-purpose language models to specialized tasks.

3 Related Work

Radford et al. [1] make several key contributions to the study of language modeling and transfer learning in natural language processing. The main contribution is the demonstration of a large, decoder-only Transformer trained with unsupervised learning, achieving strong performance on a vast range of NLP tasks. This result challenged previous assumptions that task-specific architectures were required for effective task performance and laid the foundation for the shift in model focus for NLP. In addition to the change in models, this paper also changes the idea of how to model downstream NLP tasks. The main idea was to model these tasks as instances of conditional text generation, which allowed a single pretrained language model to perform different tasks in a zero-shot setting. This established a unified modeling framework in which the same architecture and learning objective are used across tasks, differing only in how inputs are formatted. Lastly, this paper provides empirical evidence that scaling model size and training set size leads to consistent improvements in both language modeling performance and downstream task accuracy, highlighting scale as an important factor in enabling unsupervised multitask learning. Together, these contributions create the foundation of a general-purpose model for NLP systems that our project directly builds upon.

While the one model paradigm established by Radford et al. is powerful, full fine-tuning of large models becomes increasingly expensive as model size grows. Hu et al. [2] propose Low-Rank Adaptation (LoRA) as a parameter-efficient alternative that addresses this challenge. The key insight behind LoRA is that changes to a model when adapting to a new task are inherently low-dimensional,

meaning only a small number of directions in the weight space actually matter for task performance. LoRA exploits this by representing weight updates as the product of two small matrices instead of updating the full weight matrix directly which can dramatically reduce the number of trainable parameters by 10,000 times and the GPU memory requirement by 3 times. Additionally, Hu et al. demonstrate empirically on RoBERTa, DeBERTa, GPT-2, and GPT-3 that LoRA matches or exceeds full fine-tuning performance across a wide range of tasks while training fewer parameters.

These papers together directly motivate our approach as we build on the GPT-2 architecture and paradigm established by Radford et al. and apply LoRA introduced by Hu et al, to enable parameter-efficient fine-tuning of the larger gpt2-large model for our paraphrase detection task. The combination of these two works allows us to explore the performance of a single pretrained model adapted efficiently across diverse NLP tasks ranging from classification to open-ended text generation.

4 Approach

Our approach builds on the GPT2 model and adapts it for the downstream tasks of sentiment analysis, paraphrase detection, and sonnet generation. We also leverage pretrained weights and evaluate frozen and fully fine-tuned settings to understand how well the model generalizes versus adapts to these tasks.

In terms of the model architecture, we followed the standard Transformer decoder design. Specifically, the model consists of token and positional embeddings, followed by a stack of Transformer blocks, each containing masked multi-head self-attention and position-wise feedforward layers with residual connections and layer normalization. Given an input sequence of tokens $x_1, x_2, \dots, x_T$, the model produces contextualized hidden representations $h_1, h_2, \dots, h_T$, where each h_t encodes information from all preceding tokens. For the downstream tasks, we pass the final hidden state corresponding to the last token in the sequence to a task-specific linear layer for classification. Additionally, we implemented the AdamW optimizer with decoupled weight decay and bias correction to ensure stable and efficient training.

To test the model on sentiment analysis, we trained and tested the model on the SST and CFIMBD datasets, which both contain movie reviews. To perform sentiment analysis, we applied dropout for regularization and passed the result through a linear projection layer to produce logits over the sentiment classes on the final hidden state. We then evaluated its performance using two training configurations: only training the last linear layer and training the entire model. This comparison allows us to weigh the effectiveness of pretrained features and the benefits of task-specific adaptation.

To test the model on paraphrase detection, we trained and tested the model on the Quora dataset. To perform paraphrase detection, we modeled the problem as a cloze-style conditional generation task. For each sentence pair, we construct a prompt of the form "Is s_1 a paraphrase of s_2 ? Answer yes or no:". The model is then tasked with predicting the next token conditioned on this prompt. More clearly, let $x_1, \dots, x_T$ denote the tokenized prompt. The model produces contextualized hidden states $h_1, \dots, h_T$, and we use the final hidden state h_T to compute logits over the vocabulary using the pretrained language modeling head. From the resulting distribution over the vocabulary, we extract the logits corresponding to the tokens "yes" and "no," and normalize them to obtain a binary prediction.

To test the model on sonnet generation, we fine-tuned GPT-2 on a corpus of 143 Shakespearean sonnets. Given a tokenized sonnet $x_1, \dots, x_T$, the model is trained to minimize the cross-entropy loss over next-token predictions across all positions. At inference time, the model is conditioned on the first three lines of a sonnet and tasked with generating the remaining lines on its own. We evaluate this generation quality using the chrF metric which measures character-level n-gram overlap between generated and reference sonnets.

In terms of baselines for our two tasks, we followed the provided baselines in the Default Project Instruction handout. For sentiment classification, we wanted 0.472 and 0.861, respectively, for the dev accuracy after training the last linear layer for SST and CFIMBD. After training the full model, we wanted a dev accuracy of 0.513 and 0.976, respectively, for SST and CFIMBD. For paraphrase detection, we wanted a score of 0.862 or more on the dev set.

To improve paraphrase detection beyond the baseline, we applied LoRA to enable fine-tuning of the larger gpt2-large model. With this, rather than updating all model parameters, the pretrained GPT-2

backbone is frozen and trainable low-rank adapter matrices are injected into the query, value, and output projection of each attention block. Only these adapter matrices and the classification head are updated during training which reduces the number of trainable parameters to under 0.2% of the total parameters. We use rank $r = 8$ and scaling factor $\alpha = 16$ following the recommendations of Hu et al.[2]. Our hope was that the increased capacity of gpt2-large would allow the model to capture more nuanced semantic relationships between sentence pairs, leading to better performance compared to the smaller gpt2 model used for the baseline.

To improve sonnet generation beyond the baseline, we made two changes to the training and inference pipeline. First, we introduced top-k sampling to replace the original sampling approach in the generate function. With this, at each generation step, we restrict the candidate tokens to the top 50 most likely tokens before sampling to prevent the model from sampling low-probability tokens that don't make sense. Secondly, we introduced early stopping based on training loss to prevent overfitting on the small 143-sonnet training set saving the best checkpoint across training.

5 Experiments

5.1 Data

For the sentiment analysis task, we used the Stanford Sentiment Treebank (SST) dataset and the CFIMDB dataset. The SST dataset consists of 11,855 single sentences from movie reviews split into 8,544 training examples, 1,101 dev examples, and 2,210 test examples. Each example has a label of negative, somewhat negative, neutral, somewhat positive, or positive. The CFIMDB dataset consists of 2,434 highly polar movie reviews split into 1,701 training examples, 245 dev examples, and 488 test examples. Each example has a binary label of negative or positive. For paraphrase detection, we used the Quora Question Pairs dataset, which consists of 400,000 question pairs with labels indicating whether the questions are paraphrases of one another. These examples are split into 141,506 training examples, 20,215 dev examples, and 40,431 test examples. For sonnet generation, we used a corpus of 154 Shakespearean sonnets, of which 143 are used for training. At inference time, the model is given the first three lines of each held-out sonnet as a prompt and tasked with generating the remaining lines to complete the full 14-line structure.

5.2 Evaluation method

For sentiment analysis, accuracy measures the proportion of correctly predicted sentiment labels across all classes. For paraphrase detection, accuracy is computed as the proportion of correct binary predictions on the Quora Question Pairs dataset. For sonnet generation, we use the chrF metric, which computes the F-score of character-level n-gram overlap between the generated sonnet and the reference Shakespeare sonnet.

5.3 Experimental details

All models are implemented following the GPT-2 architecture and initialized with pretrained weights. For sentiment classification, we experimented with two fine-tuning strategies: only training the last linear layer and training the entire model. All models were trained using the AdamW optimizer with a learning rate of 1×10^{-3} for frozen models and 1×10^{-5} for full fine-tuning. We used a batch size of 64 and trained for 5 epochs on both SST and CFIMDB.

For paraphrase detection, we reformulated the problem as a cloze-style generation task and trained the baseline GPT-2 model using the AdamW optimizer with a learning rate of 1×10^{-5} and a batch size of 8 for one epoch. For our LoRA extension, we used the gpt2-large model with rank $r = 8$ and scaling factor $\alpha = 16$, a batch size of 4, and trained for one epoch. All backbone parameters were frozen and only the injected adapter matrices and the classification head were updated during training.

For sonnet generation, we fine-tuned the gpt2 model using the AdamW optimizer with a learning rate of 1×10^{-5} and a batch size of 8 for up to 10 epochs. For our generation extension, we applied top- k sampling with $k = 40$ and temperature 0.9 at inference time. We also introduced early stopping with a patience of 3 epochs based on training loss to prevent overfitting on the small training set, saving the best checkpoint across all epochs.

5.4 Results

Task	Model	Dev	Test
SST	Last Linear Layer	0.468	0.462
SST	Full Fine-Tune	0.510	0.543
CFIMDB	Last Linear Layer	0.869	–
CFIMDB	Full Fine-Tune	0.984	–
Paraphrase	GPT-2 Baseline	0.872	0.857
Paraphrase	GPT-2 + LoRA (large)	0.875	0.863
Sonnet	GPT-2 Baseline	40.815	41.935
Sonnet	GPT-2 + Top-k + Early Stop	40.983	40.389

Table 1: Results across all tasks. Sentiment analysis results are reported on both dev and test sets for SST. CFIMDB results are reported on the dev set only. Paraphrase detection accuracy and sonnet chrF scores are reported on both dev and test sets.

For sentiment classification, full fine-tuning consistently outperforms training only the final linear layer. On SST, full fine-tuning improves accuracy from 0.468 to 0.510, and on CFIMDB it achieves a substantial improvement from 0.869 to 0.984. These results are as expected as full fine-tuning allows every layer of the model to update its parameters. This allows the model to better capture the patterns relevant to the downstream task. However, when only the final linear layer is trained, the model is forced to rely entirely on the pretrained representations, which were learned for a general language modeling objective rather than sentiment classification.

For paraphrase detection, our GPT-2 baseline achieves a dev accuracy of 0.872 and test accuracy of 0.857, exceeding the provided baseline target of 0.862. Our LoRA extension using gpt2-large further improves both dev accuracy to 0.875 and test accuracy to 0.863. While the improvement is modest, it is consistent across both dev and test, suggesting that the larger model capacity does provide a small but reliable benefit for capturing semantic similarity between sentence pairs.

For sonnet generation, our baseline achieves chrF scores of 40.815 on dev and 41.935 on test. After applying top-k sampling, lower temperature, and early stopping, the dev chrF improves slightly to 40.983. However, the test chrF decreases from 41.935 to 40.389. This divergence between dev and test performance suggests that our early stopping condition, which was based on training loss rather than a held-out validation loss, did not effectively prevent overfitting.

6 Analysis

Taking a deeper look at the results of our model, we analyze its behavior across all three tasks, seeing where it succeeds and fails. In terms of sentiment analysis, we can see the large performance gap between last-layer and full fine-tuning across both SST and CFIMBD reveals that GPT-2’s pretrained representations are not directly well-suited for sentiment classification without adaptation. When only the final linear layer is trained, the model is limited by the quality of frozen representations that were learned for next-token prediction rather than sentiment discrimination. Full fine-tuning allows every layer to adjust toward the task. This suggests that for classification tasks that require extracting specific semantic signals from text, the pretrained representations alone are insufficient, and task-specific adaptation throughout the entire model is important for strong performance.

In terms of paraphrase detection, our cloze-style formulation achieves a strong baseline performance of 0.872 dev accuracy, confirming that framing paraphrase detection as next-token prediction over "yes" and "no" tokens is an effective approach. Additionally, our LoRA extension with gpt2-large modestly improves both dev and test accuracy, suggesting that increased model capacity helps capture more nuanced semantic relationships. Additionally, it shows that LoRA is an effective way to improve model performance without the full memory cost of fine-tuning a larger model.

In terms of sonnet generation, our baseline achieves competitive chrF scores of 40.815 on dev and 41.935 on test, confirming that fine-tuning GPT-2 on a small corpus of Shakespearean sonnets is sufficient to learn basic stylistic patterns. Our top-k sampling and temperature scaling extension produces a modest improvement on dev chrF from 40.815 to 40.983, suggesting that restricting the candidate token set during generation reduces incoherence and produces output that more closely

resembles Shakespeare’s vocabulary and phrasing. However, the decrease in test chrF from 41.935 to 40.389 reveals that our early stopping criterion based on training loss was insufficient to select a checkpoint that generalizes robustly across different prompt styles. This suggests that with such a small training set of only 143 sonnets, the model is highly sensitive to both the specific checkpoint selected and the particular prompts seen at inference time.

7 Conclusion

In this project, we implemented and fine-tuned GPT-2 across three diverse NLP tasks: sentiment analysis, paraphrase detection, and sonnet generation. For sentiment analysis, we demonstrated that full fine-tuning consistently outperforms last-layer training on both SST and CFIMDB, confirming that task-specific adaptation throughout the entire model is crucial for ensuring optimal model performance. For paraphrase detection, we showed that framing the task as a cloze-style next-token prediction problem is an effective approach. Our LoRA extension further improved both dev and test accuracy by enabling fine-tuning of the larger gpt2-large model with a fraction of the trainable parameters, demonstrating that parameter-efficient fine-tuning is a practical and effective strategy for improving performance within memory constraints. For sonnet generation, we improved the generation pipeline through top-k sampling and temperature scaling, achieving a modest improvement in dev chrF, though the decrease in test chrF highlighted the challenge of generalizing on a very small training set. The primary limitation of our work is the small size of the sonnet training corpus, which makes the model highly sensitive to checkpoint selection and decoding hyperparameters.

Team contributions (Required for multi-person team)

Worked in a one person group.

References

- [1] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Technical Report*, 2019.
- [2] 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.