

Fine-tuning GPT-2 for Sentiment Analysis, Paraphrase Detection, and Sonnet Generation

Stanford CS224N {Default} Project

Liliana Santos-Deonizio
Department of Computer Science
Stanford University
lilianas@stanford.edu

Abstract

In 2019, the GPT-2 model was presented as an improvement on the GPT-1 model (Radford et al. 2019). With more parameters, more training data, GPT-2 showed that a model trained on a large amount of data with unsupervised training could later be used on different downstream tasks successfully. GPT-2 marked a change in how models with more training data and more parameters could be used for downstream tasks. In this project, I implement code to complete a GPT-2 model building off of code provided by the CS224N teaching team. Next, I fine-tune a GPT-2 for downstream tasks such as sentiment analysis, paraphrasing, and sonnet generation. Additionally, Parameter-Efficient Fine-Tuning (PEFT) is used to try to improve performance on sonnet generation. Results show successful performance on baselines for sentiment analysis and paraphrase detection. Training the GPT-2 medium model with Low-Rank Adaptation (LoRA) show strong baseline performance on only one epoch of training. The GPT-2 medium model succeeds in identifying positive and negative paraphrase pairs when there is high similarity or overlap in words between sentence pairs but struggles when there are dissimilar words with the same meaning. For the sonnet generation task, there is an increase in performance with larger GPT-2 models and LoRA. Ultimately, this work serves as another example of the ability to use GPT-2 for diverse downstream tasks.

1 Key Information to include

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

2 Introduction

The GPT-2 model, presented in 2019 (Radford et al. 2019), was a bigger model trained on a larger and more diverse dataset than its predecessor, GPT-1 (Radford, 2018). GPT-2 showed that a large model could be fine-tuned for varied downstream tasks. This impacted future work where models were used to fine-tune on context or task specific data. In this paper, I explore a few of these downstream applications such as sentiment analysis, paraphrase detection, and sonnet generation. Sentiment analysis can be helpful for analyzing data such as reviews, feedback, or internet forums. Paraphrase detection may be helpful for language models as a training task that can help a language model accurately check if one sentence, text, or document is actually paraphrasing another. Finally, sonnet generation enters a more creative space where the limits and possibilities of language model performance can be better gauged. Additionally, Parameter-Efficient Fine-Tuning (PEFT) techniques like Low-Rank Adaptation (LoRA) can support the training of models in general and for downstream tasks (Hu et al., 2021). In this project LoRA is used to fine-tune the base GPT-2 model used for paraphrase detection and sonnet generation.

3 Related Work

The GPT-2 model has masked multi-head self attention which uses masking so the model cannot look ahead at the next word for prediction which builds on the work of Vasmani et al. (2017) where they presented the importance and structure of attention. An Adam Optimizer is implemented which only uses first-order gradients, (Kingma and Ba, 2014), and weight decay regularization is used for the implementation (Loshchilov and Hutter, 2017). For training on a small dataset and preventing overfitting, early stopping is used in machine learning to stop training if validation loss is not improving (Raskutti, 2011).

Additionally, PEFT techniques can help train models with many parameters in more efficient ways. This can help train a much larger model with less computing and with less time overall. Techniques like LoRA can help researchers take advantage of the strengths of large models without running into computing issues and also help increase overall efficiency in how these language models can be trained for downstream tasks (Hu et al., 2021). For example, the LoRA technique uses a low rank representation of weight matrices to help reduce the number of parameters that need to be updated (Hu et al., 2021). Instead of updating the full weight matrix LoRA only updates two matrices A and B that are of lower rank.

4 Approach

The first part of this project includes a GPT-2 model that is implemented with starter code provided by the Stanford CS 224N teaching team. Using this GPT-2 model, I then fine-tune for a sentiment analysis task. In addition to sentiment analysis, the GPT-2 model is used to fine-tune for paraphrase detection and sonnet generation. For the sonnet generation, early stopping was implemented so that the model training would stop if validation loss was not improving (Raskutti, 2011). After two epochs, if the validation loss was not improving then a checkpoint for the model would be created and this model would be used to check for a validation chrF score as well. In order to improve performance, LoRA was implemented which allowed the use of GPT-2 medium for the paraphrase detection task and GPT-2 medium and large for the sonnet generation task. Finally, top-k sampling is also used for the sonnet generation task as an alternative to top-p sampling.

5 Experiments

5.1 Data

For the sentiment analysis tasks, two datasets were used. The Stanford Sentiment Treebank5 which has 11,855 single sentences from movie reviews and CFIMDB dataset which has of 2,434 highly polar movie reviews. The data include reviews of various movies. For the Treebank dataset the labels include negative, somewhat negative, neutral, somewhat positive, or positive. For the CFIMDB data the labels are only negative or positive. These datasets have ground truth or gold labels to compare to as well. For paraphrase detection a Quora dataset is used which includes • 141,506 examples for training, 20,215 examples for a development set and 40,431 for a test set. The data include pairs of questions that are labeled indicating whether the pairs are paraphrases of one another or not. Finally, for the sonnet generation task sonnets from Shakespeare are used to form a training set of 143 sonnets and a test set of 12 sonnets. Additionally, I used 12 sonnets from the training set to form a validation set.

5.2 Evaluation method

Evaluation is based on accuracy for the sentiment analysis and paraphrase detection tasks. Accuracy is a helpful metric here since the Quora, SST, and CFIMDB datasets consist of labeled data so we can compare to "gold" labels or a ground truth. With the sonnet generation task we do not use accuracy rather, we use a chrF metric which is similar to the n-gram metric that compares the generated sonnet text to the original sonnet endings (Popović, 2015). This metric helps gauge the similarity between the generated sonnet and the original text as a measure of how well the model was able to follow Shakespeare's writing style.

5.3 Sentiment Analysis and Paraphrase Detection Experiments

Model configurations for sentiment analysis and paraphrase detection experiments were as shown in Figure 2. For the paraphrase detection, I tried to run with 8, 16, and then 32 batch-size. With a batch size of 8 and 16 the training took too much time (about one percent of an epoch completed each minute) so I moved to a batch size of 32. This batch size was the limit of what I could do with my GPU. I also encountered issues with the labels so I had to trouble shoot and run the experiment multiple times. On the final run I only did 3 epochs as each epoch took over an hour to run. Finally, I experimented with applying LoRA to the GPT-2 medium model and training this model for the paraphrase detection task. Though I was only able to run for one epoch, the GPT-2 medium model with LoRA showed a positive performance that was close to the baseline.

Figure 1: Sentiment Analysis and Paraphrase Detection Experiment Results

Task	Dataset	Layer	Batch Size	Learning Rate	Hidden Dropout Probability	Epochs	Dev Acc	Time
Sentiment	SST	Last linear layer	8	1e-3	0.3	10	0.451	~10min
Sentiment	CFIMDB	Last linear layer	64	1e-3	0.3	10	0.829	~10 mins
Sentiment	SST	Full model	8	1e-5	0.3	10	0.518	~18 mins
Sentiment	CFIMDB	Full model	64	1e-5	0.3	10	0.980	~26 mins
Paraphrase	Quora	Full model	32	1e-5	0.3	3	0.884	~4 hours
Paraphrase (LoRA + GPT-2 medium)	Quora	Full model	8	2e-4	0.3	1	0.867	~2 hours and 52 minutes

5.4 Sonnet Generation Experiments

For the sonnet generation experiments the configurations and results are shown in Figure 3. Initially, I tried to run the sonnet generation on the baseline GPT-2 model that was trained on paraphrase detection. As aforementioned, this include an early stopping criteria and the use of a validation set of 12 sonnets. With only three epochs of training the sonnet generation task achieved a chrF score of about 36. One trained for 10 epochs the outcome was 38.40 on the validation set. Next, I experimented with using LoRA and GPT-2 medium and large in the training. For the larger models I adjust the batch size, learning rate, dropout, and number of epochs to make sure I did not have a memory crash. Finally, the last experiments were done with the GPT-2 large model. After this, I changed the top-p sampling to top-k to see if this would help generate better sonnets (Zaharov, 2025). Additionally, I changed some of the hyperparameters to 1) increase the rank for the LoRA implementation, and 2) change the temperature to get more genre specific tokens. Finally, I experimented with increasing rank for the LoRA implementation to see if this improved performance on the sonnet generation task.

5.5 Results

The results for sentiment analysis were close to the provided baselines (Last Linear Layer for SST: Dev Accuracy: 0.462, Full Model for SST: Dev Accuracy: 0.513, Last Linear Layer for CFIMDB: Dev Accuracy: 0.861, Full Model for CFIMDB: Dev Accuracy: 0.976). Additionally, the results for the paraphrase detection experiment was a bit above the baseline. I was surprised by the performance on the paraphrase detection given that I only trained for 3 epochs and had such a hard time getting the development accuracy measure to work. I believe this is showing how quickly the model can

learn even over only a few epochs. To this point, when training the GPT-2 medium model with LoRA on the paraphrase detection task for only one epoch showed a strong baseline performance. On the other hand, the sonnet generation task was more difficult to improve performance for. Running the chrF score on a validation set for a baseline showed a performance of 38.40 and by running GPT-2 large with LoRA that performance increased to 42.212 after increasing the rank. Further, when I experimented with changing top-p to top-k sampling the performance did not change very much on the validation set.

The leaderboard test results were as follows: paraphrase detection: 0.870, sonnet generation: 42.212.

Figure 2: Sonnet Generation Experiment Results

Task	Batch Size	Learning Rate	Hidden Dropout Probability	Epochs	Validation chrF	Time
Sonnet Generation (Baseline)	32	1e-5	0.3	10	38.40	7min 14s
Sonnet Generation (LoRA +GPT-2 Medium) Default rank =8, alpha =16	4	2e-4	0.05	1	35.06	2min 53s
Sonnet Generation (LoRA + GPT-2 medium) Default rank =8, alpha =16	4	2e-4	0.05	10	41.54	20min 7s
Sonnet Generation (LoRA + GPT-2 large) Default rank =8, alpha =16	4	2e-4	0.05	3	41.54	13m58.52 7s
Sonnet Generation (LoRA + GPT-2 large) Default rank =8, alpha =16	4	2e-4	0.05	10	41.67	13m58.52 7s

6 Analysis

Table 1: Paraphrase Detection Confusion Matrix (Dev)

True \ Pred	Pred 0 (No)	Pred 1 (Yes)
True 0 (Not paraphrase)	21273 (TN)	4264 (FP)
True 1 (Paraphrase)	1131 (FN)	13761 (TP)

6.1 Paraphrase Detection

Looking at the results qualitatively for the paraphrase detection some patterns begin to emerge. As shown in Table 1, the majority of the errors made by the model were false positives. For these false positives examples the sentences seemed to have similar words or phrases but hold different meanings. This is shown in Table 2 where the GPT-2 medium model annotated "how can i kill cockroaches?" as a paraphrase for "should i kill that cockroach?" These sentences hold a different meaning because of the words "how" and "should" but they have many other words in common. Conversely, for the false negative examples it seems the GPT-2 medium model had difficulty recognizing a paraphrase when different words were used that related to the original question. For example, Table 2 shows a question pair where instead of "offerings" the second question says, "clothes and ornaments." Since these words are dissimilar the model incorrectly does not mark the questions as a paraphrase. The

model seemed to succeed when a paraphrase pair had similar words and when a non-paraphrase pair had very dissimilar words or less overlap.

Table 2: Paraphrase Detection Examples by Category

Type	Question 1	Question 2
False Pos	should i kill that cockroach ?	how can i kill cockroaches ?
False Pos	why do quorans answer questions that are already answered ?	why do quorans downvote questions they cannot answer ?
False Neg	what is shaoxing vinegar ?	what is shaoxing vinegar used for ?
False Neg	what happens to offerings which are given to gods or goddesses in the temple ?	what happens to the clothes and ornaments given to the gods in temples ?
True Pos	is it biologically good or bad to marry other caste ?	is it good or bad to marry other caste ?
True Pos	what is addiction ?	what exactly is addiction ?
True Neg	what are some great ways to generate passive income in india ?	can a full time employee be successful in investing on stock markets and other potential investments to generate passive income ?
True Neg	if i am on private on twitter and i favorite a tweet will it be seen/notified by the person i favorited ?	how can i know how much favorites i have on twitter ?

6.2 Sonnet Generation

In the initial epochs of training the baseline GPT-2 model qualitative review of the sonnets generated showed there were many phrases that did not follow the sonnet style. Phrases like "Getty images" and page numbers were in the generated sonnets. Once I moved on to experimenting with GPT-2 medium, I noticed that qualitatively these sonnets looked better (less outwardly random phrases, no page numbers or modern phrases like "Getty Images") but they still needed some refinement to match the exact style required. In the earlier epochs of training, there were still a majority of phrases that did not make sense and some nonsensical words like "542to" and "pl." However, towards the end of training, the sonnets appeared more cohesive overall with less words that stood out. The sonnets still lacked a complete narrative with each line seeming to stand on its own in terms of meaning. Using the GPT-2 medium model seemed to boost the chrF score for the validation set by about six points. Finally, the last experiment with GPT-2 large was first done with 3 epochs then 10 epochs. Training for 10 epochs showed minimal change on the validation chrF score. This could be due to the training corpus is small and the large model is already able to perform significantly well on a few epochs.

Next, I added a top k sampling which will choose the k most likely tokens as possible next tokens. Additionally, I experimented with the temperature hyperparameter to see if this would also help the sonnet generation be more conservative in generating next tokens. From a qualitative analysis, the top-k and temperature change changed the generated sonnets even in earlier epochs. There were no visible nonsensical words and the sentences seemed much more cohesive. However, there was more repetition of words or phrases. Finally, I modified the rank levels for LoRA and this showed some improvement. With top-k sampling and increase in rank, the test performance was able to increase to a score of 42.212. This is likely due to the increase in rank allowing for a more complex representation of the full weight matrix and giving the model more information to change performance.

7 Conclusion

In this project I used a GPT-2 model to train for downstream tasks including sentiment analysis, paraphrase detection, and sonnet generation. Across these diverse tasks, the GPT-2 model had successful baseline performance (as compared to baselines provided by the CS224N teaching team). Additionally, for sonnet generation, the GPT-2 model was able to produce sonnets that were qualitatively similar to the Shakespeare sonnets. The chrF scores showed a test performance of 42.212 which was an improvement from the baseline. Finally, using LoRA for both the paraphrase detection

Figure 3: Sonnet Generation Top-K Sampling

Task	Batch Size	LR	Dropout	Epochs	Top-k	Temp	Rank	Val chrF	Time
Sonnet (LoRA + GPT-2 large) alpha = 16	4	2e-4	0.05	10	40	.8	16	41.32	26m 18.779s
Sonnet (LoRA + GPT-2 large) alpha = 16	4	2e-4	0.05	10	40	.7	16	41.32	7min 14sec
Sonnet (LoRA + GPT-2 large) alpha = 32	4	2e-4	0.05	10	40	.7	32	41.801	29m 31.073s
Sonnet (LoRA + GPT-2 large) alpha = 32	4	2e-4	0.05	10	40	.7	64	40.975	25m 58.203s
Sonnet (LoRA + GPT-2 large) alpha = 32	4	2e-4	0.05	10	40	.7	128	40.983	30m 22.379s

and sonnet generation tasks showed that this PEFT technique could be successful in training for downstream tasks. For the paraphrase detection, LoRA was used with GPT-2 medium which on only one epoch of training achieved a performance close to the provided baseline. Using LoRA with GPT-2 large for sonnet generation showed an increase in performance from the original sonnet baseline on GPT-2 small. These experiments show that even with few epochs of training and small datasets, fine-tuning models for downstream tasks can be a strong strategy.

7.1 Limitations

I was not able to run further experiments across all the downstream tasks, primarily the paraphrase detection. The paraphrase detection task took the most time and compute even with LoRA due to the large dataset so I was only able to run LoRA with the GPT-2 medium model. Additionally, I had constraints in terms of available funds for computing so that determined the GPU and computing time limits for the project. For future work, this study could extend to include domain specific tasks. For example, in the field of education there has been research showing how language models can be used to annotate for academic talk moves during conversations.

7.2 Ethical Considerations

All the datasets used were available for the public to use. The Shakespeare sonnets were also part of the public domain and available to use to train on. However, there has been recent push back against using authors' work without their explicit consent especially for modern writing. This is something to keep in mind for training language models as much of the data these models were trained on was taken from the internet without explicit consent from some authors (Pontefract, 2025).

Team contributions (Required for multi-person team)

I was a one person team so I completed all the work for the project.

References

Alec Radford. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019.

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, pages 5998–6008, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

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.

Ilya Loshchilov and Frank Hutter. arXiv:1711.05101, 2017. Decoupled weight decay regularization. arXiv preprint

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.

Pontefract, Dan. (2025). Authors challenge meta’s use of their books for training AI. url=<https://www.forbes.com/sites/danpontefract/2025/03/25/authors-challenge-metas-use-of-their-books-for-training-ai/>.

Raskutti, G., Wainwright, M. J., Yu, B. (2011). Early stopping for non-parametric regression: An optimal data-dependent stopping rule. 2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference On, 1318–1325. <https://doi.org/10.1109/Allerton.2011.6120320>

Zaharov, N. (2025, May 30). Understanding Top-K and Top-P in Prompt Engineering: Medium. <https://medium.com/@8926581/understanding-top-k-and-top-p-in-prompt-engineering-00a3b93dcd40>