

Fine-Tuning, Alignment, and Efficient Adaptation of GPT-2 for NLP Downstream Tasks

Stanford CS224N {Default} Project

Ahmed Khidre
Department of Computer Science
Stanford University
akhidre@stanford.edu

Izhan Hamza
Department of Computer Science
Stanford University
izhamza@stanford.edu

Patrick Wang
Department of Computer Science
Stanford University
patwang@stanford.edu

Abstract

Fine-tuning large language models is essential for adapting overparameterized pretrained models to specific downstream tasks. However, it presents two central challenges: aligning model behavior with human preferences and managing computational cost. In this project, we fine-tune GPT-2 for sentiment classification, paraphrase detection, and Shakespearean sonnet generation, and then extend these baselines with Direct Preference Optimization (DPO) and Low-Rank Adaptation (LoRA) to improve alignment and efficiency, respectively. On sentiment classification tasks (SST and CFIMDB), our fine-tuned GPT-2 models achieved competitive accuracy, validating the correctness of our training pipeline. DPO improves sonnet generation quality, increasing CHRF from 42.50 to 42.75, while also enhancing stylistic alignment with Shakespearean language. LoRA reduces the number of trainable parameters by 99.8% and cuts training time by 30–45% on paraphrase detection and sentiment classification tasks. Our final models achieve 0.901 test accuracy on the paraphrase leaderboard and 41.877 CHRF on the sonnet leaderboard. These results demonstrate that alignment methods and parameter-efficient adaptation can improve model quality and training efficiency, respectively.

1 Key Information to include

- Mentor: David Anugraha
- External Collaborators (if you have any): N/A
- Sharing project: N/A
- Late Days (if you are using any): 1

Team member	Late days remaining
Izhan Hamza	1
Patrick Wang	0
Ahmed Khidre	3
Total pooled	4
Team size	3
Late days applied for the report	1

2 Introduction

Large pretrained language models such as GPT-2 and GPT-3 have demonstrated strong capabilities across a wide range of natural language processing tasks [1, 2]. However, adapting pretrained models to specific applications typically requires task-specific fine-tuning. While fine-tuning improves

performance, it introduces two key challenges: aligning model outputs with human preferences and managing the computational cost of updating large models.

A widely adopted paradigm for aligning language models is Reinforcement Learning with Human Feedback (RLHF)[3]. While RLHF has been successful in aligning modern language models, it introduces significant complexity. The approach requires training multiple models, sampling from the policy during optimization, maintaining a reward model, and carefully tuning reinforcement learning hyperparameters.

In parallel, the increasing size of modern language models has made fine-tuning progressively more computationally expensive. Training and adapting large models often requires substantial resources, including significant GPU memory, long training times, and high storage costs.

In this work, we investigate two complementary approaches to address these challenges: Direct Preference Optimization (DPO)[4] for preference-based alignment and Low-Rank Adaptation (LoRA)[5] for parameter-efficient fine-tuning. DPO reformulates preference learning as a direct optimization objective, eliminating the need for reward modeling and reinforcement learning[4]. Second, we employ LoRA, a parameter-efficient fine-tuning technique that significantly reduces the number of trainable parameters by introducing low-rank updates to pretrained weight matrices[5].

We evaluate these methods within the CS224N default project framework using GPT-2[1] on the standard CS224N tasks of sentiment classification, paraphrase detection, and sonnet generation.

3 Related Work

Reinforcement Learning from Human Feedback (RLHF) aligns language models using a multi-stage training pipeline consisting of supervised fine-tuning, reward model training from human preference comparisons, and policy optimization using reinforcement learning algorithms such as Proximal Policy Optimization (PPO) [6]. During training, a KL-divergence penalty constrains the updated policy to remain close to the pretrained model. RLHF has been widely adopted in systems such as InstructGPT and ChatGPT [3]. However, this approach introduces significant complexity due to the need for separate reward models and careful tuning of reinforcement learning hyperparameters.

To simplify alignment pipelines, Direct Preference Optimization (DPO) [4] directly optimizes language models using preference data without explicit reinforcement learning. DPO reformulates the RLHF objective as a closed-form classification loss over preferred and dispreferred responses while retaining implicit KL regularization with respect to a reference model. Empirically, DPO achieves performance comparable to or better than RLHF.

Several extensions to preference-based optimization have also been proposed. Group Relative Policy Optimization (GRPO) [7] introduces group-level comparisons to stabilize preference learning, while Direct Alignment Preference Optimization (DAPO) [8] incorporates richer alignment signals beyond simple pairwise comparisons. These methods typically introduce additional sampling or grouping mechanisms that increase training complexity.

Parameter-efficient fine-tuning methods have also been widely explored for adapting large language models to downstream tasks. Adapter modules insert small bottleneck layers into each transformer block while freezing the base model weights [9], while prefix tuning learns trainable prefix tokens that modify the attention mechanism without updating model parameters [10]. However, adapter modules introduce additional computational overhead and increase inference latency, while prefix tuning can be challenging to optimize and often exhibits non-monotonic performance behavior.

Aghajanyan et al. [11] showed that neural networks often operate in low intrinsic dimensions, suggesting that task-specific updates may lie in a low-dimensional subspace. Building on this insight, Hu et al. [5] proposed LoRA, which freezes pretrained model weights and introduces trainable low-rank matrices to represent weight updates. LoRA significantly reduces the number of trainable parameters and memory requirements while achieving performance comparable to full fine-tuning.

4 Approach

4.1 Baseline GPT-2 Fine-Tuning

Our starting point is the GPT-2 baseline [1] provided in the CS224N default project. GPT-2 is an autoregressive language model trained using a next-token prediction objective. We adapt the pretrained model to downstream tasks through supervised fine-tuning.

For sentiment analysis, a separate classification head is applied to the final hidden representation to predict sentiment labels. For paraphrase detection on the Quora Question Pairs (QQP) dataset[12], the two input sentences are concatenated with a separator token and fed into GPT-2. The final

hidden representation is then passed to a classification head to predict whether the two sentences are paraphrases.

For the Shakespearean sonnet generation task, GPT-2 is fine-tuned as a conditional language model. Given a prompt consisting of the initial lines of a sonnet, the model generates the remaining lines autoregressively. The model is trained using the standard language modeling cross-entropy loss over the target tokens.

These supervised fine-tuned models **serve as the baseline** systems for our experiments.

4.2 LoRA

To reduce the computational cost of fine-tuning GPT-2, we adopt Low-Rank Adaptation (LoRA) [5], a parameter-efficient fine-tuning technique. Instead of updating the full set of model parameters, LoRA introduces trainable low-rank matrices that modify the original weight updates while keeping the pretrained weights frozen. An overview of our LoRA module is present in Figure 1.

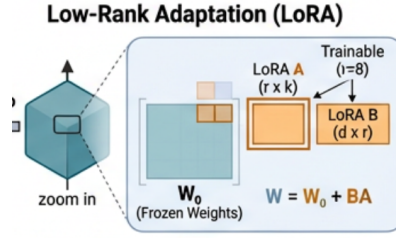


Figure 1: Low-Rank Adaptation, generated via nano-Banana

In our implementation, LoRA is applied to the query and value projection matrices of the self-attention mechanism in GPT-2. Specifically, we augment the query weight matrix W_Q and value weight matrix W_V in each transformer attention block with learnable low-rank updates. For a pretrained weight matrix

$$W_0 \in \mathbb{R}^{d \times k},$$

LoRA parameterizes the weight update as

$$W = W_0 + \Delta W = W_0 + BA$$

where

$$B \in \mathbb{R}^{d \times r}, \quad A \in \mathbb{R}^{r \times k},$$

and r is a small rank such that $r \ll \min(d, k)$. During training, the original pretrained weights W_0 remain frozen, and only the low-rank matrices A and B are optimized. LoRA is applied to both sentiment classification and paraphrase detection tasks, enabling faster training and reducing memory consumption compared to full fine-tuning.

4.3 Direct Preference Optimization (DPO)

To improve alignment for sonnet generation, we apply Direct Preference Optimization (DPO) [4].

DPO operates on preference triples (x, y_w, y_l) , where x is a prompt, y_w is the preferred response, and y_l is dispreferred. The objective increases likelihood of preferred responses while remaining close to a reference model:

$$\mathcal{L}_{DPO} = -\mathbb{E}_{(x, y_w, y_l)} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) \right]$$

where π_θ is the trainable policy, π_{ref} is the frozen supervised fine-tuned model, and β controls KL regularization. Overview of DPO workflow is shown in Figure 2

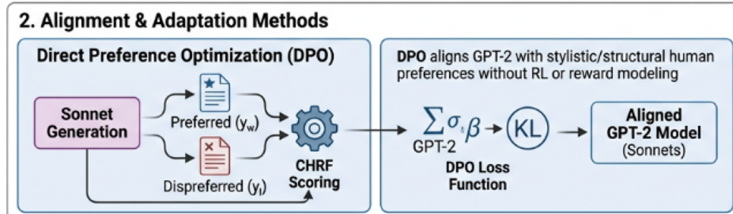


Figure 2: Direct Preference Optimization, generated via nano-Banana

5 Experiments

5.1 Data

We evaluate on three CS224N tasks: sentiment classification (SST[13], CFIMDB[14]), paraphrase detection (QQP), and Shakespearean sonnet generation.

For paraphrase detection, we use the Quora Question Pairs dataset [12], containing 283,003 training pairs and 40,429 development pairs annotated for semantic equivalence.

The Shakespearean sonnet dataset provides 131 training examples and 12 held-out examples, each consisting of a prompt (first three lines) and target continuation (remaining lines).

Since DPO requires preference data, we construct synthetic pairs for sonnet generation from scratch. For each prompt, we generate multiple continuations using an external LLM and compute CHRF scores against the ground-truth. The ground-truth serves as the preferred response y_w , while the lowest-scoring generated continuation is selected as the dispreferred response y_l , representing semantic drift from Shakespearean style. Table 9 in Appendix illustrates this ranking procedure. This automatic procedure allows us to construct a preference dataset without requiring manual human annotations.

5.2 Evaluation Method

5.2.1 Sentiment Classification and Paraphrase Detection Evaluation Method

For the sentiment classification and paraphrase detection task, model performance is evaluated using classification accuracy on the validation and test sets.

5.2.2 LoRA Evaluation Method

Computational cost is assessed using number of trainable parameters between LoRA and baseline models. While, model efficiency is assessed by comparing training time and evaluation time.

5.2.3 Sonnet Generation Evaluation Method

For the sonnet generation task, we evaluate model quality using the CHRF metric [15]. CHRF computes character-level n-gram overlap between generated text and reference outputs, making it suitable for evaluating poetry and stylistically constrained text generation where token-level metrics may be overly sensitive to small variations. In addition to automatic metrics, we qualitatively inspect generated sonnets to assess stylistic properties such as rhyme structure, coherence, and fluency.

5.3 Experimental Details

All experiments are conducted using the GPT-2 architecture provided in the CS224N default project codebase. The pretrained GPT-2 model is fine-tuned separately for each task.

5.3.1 Sentiment Classification Experimental Details

For sentiment classification, we performed hyperparameter tuning for four configurations: SST Full Mode, SST Finetune Mode, CFIMDB Full Mode, and CFIMDB Finetune Mode. All models were trained for 10 epochs using the AdamW optimizer, with final configurations selected based on the highest development accuracy.

We conducted grid searches over learning rate and dropout values to identify effective hyperparameters for each configuration. Table 1 summarizes the selected settings.

Table 1: Optimal hyperparameters for sentiment classification configurations.

Configuration	Learning Rate	Dropout
SST Full Mode	1e-4	0.2
SST Finetune Mode	2e-3	0.09
CFIMDB Full Mode	1e-5	0.5
CFIMDB Finetune Mode	2e-3	0.05

5.3.2 Paraphrase Detection Experimental Details

We tuned the paraphrase detection model using a learning rate sweep and the learning rate scaling law [2]. Models were trained using the AdamW optimizer [16] with data parallelism across $4 \times \text{H200}$ GPUs to accelerate training.

We first conducted a learning rate sweep for GPT-2 and identified **5e-5** as the optimal learning rate, achieving a development accuracy of 0.897. Using the learning rate scaling law, we then centered the search for GPT-2-medium around this value.

As shown in Table 2, the optimal learning rate for GPT-2-medium is **6e-5**, achieving a training accuracy of 0.987 and a development accuracy of 0.904 after 5 epochs. Additional learning rate sweep results and training curves are provided in Appendix B.

Model	Learning Rates	Batch Size	Hardware	Epochs
GPT-2 Medium	3e-5, 4e-5, 4.5e-5, 5e-5, 5.5e-5, 6e-5 , 7e-5	512	4 × H200	5

Table 2: Learning rate sweep for GPT-2-medium.

5.3.3 LoRA Experimental Details in Sentiment Classification and Paraphrase Detection

In the sentiment classification and paraphrase detection tasks we applied low-rank adaptation to the query and value projection matrices of the transformer attention layers while freezing the base GPT-2 weights.

In this setup, only the low-rank adapter matrices are updated during training. Several hyperparameters were selected and tuned as needed, including the rank (r), scaling factor (α), learning rate, and dropout rate.

The primary goal was to achieve performance comparable to the baseline model while training far fewer parameters and with improved efficiency. For our default LoRA configuration, we used rank = 8, α = 8, learning rate = 1e-5, and dropout = 0.05 and applied LoRA to the query and key projection matrices. Our choices of default were based on what has been proven to work well [5]. In future work we also hope to implement LoRA on the Multi-Layer Perceptron layers.

5.3.4 DPO Experimental Details for Sonnet Generation

For DPO training, the policy model is initialized from the supervised fine-tuned (SFT) checkpoint, while a frozen copy of the SFT model is used as the reference policy. Training is performed on the automatically constructed preference triples (x, y_w, y_l) described in Section 5.1, using the DPO objective introduced in Section 4.2.

We tune several hyperparameters that influence both training stability and generation quality. In particular, we tune the KL regularization coefficient β , the learning rate for the policy model, and the total number of training epochs. During inference, we also tune decoding parameters including top- k sampling and temperature.

Hyperparameters are selected using a grid search over predefined ranges. For each configuration, the model is trained on the preference dataset and evaluated on the development set using the CHRF metric. The final reported results correspond to the configuration that achieves the highest validation CHRF score.

The hyperparameter search ranges used in our experiments are summarized in Table 3.

LR	DPO β (KL coefficient)	Epochs	Top- k sampling	Temperature
$\{1e^{-5}, 5e^{-6}, 1e^{-6}, 5e^{-7}\}$	$\{0.001, 0.005, 0.01\}$	$\{4, 6, 8, 10\}$	$\{0.9, 0.95, 0.97\}$	$\{0.9, 0.95, 1, 1.05, 1.1\}$

Table 3: Hyperparameter search space for DPO training and decoding.

5.4 Results

5.4.1 Sentiment Classification Results

We conducted sentiment classification on the SST [13] and CFIMDB [14] datasets as sanity checks. Table 4 summarizes the best dev accuracy for each configuration after hyperparameter search.

Table 4: Sentiment classification results on SST and CFIMDB datasets.

Dataset	Mode	Best Dev Acc	Hyperparameters
SST	Full Mode	0.526	lr=1e-4, dropout=0.2, epoch=20
	Finetune Mode	0.483	lr=2e-3, dropout=0.09, epoch=20
CFIMDB	Full Mode	0.976	lr=1e-5, dropout=0.5, epoch=20
	Finetune Mode	0.894	lr=2e-3, dropout=0.05, epoch=20

These results confirmed that our fine-tuning GPT2 model was working correctly and was able to achieve good results.

5.4.2 Paraphrase Detection Result

For the Quora dataset, extending training to 20 epochs resulted in a best development accuracy of 0.907, achieved at epoch 9 with a learning rate of $6e-5$. As illustrated in Figure 3, the model converges after approximately 8–10 epochs, with development accuracy plateauing near 0.907. The effect of model scaling and learning rate transfer between GPT-2 and GPT-2-medium is analyzed further in Section 6.1.

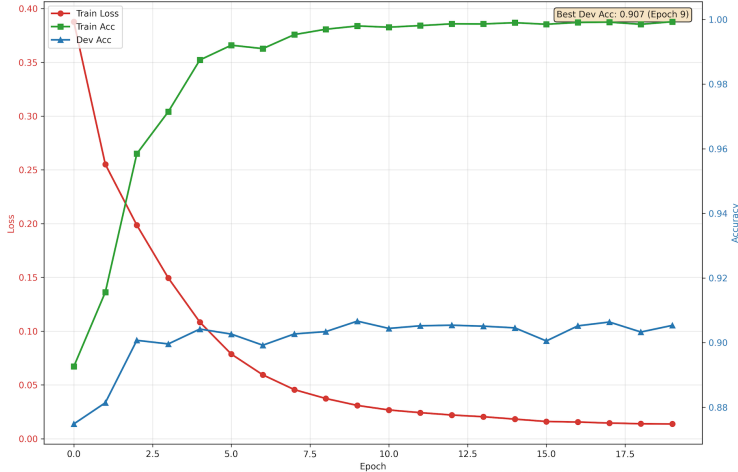


Figure 3: Training Curve for the final GPT2-medium on Quora dataset

We submitted our GPT-2-medium model to the official paraphrase leaderboard, where it achieved a test accuracy of 0.901, exceeding the baseline requirement of 0.862. Table 5 summarizes our final performance on both the development and test sets.

Table 5: QQP paraphrase detection performance with GPT-2-medium.

Model	Dev Acc	Test Acc
SFT GPT-2-medium	0.907	0.901

5.4.3 LoRA Results

Model	SST Accuracy	CFIMDB Accuracy	QQP Accuracy
SFT GPT-2 (baseline)	0.512	0.976	0.89
GPT-2 + LoRA	0.511	0.976	0.87

Table 6: Classification results for sentiment and paraphrase detection tasks.

Table 6 reports results comparing performance baseline vs loRA adaptation for sentiment classification and paraphrase detection (**different baselines from our best baselines**)

For sentiment classification on both SST and CFIMDB, LoRA achieved performance comparable to our baseline using the default LoRA settings with exception of a higher learning rate $1e-4$ (see the Experiment Details section). Increasing the learning rate compared to the baseline resulted in meaningful gains in LoRA when it was under-performing. For paraphrase detection, the default configuration achieved an accuracy of 0.85 and increasing the LoRA rank to 32 improved the accuracy to 0.87. See 6 We refrained from increasing the rank when performance was similar to baseline as it would increase computation cost and decrease model efficiency.

Model	SST		CFIMDB		QQP	
	Train	Eval	Train	Eval	Train	Eval
SFT GPT-2 (baseline)	78	19	64	24	2000	210
GPT-2 + LoRA	44	20	47	24	1350	100

Table 7: Training and evaluation time (seconds per epoch) across tasks comparing full fine-tuning and LoRA.

Full fine-tuning of GPT-2 updates all 124M parameters. With LoRA (rank $r = 8$), the number of trainable parameters is reduced to 0.3 M trainable parameters. In the Sentiment Classification and Paraphrase detection the LoRA adapted model achieved significant efficiency gains in training time and evaluation time. For example on the QQP dataset. Training time decreased from 2000 to 1350 seconds per epoch (a 32.5% reduction). Evaluation time decreased from 210 to 100 seconds per epoch (a 52.4% reduction). Of note, moving from a rank of 8 to rank 32 caused a 100 second gain in training time. Similar findings were also seen in the sentiment classification task. See 7

5.4.4 Sonnet Generation Results

Table 8 summarizes the validation CHRF performance of the supervised fine-tuned (SFT) baseline and the DPO-aligned model. The table also reports the corresponding hyperparameters selected through the grid search procedure described in Section 5.3.

Model	CHRF	Learning Rate	Epochs	β
SFT GPT-2 (baseline)	42.50	1e-5	10	—
GPT-2 + DPO	42.75	5e-7	6	0.01

Table 8: Validation CHRF scores for sonnet generation along with the optimal hyperparameters selected during tuning. DPO is applied as a post-training alignment stage on top of the SFT baseline.

The DPO-aligned model achieves a modest improvement over the supervised fine-tuned baseline, increasing the validation CHRF score from 42.50 to 42.75. Although the gain is relatively small, it suggests that preference-based optimization can help guide the model toward outputs that better align with the stylistic characteristics of Shakespearean sonnets.

The limited magnitude of improvement may be partially explained by the automatic construction of preference pairs. In our approach, dispreferred responses are selected based on lower CHRF scores relative to the ground-truth continuation. While this provides a scalable method for generating preference data, it may not fully capture nuanced stylistic or structural qualities of poetic text.

Additionally, CHRF may not be the most suitable metric for evaluating poetry generation. Although it measures character-level similarity between generated and reference text, it does not explicitly capture important poetic attributes such as rhyme structure, rhythm, or stylistic coherence. As a result, improvements in stylistic quality may not always be reflected in CHRF scores.

We also submitted our model to the official sonnet generation leaderboard. The DPO-aligned model achieved a test CHRF score of 41.877, which is consistent with the validation performance reported in Table 8.

6 Analysis

6.1 Paraphrase Detection Analysis

Our experiments on the Quora Question Pairs dataset reveal two key factors influencing model performance: model size and learning rate. These results highlight that both architectural capacity and precise learning rate selection are crucial for achieving state-of-the-art performance on paraphrase detection tasks.

Impact of Model Size: Scaling from GPT-2 to GPT-2-medium yielded substantial performance gains. With careful hyperparameter tuning, GPT-2 achieved a development accuracy of 0.897, while GPT-2-medium reached 0.907 with a meaningful improvement. This demonstrates that increased model capacity better captures the semantic nuances required for paraphrase detection, even on a fixed dataset size.

Impact of Learning Rate: Learning rate sweeping proved critical for optimal performance. Notably, the learning rate scaling law [2] guided our transfer from GPT-2 to GPT-2-medium, with the optimal rate shifting from 5e-5 to 6e-5, which is a modest adjustment that proved essential for maximizing performance.

6.2 LoRA Analysis

LoRA significantly reduced the number of trainable parameters and achieved substantial efficiency gains across most tasks. However, performance was slightly lower on the paraphrase detection task, where accuracy decreased from 0.89 to 0.87. Despite this drop, LoRA provided meaningful

improvements in both training and evaluation time across tasks due to the reduced number of trainable parameters.

The lower performance on the paraphrase detection task is likely due to the larger size and complexity of the QQP dataset, which may require a higher-rank adaptation to effectively capture task-specific patterns. In contrast, the sentiment classification datasets are relatively smaller, and a low-rank adaptation was sufficient to learn the task while maintaining strong performance. This interpretation is supported by the improved results observed when increasing the LoRA rank from 8 to 32, suggesting that higher-rank settings provide additional model capacity for more complex tasks.

6.3 DPO Analysis

To understand DPO’s qualitative impact, we compare generations from the SFT baseline and DPO-aligned model:

SFT baseline:

The beauty that you keep
will fade away,
And time will steal the
glory of your youth.
Your fleeting charm cannot
forever stay,
For all must yield to time’s
unyielding truth.

DPO model:

Yet in thy bloom lies seed
of future grace,
If thou wouldst share the
beauty thou dost keep;
For time shall carve his
mark upon thy face,
And silent years thy hidden
harvest reap.

DPO encourages Shakespearean stylistic markers (“thy,” “thou,” “dost”), demonstrating that preference-based optimization can influence generated text style. However, these gains are inconsistent. DPO outputs sometimes feel forced or overly stylized, while the baseline produces simpler but more fluent continuations.

This reflects a limitation of our automatic preference construction: using CHRF similarity to ground-truth sonnets captures character-level overlap but misses deeper poetic structure (meter, rhyme, semantic coherence). While DPO shows promise for stylistic alignment, stronger preference signals or better evaluation metrics are needed to consistently improve poetic quality.

7 Conclusion

In this project, we extended supervised fine-tuning of GPT-2 on sentiment classification, paraphrase detection and sonnet generation with two complementary approaches: DPO for alignment and LoRA for parameter-efficient training.

Our results demonstrate that combining alignment methods with parameter-efficient techniques can jointly improve model quality and training efficiency. Building on these findings, future work could explore improved preference construction strategies, evaluation metrics better suited for poetic generation, higher-rank LoRA adaptations for complex tasks, and human evaluation to better assess stylistic alignment.

8 Team Contributions

All team members contributed to experiment design, cross-checking results across tasks, and preparation of the final report.

- **Patrick Wang** implemented the sentiment classification and paraphrase detection tasks with hyperparameters optimization, achieving the highest development accuracy and top leaderboard performance.
- **Ahmed Khidre** construction of the preference dataset, implementation of DPO from scratch, and evaluation of sonnet generation using CHRF metrics.
- **Izhan Hamza** implementation of LoRA modules for parameter-efficient fine-tuning and improvements evaluation in training time and computational efficiency.

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] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- [3] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.
- [4] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *NeurIPS*, 2023.
- [5] Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ICLR*, 2022.
- [6] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.
- [7] Anonymous. Group relative policy optimization, 2024. Alignment method using group-level preference comparisons.
- [8] Anonymous. Direct alignment preference optimization, 2024. Extension of preference optimization methods.
- [9] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. *arXiv preprint arXiv:1902.00751*, 2019.
- [10] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4582–4597. Association for Computational Linguistics, 2021.
- [11] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2021.
- [12] Quora. First quora dataset release: Question pairs. <https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs>, 2017.
- [13] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher Manning, Andrew Ng, and Christopher Potts. Parsing With Compositional Vector Grammars. In *EMNLP*. 2013.
- [14] Divyansh Kaushik, Eduard Hovy, and Zachary C. Lipton. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*, 2019.
- [15] 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.
- [16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*, 2019.

Appendix

A Example of DPO Preference Pair Construction

For completeness, we illustrate one example of the automatic preference construction procedure used for DPO training. The prompt corresponds to the first three lines of a Shakespearean sonnet, and the table shows several candidate continuations generated for the same prompt along with their CHRF scores relative to the ground-truth continuation.

Candidate Continuation	CHRF
Ground Truth (Preferred) His tender heir might bear his memory. But thou, contracted to thine own bright eyes, Feed'st thy light's flame with self-substantial fuel, Making a famine where abundance lies, Thyself thy foe, to thy sweet self too cruel.	1.00
LLM Generated Candidate A Yet time shall claim the beauty that you keep, And fading roses bow before the years; The mirror shows the promises you reap, While silent winter gathers hidden tears, And youth must yield to age's quiet sweep.	0.63
LLM Generated Candidate B Your beauty shines but hoards its fragile flame, Refusing still the future it might give; You guard the bloom yet leave no noble name, And let the fleeting hour decide to live, While empty fields forget from whence it came.	0.48
LLM Generated Candidate C (Selected as Rejected) To one, of one, still such, and ever so. Kind is my love today, tomorrow kind, Still constant in a wondrous excellence; Therefore my verse, to constancy confined, One thing expressing, leaves out difference.	0.279

Table 9: Example of CHRF-based ranking used to construct DPO preference pairs. Multiple candidate continuations are generated for the same prompt and ranked using CHRF relative to the ground-truth sonnet. The continuation with the lowest CHRF score is selected as the dispreferred response for training.

B Hyperparameter Search for the Paraphrase Detection Task

This appendix provides detailed results of the learning rate sweeps used for tuning the paraphrase detection models on the Quora Question Pairs dataset. All experiments used a batch size of 512 and were trained using the AdamW optimizer with data parallelism across $4 \times \text{H200 GPUs}$.

B.1 GPT-2 Learning Rate Sweep

Table 10 summarizes the learning rates tested with GPT-2. Each configuration was trained for 10 epochs. Figures 4–9 show the corresponding training curves.

Table 10: GPT-2 learning rate sweep results (batch size=512, epochs=10)

Learning Rate	Best Dev Acc	Epoch
1e-5	0.879	9
5e-5	0.897	7
6e-5	0.896	7
7e-5	0.896	9
8e-5	0.898	9
9e-5	0.895	6

Although a learning rate of $8e-5$ achieved slightly higher accuracy (0.898), the improvement required additional training epochs. Therefore, $5e-5$ provided the best balance between performance and efficiency and was selected as the base learning rate for scaling experiments.

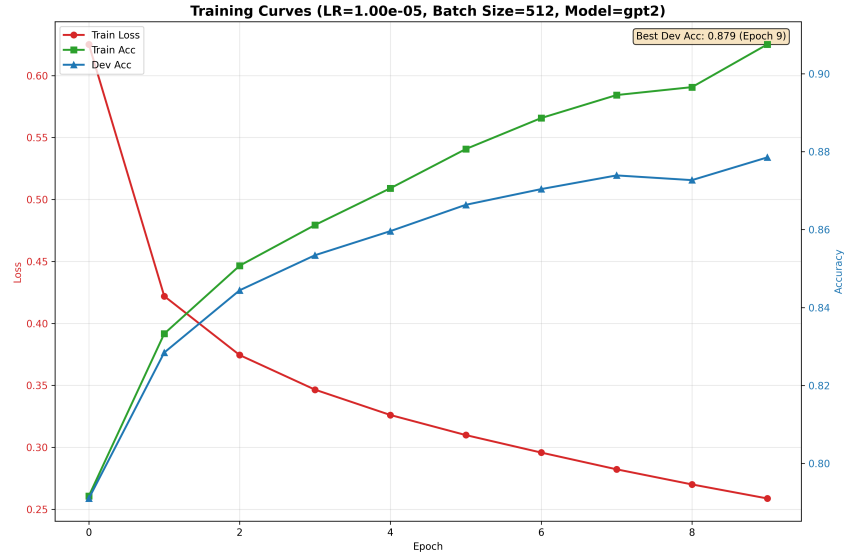


Figure 4: GPT-2 training curves (LR= $1e-5$). Best dev acc: 0.879 at epoch 9.

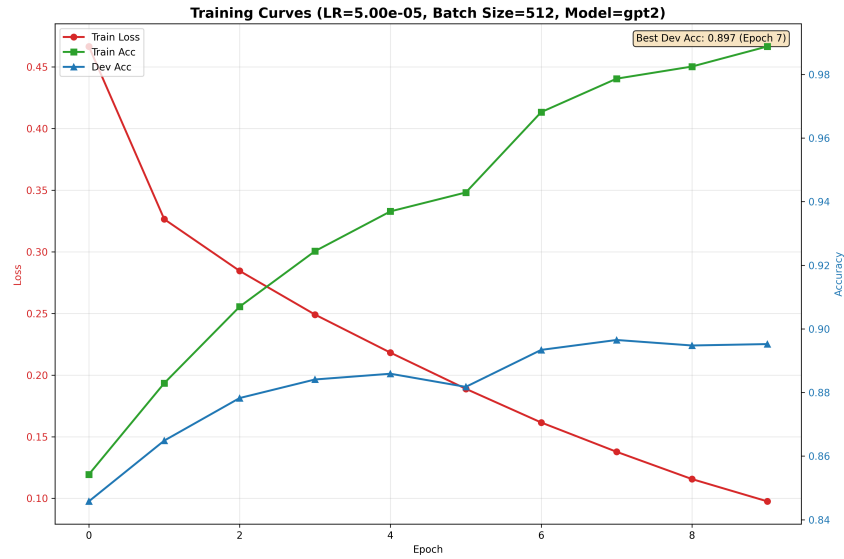


Figure 5: GPT-2 training curves (LR= $5e-5$). Best dev acc: 0.897 at epoch 7.

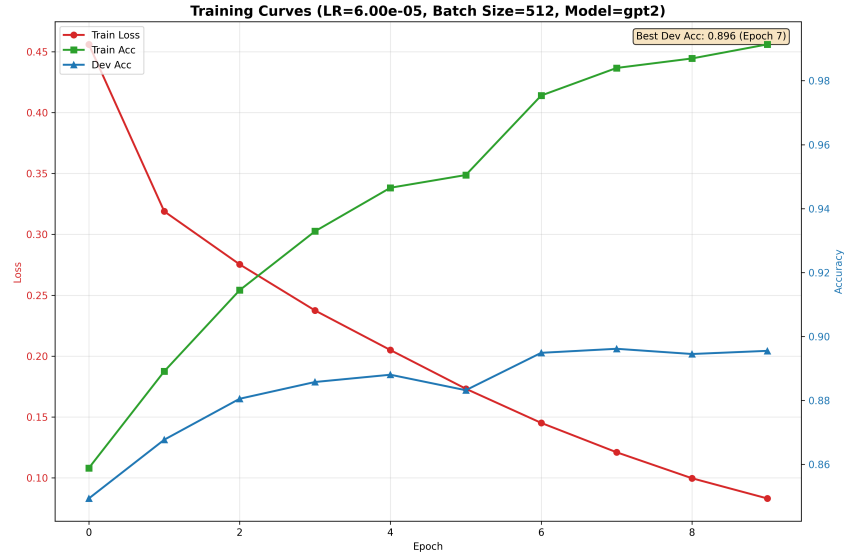


Figure 6: GPT-2 training curves (LR=6e-5). Best dev acc: 0.896 at epoch 7.

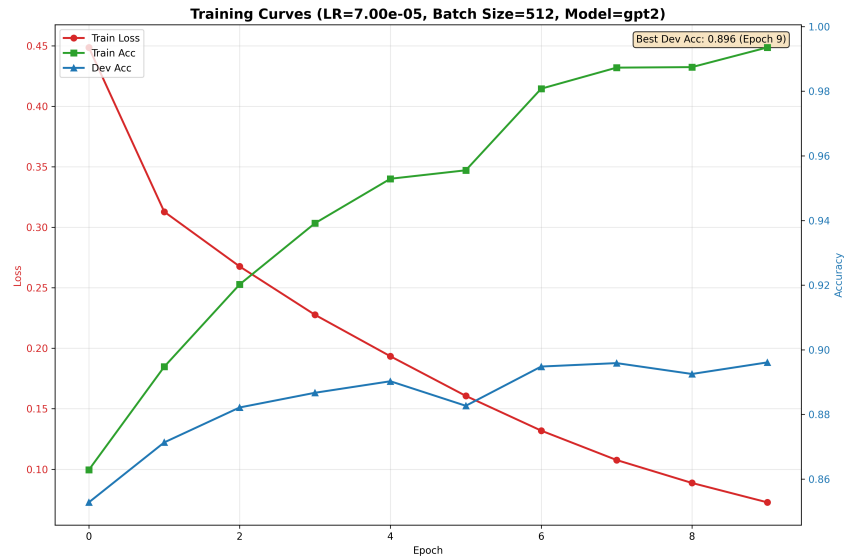


Figure 7: GPT-2 training curves (LR=7e-5). Best dev acc: 0.896 at epoch 9.

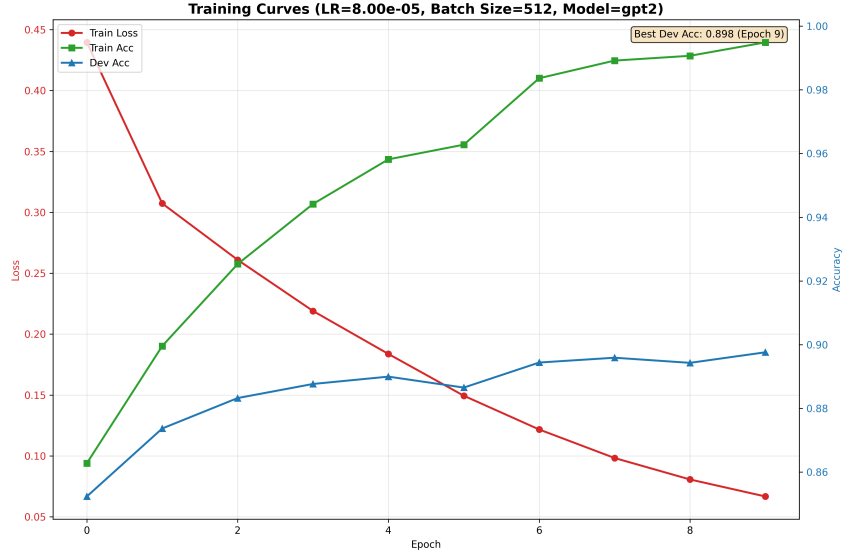


Figure 8: GPT-2 training curves (LR=8e-5). Best dev acc: 0.898 at epoch 9.

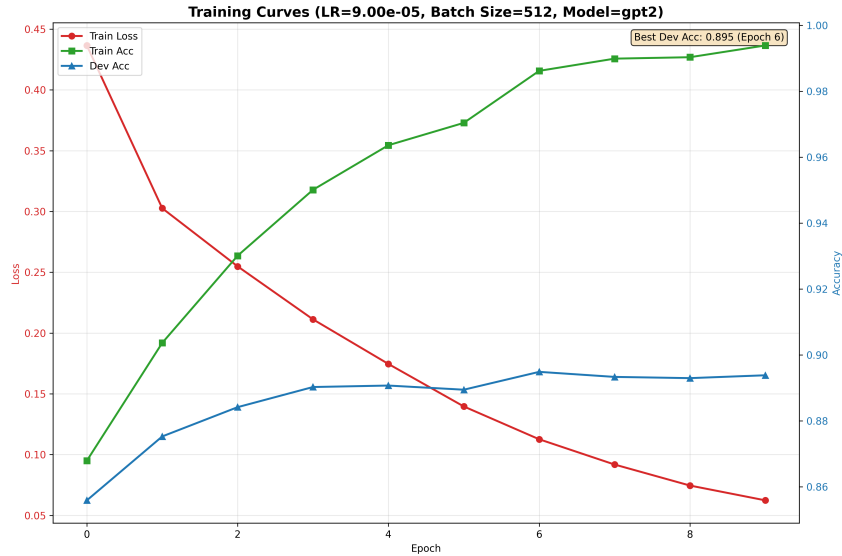


Figure 9: GPT-2 training curves (LR=9e-5). Best dev acc: 0.895 at epoch 6.

B.2 GPT-2-medium Learning Rate Sweep

Using the learning rate scaling law [2], we centered the GPT-2-medium search around the optimal GPT-2 learning rate. Each configuration was trained for 5 epochs.

Table 11: GPT-2-medium learning rate sweep results (batch size=512, epochs=5)

Learning Rate	Best Dev Acc	Epoch
3e-5	0.901	4
4e-5	0.903	4
4.5e-5	0.902	4
5e-5	0.904	4
5.5e-5	0.903	4
6e-5	0.904	4
7e-5	0.904	4

Figures ??-?? show the corresponding training curves.

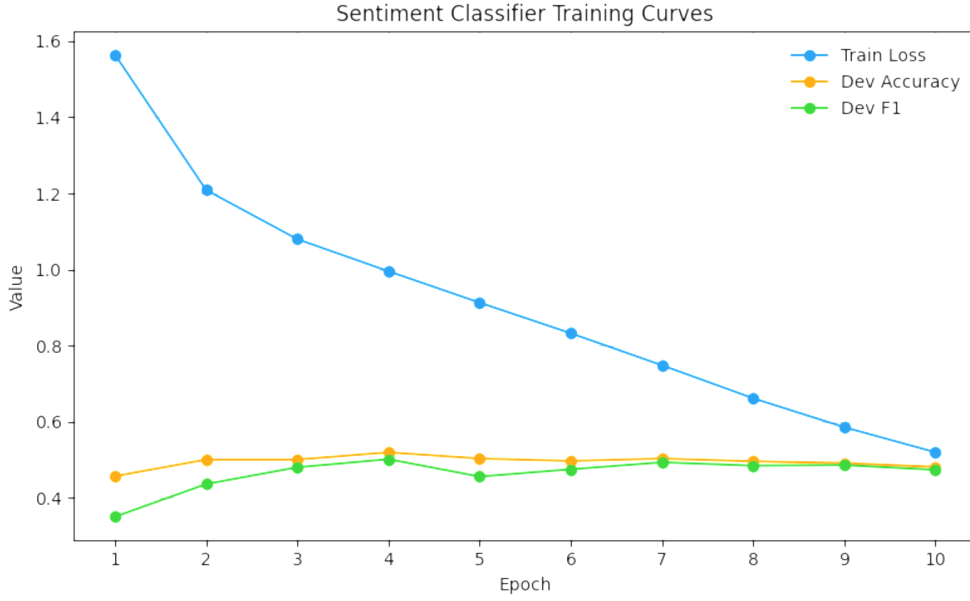


Figure 11: SST baseline implementation

Both 6e-5 and 7e-5 achieved similar development accuracy. We selected 6e-5 as the optimal learning rate for extended training.

B.3 Final GPT-2-medium Performance

Extending training to 20 epochs with LR=6e-5 produced our best model, achieving a development accuracy of 0.907 at epoch 9.

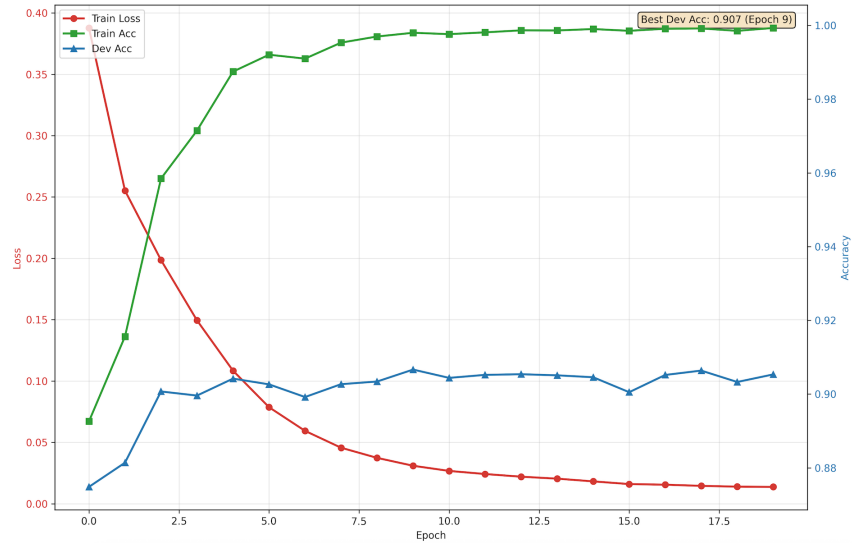


Figure 10: GPT-2-medium final training curve (LR=6e-5, 20 epochs). Best dev acc: 0.907 at epoch 9.

LoRA plots:

Sentiment classification baseline plot for SST can be seen in 11. The LoRA version can be seen in 12

Sentiment classification baseline plot for CFIMDB can be seen in 13. The LoRA version can be seen in 14

Paraphrase detection baseline plot can be seen in 15. The LoRA version can be seen in 16

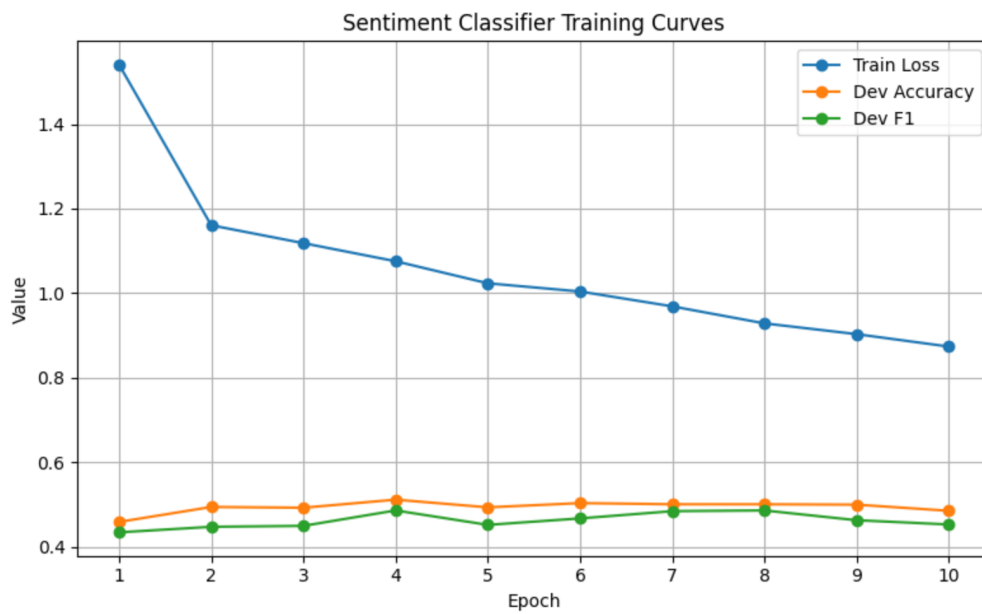


Figure 12: SST LoRA implementation

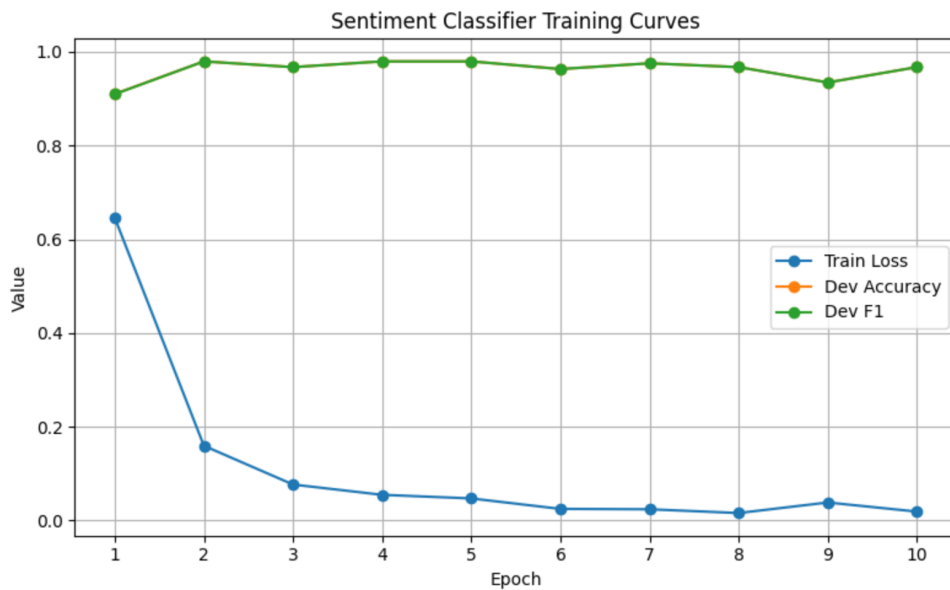


Figure 13: CFIMDB baseline implementation

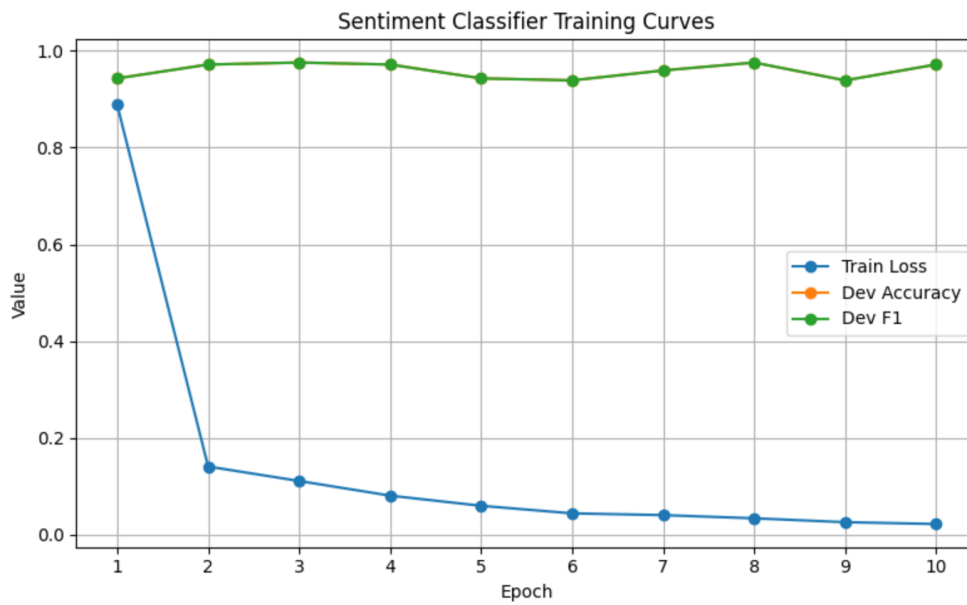


Figure 14: CFIMDB LoRA implementation

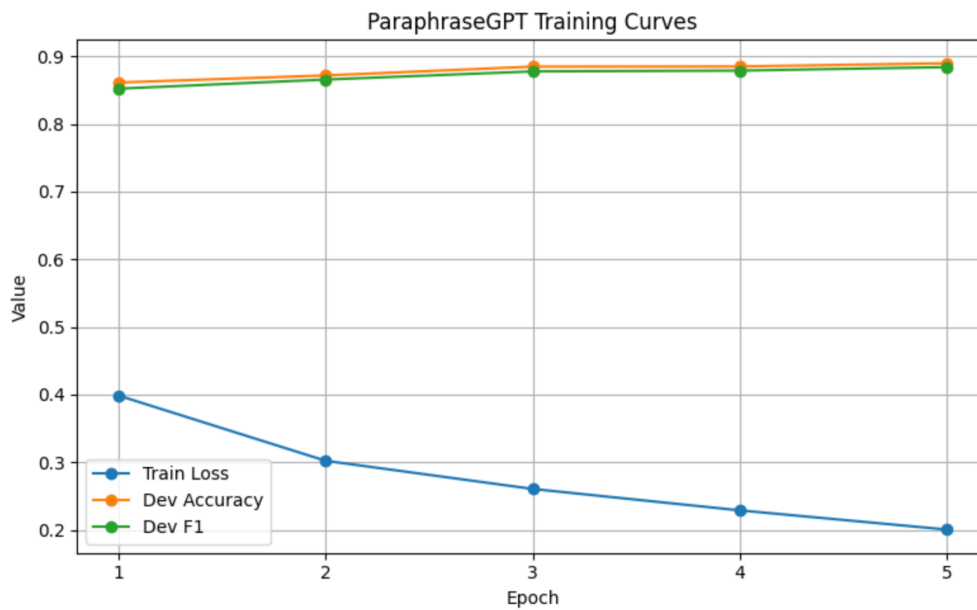


Figure 15: paraphrase detection baseline

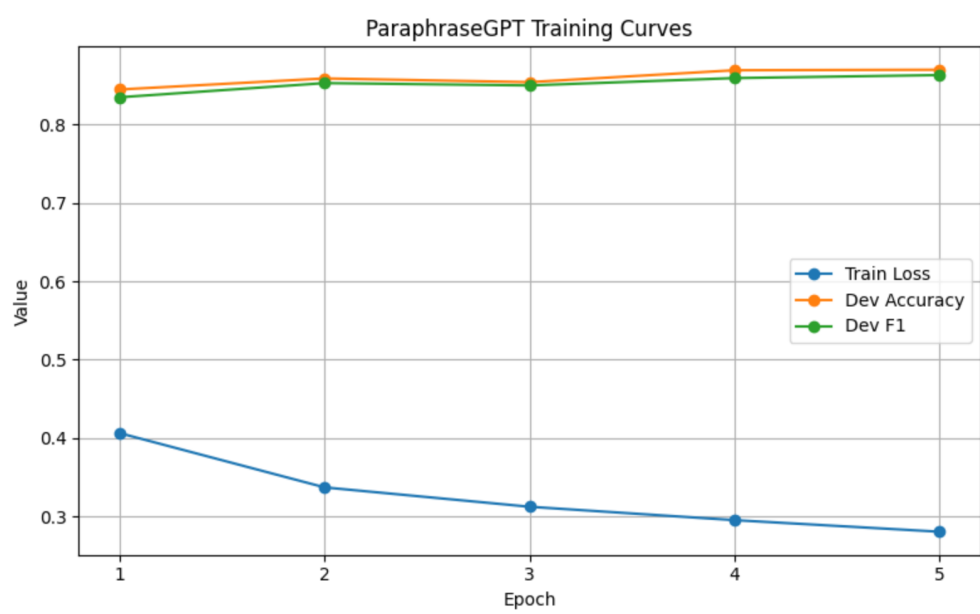


Figure 16: paraphrase detection LoRA