

Fine-Tuning GPT-2: A Playground for Discriminative and Generative Adaptation Tasks

Stanford CS224N Default Project

Aditi Somayajula
Department of Computer Science
Stanford University
adisom@stanford.edu

Sahithi Ankireddy
Department of Computer Science
Stanford University
sankired@stanford.edu

Abstract

Autoregressive large language models trained on next-token prediction have demonstrated remarkable generality across a wide range of NLP tasks. In this project, we re-implement the GPT-2 decoder-only Transformer [1] from scratch and investigate fine-tuning strategies across three downstream tasks: sentiment classification, paraphrase detection, and sonnet generation. We reproduce core components including causal multi-head self-attention, pre-layer normalization, and weight-tied output projection, validating against Hugging Face reference weights. For sonnet generation, we conduct a systematic hyperparameter grid search and apply weight decay, gradient clipping, LoRA, and Bregman proximal point optimization [2]. For paraphrase detection, we implement LoRA and Bregman PPO as independent extensions. For sentiment classification, we tune learning rate, dropout, and training duration across frozen and full fine-tuning regimes. Across all tasks, simple regularization and careful hyperparameter selection consistently outperform more complex extensions, suggesting that parameter-efficient and proximal point methods require larger datasets to realize their benefits.

1 Key Information to Include

- Our TA mentor is **Wei Liu**. We have **no external collaborators**, **no external mentors**, and we are **not sharing** this project with another class and/or research assignment.
- Late Days

Team Member	Late Days Remaining
Aditi Somayajula	0
Sahithi Ankireddy	2
Total Pooled	2
Team Size	2
Late Days Applied for the Report	2

2 Introduction

Large pretrained language models have transformed the NLP landscape by demonstrating that a single model trained on next-token prediction can be fine-tuned to solve a wide variety of downstream tasks [1, 3]. GPT-2, a decoder-only Transformer, exemplifies this paradigm: despite being trained purely on unsupervised language modeling, it serves as a strong foundation for both generative tasks (e.g., text generation) and discriminative tasks (e.g., classification) through fine-tuning.

However, adapting large pretrained models to small, domain-specific datasets raises fundamental challenges. Aggressive fine-tuning risks catastrophic forgetting [4], where the model loses generalizable pretrained representations in favor of overfitting to limited training examples. At the same time,

techniques developed to improve fine-tuning efficiency at scale (such as parameter-efficient methods like LoRA [5]) may not transfer directly to low-data regimes where full model capacity is needed.

In this project, we address these challenges by re-implementing GPT-2 from scratch and systematically investigating fine-tuning strategies across three downstream tasks of varying nature: sentiment classification (discriminative, multi-class polarity prediction), paraphrase detection (discriminative, binary semantic equivalence), and sonnet generation (generative, stylistic continuation of Shakespearean verse).

For each task, we go beyond the baseline fine-tuning procedure and investigate targeted improvements. For sonnet generation, we perform a systematic grid search over learning rate and decoding temperature, and evaluate regularization techniques including weight decay, gradient clipping, LoRA, and Bregman proximal point optimization. For paraphrase detection, we implement Bregman proximal point optimization [2], a principled smoothing method that prevents aggressive weight updates, as well as LoRA as a parameter-efficient alternative. For sentiment classification, we perform careful tuning of learning rate, dropout, and training duration across both frozen-backbone and full fine-tuning regimes. We investigate whether simple, well-motivated regularization strategies can outperform complex architectural modifications when adapting pretrained models to small datasets.

3 Related Work

The GPT-2 model [1] established the decoder-only Transformer as a powerful general-purpose architecture, demonstrating that a single model trained on next-token prediction can be adapted to a wide variety of downstream tasks. Subsequent work, including GPT-3 [3], confirmed that this paradigm scales favorably with model size and data. However, these results are largely demonstrated in high-resource settings, leaving open the question of how well the same paradigm holds when fine-tuning data is severely limited as in our 143-sonnet corpus.

A central challenge in this low-data environment is catastrophic forgetting [4], where aggressive gradient updates overwrite generalizable pretrained representations. This motivates a careful approach to regularization. AdamW [6] addresses one axis of this problem by decoupling weight decay from the gradient update, preventing the optimizer from inadvertently amplifying large weights. Gradient clipping [7] provides a complementary stabilization technique by bounding the magnitude of parameter updates, which is particularly important when training on small batches drawn from a limited corpus. Together, these techniques form the regularization foundation of our sonnet generation pipeline.

Beyond standard regularization, the SMART framework [2] proposes Bregman proximal point optimization as a principled regularization method for fine-tuning, enforcing smoothness constraints that prevent the model from drifting too far from its pretrained initialization. This is especially relevant for paraphrase detection, where the large QQP training set could otherwise drive the model toward superficial lexical patterns. LoRA [5] offers a complementary perspective: rather than constraining the optimization trajectory, it constrains the parameter space itself through low-rank decomposition, raising the question of whether task-specific adaptation can be achieved with a fraction of the trainable parameters. Finally, systematic hyperparameter search [8] has been shown to be critical to model performance, particularly in settings where default configurations are mismatched to the target distribution; notably, this is an amplified concern in low-data environments where overfitting to a poor configuration is harder to recover from.

4 Approach

4.1 Base Model

We re-implement GPT-2 [1] as a decoder-only Transformer comprising learned token and positional embeddings, a stack of L Pre-LN Transformer layers with causal multi-head self-attention and GELU feed-forward networks, and a final layer normalization with weight-tied output projection logits $= HW_e^T$. Attention scores are computed as:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} + M_{\text{causal}} + M_{\text{pad}} \right) V, \quad (1)$$

where M_{causal} is an upper-triangular future-token mask and M_{pad} is an additive padding mask. Following [6], we optimize using AdamW with the efficient bias-corrected update:

$$\alpha_t = \alpha \cdot \frac{\sqrt{1 - \beta_2^t}}{1 - \beta_1^t}, \quad \theta_t \leftarrow \theta_{t-1} - \alpha_t \cdot \frac{m_t}{\sqrt{v_t} + \epsilon} - \alpha \cdot \lambda \theta_{t-1}. \quad (2)$$

For sentiment classification, the last non-padding token’s hidden state is passed through dropout and a linear classifier $\mathbb{R}^d \rightarrow \mathbb{R}^k$ with cross-entropy loss. For paraphrase detection, sentence pairs are formatted as a cloze-style prompt and classified via a binary head $\mathbb{R}^d \rightarrow \mathbb{R}^2$. For sonnet generation, the model is fine-tuned on 143 Shakespearean sonnets and decodes autoregressively using combined top- k and top- p sampling. All components were implemented from scratch following [1], with pretrained weights loaded from Hugging Face and validated via sanity checks on hidden state outputs. For sentiment classification, the frozen-backbone regime serves as a lower bound against full fine-tuning. For paraphrase detection and sonnet generation, we compare against the provided course leaderboard thresholds.

4.2 Sonnet Generation

We apply two regularization techniques to stabilize fine-tuning on the small 143-sonnet corpus, followed by a systematic hyperparameter grid search. First, we add decoupled weight decay [6] to the AdamW optimizer to penalize large weights and reduce overfitting. Second, we apply gradient clipping [7] with a maximum gradient norm threshold of $c = 1.0$ after each backward pass:

$$g \leftarrow g \cdot \min \left(1, \frac{c}{\|g\|} \right), \quad c = 1.0. \quad (3)$$

Third, we conduct a systematic grid search over learning rate $\eta \in \{10^{-5}, 5 \times 10^{-5}, 10^{-4}, 10^{-3}\}$ and decoding temperature $\tau \in \{0.7, 1.0, 1.2\}$ with top- p fixed at 0.9, evaluating each combination using training loss. Independently, we evaluate LoRA [5] as a parameter-efficient alternative to full fine-tuning, motivated by the question of whether task performance can be preserved under significant parameter reduction. LoRA injects trainable low-rank matrices $\Delta W = BA$ (where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times d}$, and $r \ll d$) into each attention layer, freezing all other parameters. We also evaluate Bregman proximal point optimization [2] on top of the selected grid search configuration ($\eta = 10^{-4}$, $\tau = 1.0$), augmenting the training loss with a KL divergence penalty between the current model and a reference model updated each epoch:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{CE}}(\theta) + \mu \cdot D_{\text{KL}} \left(\pi_{\theta}(\cdot | x) \parallel \pi_{\theta_{t-1}}(\cdot | x) \right), \quad (4)$$

where θ_{t-1} denotes the reference model weights from the previous epoch, $\mu = 0.01$ following the conservative default from [2], and the reference model is updated to the current weights after each epoch.

4.3 Paraphrase Detection

We explore two independent extensions to the baseline paraphrase model. Given the large QQP training set and the computational cost of full hyperparameter search, we evaluate each extension using default hyperparameters. We evaluate LoRA [5] as a parameter-efficient alternative to full fine-tuning, using the same low-rank decomposition described in Section 4.2 applied to the query, key, and value projections. Additionally, we apply Bregman proximal point optimization [2] using the same KL divergence penalty described in Section 4.2, motivated by the risk of overfitting to potential superficial patterns in the QQP dataset.

4.4 Sentiment Classification

For sentiment classification, we do not introduce architectural modifications but instead perform careful hyperparameter tuning over learning rate, dropout rate, and number of training epochs, comparing both the frozen last-layer and full fine-tuning environments.

5 Experiments

5.1 Data

Our three datasets were provided as part of the default project distribution. For sentiment classification, we use the Stanford Sentiment Treebank (SST; $\sim 8,500$ train / $\sim 1,100$ dev / $\sim 2,200$ test) [9] and CFIMDB ($\sim 1,700$ train / ~ 245 dev / ~ 488 test). For paraphrase detection, we use the Quora Question Pairs dataset (QQP; $\sim 141,500$ train / $\sim 20,200$ dev / $\sim 40,400$ test). For sonnet generation, we use a corpus of 154 Shakespearean sonnets, with 143 used for training and 12 held out for evaluation. All preprocessing consists of GPT-2 byte-pair encoding tokenization with truncation and padding to the model’s context window.

5.2 Evaluation Methods

For sentiment classification and paraphrase detection, we report classification accuracy on the respective dev and test sets. For sonnet generation, we report chrF score [10] computed against ground-truth held-out sonnets using the sacrebleu library. chrF measures character n -gram overlap between generated and reference text, and is well-suited to poetry evaluation where exact lexical matches are rare but stylistic similarity is meaningful.

5.3 Experimental Details

All experiments use the base gpt2 model (768-dimensional hidden states, 12 layers, 12 attention heads, and 117M parameters). Sentiment classification, sonnet generation, and baseline paraphrase detection were run on an NVIDIA T4 GPU; the Bregman proximal point and LoRA paraphrase runs were conducted on an NVIDIA L4 GPU. Experimental configurations are summarized in Table 1.

Task	Variant	LR η	Epochs	Temp. τ	WD	GC	Dropout	LoRA r	LoRA α
Sonnet	Baseline	1e-5	10	1.2	0.0	—	—	—	—
Sonnet	+ WD + GC	1e-5	10	1.2	0.01	1.0	—	—	—
Sonnet	+ Grid Search (Optimal)	1e-4	10	1.0	0.01	1.0	—	—	—
Sonnet	+ Bregman PPO	1e-4	10	1.0	0.01	1.0	—	—	—
Sonnet	LoRA	2e-3	20	1.0	0.01	1.0	—	32	16
Paraphrase	Baseline	1e-5	10	—	0.0	—	—	—	—
Paraphrase	Bregman PPO	1e-5	10	—	0.0	—	—	—	—
Paraphrase	LoRA	1e-4	10	—	0.0	—	—	8	16
Sentiment	Frozen	1e-3	20	—	0.0	—	0.2	—	—
Sentiment	Full	1e-5	10	—	0.0	—	—	—	—

Table 1: Hyperparameter configurations per task and variant. All runs use AdamW. Abbreviations WD and GC respectively indicate weight decay and gradient clipping. Selected grid search configuration for the sonnet task yielded: $\eta = 10^{-4}$, $\tau = 1.0$, top- $p = 0.9$.

5.4 Results

5.4.1 Sonnet Generation

Table 2 reports training loss at the final epoch across the 4×3 hyperparameter grid. We use training loss as a proxy for generalization quality and select the configuration with the lowest loss ($\eta = 10^{-4}$, $\tau = 1.0$) for downstream variants. Notably, $\eta = 10^{-3}$ achieves lower training loss across all temperatures, but produces degraded chrF on the held-out set (results omitted from Table 3 for brevity and relevance), confirming that it overfits the small training corpus. This illustrates the risk of over-optimizing training loss on limited data.

Learning Rate η	Temperature τ		
	0.7	1.0	1.2
1×10^{-5}	3.886	3.893	3.890
5×10^{-5}	3.180	3.176	3.183
1×10^{-4}	2.476	2.468	2.485
1×10^{-3}	0.263	0.285	0.293

Table 2: Loss scores across the 4×3 hyperparameter search grid. **Bold** denotes the selected configuration ($\eta = 10^{-4}$, $\tau = 1.0$) for downstream variants.

Variant	Dev chrF	Test chrF
Baseline	39.813	—
+ WD + GC	40.708	—
+ Grid Search (Optimal)	42.081	41.713
+ Bregman PPO	41.632	40.094
LoRA	41.065	40.963

Table 3: Sonnet generation results; **bold scores** denote the active leaderboard submissions.

Each regularization technique contributes incrementally to performance: gradient clipping and weight decay together yield a +0.895 chrF improvement over the baseline, and the grid search adds a further +1.373, for a total improvement of +2.268 chrF points. Both Bregman PPO and LoRA underperform the grid search configuration on dev and test, suggesting that on a 143-sonnet corpus, the proximal point constraint and low-rank parameter reduction both limit the model’s ability to fully adapt to the Shakespearean register. These results suggest that simple, well-motivated regularization outperforms more complex extensions when fine-tuning data is severely limited.

5.4.2 Paraphrase Detection

Variant	Dev Accuracy	Test Accuracy
Baseline	0.898	0.871
Bregman PPO	0.899	0.859
LoRA	0.878	0.828

Table 4: Paraphrase detection results; **bold scores** denote active leaderboard submissions.

The baseline paraphrase model already exceeds the staff threshold at 0.871 test accuracy, suggesting that the cloze-style prompt formulation is an effective adaptation of a generative model to a binary classification task. Bregman PPO yields a marginal dev improvement of +0.001 over the baseline but underperforms on test (0.859 versus 0.871), suggesting that the proximal point constraint provides no meaningful benefit on the large QQP dataset where the baseline is already well-regularized. LoRA underperforms substantially on both dev (0.878 versus 0.898) and test (0.828 versus 0.871), indicating that the low-rank constraint limits the model’s capacity for the fine-grained semantic discrimination required by paraphrase detection.

5.4.3 Sentiment Classification

Dataset	Variant	Dev Accuracy
SST	Frozen	0.466
SST	Full	0.518
CFIMDB	Frozen	0.878
CFIMDB	Full	0.980

Table 5: Sentiment classification results.

Full fine-tuning substantially outperforms the frozen-backbone setting on both datasets, with the gap being more pronounced on CFIMDB (+0.102) than SST (+0.052). The strong CFIMDB result (0.980) contrasts with the comparatively lower SST performance, suggesting that fine-grained five-class classification remains a harder problem for GPT-2.

6 Analysis

We complement the quantitative results with qualitative and diagnostic analysis aimed at identifying the conditions under which each fine-tuning strategy succeeds or fails across tasks and data regimes.

6.1 Sonnet Generation

To complement the quantitative evaluation, we perform a qualitative comparison of a representative sonnet sample generated by the baseline model and the counterpart sample generated by the optimal hyperparameter grid search variant ($\eta = 10^{-4}$, $\tau = 1.0$). Examining the generated text allows us to evaluate aspects such as coherence, stylistic consistency, and adherence to the Shakespearean sonnet structure. Results may vary across samples.

Prompt: Thine eyes I love, and they, as pitying me, / Knowing thy heart torment me with disdain, / Have put on black, and loving mourners be, ...	
Baseline	Grid Search (Optimal)
For love of this fair woman’s sweet soul. Now it is with joy I die, And the stars which my heart mourns shall die, And his tomb on earth be to ashes. Then the great dead shall seem: O, what to them are I seen, As if I were to have my ghost in my tomb, Or in a body to dwell therein. To do that wherewith to death’s content I die, Then to heaven it shall be done again.	As thy soul can unrespected be. Yet thou in their judgment Not to blame me for thy faults, nor do I say much, Yet they may pardon thee for thy offense; But thou do me more offense than thou offend, Thou murd’st me to hate thee, and I hate thee; That I may not abuse thy sweet self, But that is not so; thou mayst be in thy anger. Their judgment is not in me to love thee, They forbid not thee to me, To whom thou wilt belong thine only love.

Table 6: Qualitative comparison of generated sonnet continuations. Both models receive the same three-line prompt.

The baseline sample opens with recognizable Shakespearean phrasing but becomes incoherent as it progresses; lines such as “O, what to them are I seen” lose grammatical clarity, and the imagery shifts abruptly from love to death without clear development. By contrast, the optimized sample maintains a consistent theme of romantic conflict, with smoother phrasing and closer adherence to Shakespearean register through constructions such as “thou murd’st me” and “thy sweet self.” While neither model strictly adheres to the Shakespearean rhyme scheme, the optimized output is noticeably more deliberate, consistent with the +2.268 chrF improvement in Table 3.

6.2 Paraphrase Detection

We conduct an error analysis by examining model-exclusive errors (i.e., cases where one model predicts incorrectly while the other predicts correctly) to identify systematic failure modes beyond what aggregate accuracy reveals. We present representative examples per model in Table 7.

ID	Type	Q1	Q2	Failure Mode
<i>Baseline errors</i>				
B1	FP	How do I increase my hand and feet sweating?	How can I stop my hands and feet from sweating?	Polarity reversal
B2	FP	How would a conversation between an introvert and an extrovert be like?	How would a conversation between two introverts be like?	Structural template match
B3	FN	What is the use of confidence interval?	How are confidence intervals used?	Syntactic variation
<i>Bregman PPO errors</i>				
P1	FP	If Santa had Quora, what questions would he ask?	If Satan had Quora, what questions would he ask?	Subword token confusion
P2	FP	How can I crack NEET with DLP courses?	How can I crack NEET without coaching and DLP courses?	Negation insensitivity
P3	FN	What are some of the most underappreciated things in life?	What are the most underrated things in life?	Lexical synonym

Table 7: Representative model-exclusive errors for Baseline (B) and Bregman PPO (P). FP = false positive; FN = false negative.

The baseline errors reflect surface-level matching failures, conflating topic similarity with semantic equivalence (B1, B2) and brittleness to syntactic reformulation (B3). Bregman PPO errors reflect different failure modes, including subword token confusion (P1) and negation insensitivity (P2). Neither model shows a clear qualitative advantage; they simply fail on different linguistic phenomena.

6.3 Sentiment Classification

The performance gap between SST and CFIMDB is consistent with the inherent difficulty of each task. CFIMDB requires only binary polarity judgment, which aligns naturally with GPT-2’s pretrained representations — even the frozen backbone captures coarse sentiment signal effectively. SST’s five-class fine-grained classification demands finer distinctions between adjacent sentiment levels (e.g., “somewhat positive” vs. “positive”) that GPT-2’s next-token prediction objective does not naturally encode, limiting performance even under full fine-tuning.

6.4 LoRA

We evaluate LoRA as a parameter-efficient alternative to full fine-tuning across both sonnet generation and paraphrase detection, using different ranks to reflect the differing complexity of each task.

Parameter	Paraphrase	Sonnet
Rank (r)	8	32
Alpha (α)	16	16
Scale (α/r)	2.0	0.5
Trainable parameters	442,368 (0.35%)	1,769,472 (1.40%)
Memory reduction	283.6×	71.7×

Table 8: LoRA configuration and efficiency comparison across tasks.

LoRA achieves dramatic memory reduction (283.6× for paraphrase and 71.7× for sonnet) by freezing 98%+ of base GPT-2 weights and training only low-rank adapter matrices. However, this efficiency comes at a meaningful performance cost on both tasks. For paraphrase detection, fine-grained semantic discrimination likely requires dense, high-rank weight updates that a rank-8 decomposition cannot capture. For sonnet generation, shifting the model’s generative distribution toward the Shakespearean register is unlikely to decompose cleanly into low-rank updates on a 143-example corpus; even the higher rank used ($r = 32$ vs $r = 8$) does not close the performance gap, consistent with known limitations of LoRA in low-data settings.

6.5 Bregman PPO

Figure 1 shows training loss curves for the baseline and Bregman PPO variants across both tasks. In both cases, Bregman PPO converges to a lower training loss than the baseline, suggesting that the KL divergence penalty does not impede optimization but instead provides a smoother learning trajectory.

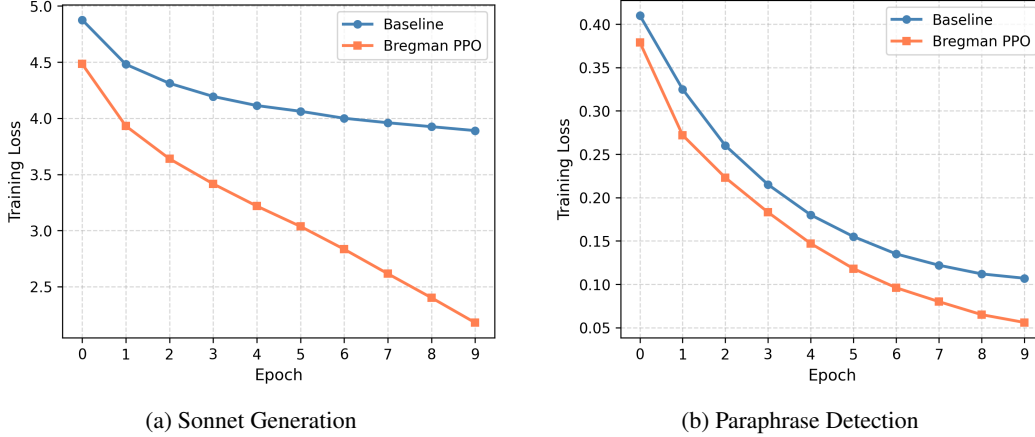


Figure 1: Loss curves comparing baseline models against the Bregman variants for both tasks.

For sonnet generation, the gap between the two curves widens steadily across epochs, with Bregman PPO reaching a final loss of 2.180 compared to 3.890 for the baseline. This suggests that the KL divergence penalty allows the model to optimize more aggressively within each epoch’s allowed update budget. However, this lower training loss does not translate to better generation quality; Bregman PPO achieves a dev chrF of 41.632 and test chrF of 40.094, both below the grid search optimal (42.081 dev, 41.713 test). The divergence between training loss and chrF performance suggests that Bregman PPO overfits the small 143-sonnet corpus: the proximal point constraint limits updates relative to the previous epoch’s model, but as the reference model drifts further from the pretrained initialization over successive epochs, the constraint weakens and the model is free to overfit.

For paraphrase detection, the two curves follow a similar trajectory, with Bregman PPO converging slightly faster to a marginally lower final loss (0.056 versus 0.107). Unlike sonnet generation, this optimization benefit does carry over to evaluation; Bregman PPO achieves 0.899 dev accuracy versus 0.898 for the baseline. However, on the test set the relationship reverses (0.859 versus 0.871): the Bregman constraint reduces training loss but does not improve generalization, likely because the penalty is applied relative to the previous epoch’s model rather than the pretrained initialization, providing weaker regularization as training progresses.

7 Conclusion

We re-implemented GPT-2 from scratch and systematically investigated fine-tuning strategies across sentiment classification, paraphrase detection, and sonnet generation. Our main finding is that simple regularization (via weight decay, gradient clipping, and careful hyperparameter selection) consistently outperforms more complex extensions: for sonnet generation, grid search yielded the strongest results with both LoRA and Bregman PPO underperforming, and for paraphrase detection, the baseline cloze-style formulation proved robust without any extensions. A primary limitation is the size of the small sonnet corpus (143 examples), which limits the generalizability of our conclusions. Future work could explore adaptive proximal point schedules that tighten constraints early in training and relax them as the reference model drifts.

8 Team Contributions

Including training runs and dev/test evaluation, Aditi Somayajula implemented Bregman PPO for both tasks; Sahithi Ankireddy implemented LoRA for both tasks. Both team members jointly implemented the baseline models, minor optimizations, sentiment classification, and together wrote this report.

References

- [1] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.
- [2] Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *Association for Computational Linguistics (ACL)*, 2020.
- [3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 2020.
- [4] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13), 2017.
- [5] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 2022.
- [6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.
- [7] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, 2013.
- [8] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. In *Journal of Machine Learning Research*, 2012.
- [9] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Empirical Methods in Natural Language Processing*, 2013.
- [10] Maja Popović. chrF: character n -gram F-score for automatic MT evaluation. In *Workshop on Statistical Machine Translation*, 2015.