

Implementing a GPT-2 Decoder for Text Generation, Classification, and Paraphrase Detection

Stanford CS224N Default Project

TA mentor: Ali Sartaz Khan

External collaborators/Mentors: No

Sharing project: No

Alma Cooper

Department of Statistics
Stanford University
alma1@stanford.edu

Louis Weisdorf

Department of Computer Science
Stanford University
weisdorf@stanford.edu

Antra Nakhasi

Department of Management Science and Engineering
Stanford University
anakhasi@stanford.edu

Abstract

We implement GPT-2 (117M parameters) and fine-tune it across sentiment analysis, paraphrase detection, and sonnet generation. Our implementation includes masked multi-head self-attention, the full GPT-2 transformer architecture, and the AdamW optimizer. Full-model fine-tuning achieves 0.514 accuracy on SST-5 and 0.980 on CFIMDB. For paraphrase detection, we reformulate the task as cloze-style classification and utilize Parameter-Efficient Fine-Tuning (PEFT) via LoRA. By targeting all linear layers (Attention and MLP) and scaling to the GPT-2 Large model, we achieve a peak development accuracy of 0.897 on the Quora dataset, significantly outperforming the GPT-2 Small full-model fine-tuning baseline of 0.881. Our analysis highlights the high sensitivity of fine-tuned classifiers to prompt syntax and the limitations of few-shot prompting in specialized classification heads. For sonnet generation, we fine-tune GPT-2 on 143 Shakespeare sonnets with training fixes including padding-aware loss and early stopping, then compare two inference-time extensions: NeuroLogic Decoding with soft phonetic rhyme constraints and deterministic beam search. Beam search achieves the highest ChrF of 42.329 on the dev set, while NeuroLogic produces structurally stronger sonnets at a small metric cost, revealing a fundamental tension between automatic evaluation and qualitative generation quality. We also report BERTScore and a custom sonnet composite metric. This composite metric provides a more task-appropriate evaluation for poetry generation, where semantic similarity alone does not fully reflect output quality.

1 Introduction

Large language models trained with next-token prediction transfer effectively to many downstream tasks, yet fine-tuning remains challenging under low-resource conditions and for structured generation. We implement GPT-2 across three increasingly complex settings: sentiment classification, paraphrase detection, and constrained creative generation.

For paraphrase detection, we apply LoRA across all linear layers and scale to GPT-2 Large, analyzing the sensitivity of fine-tuned classification heads to prompt formatting and few-shot context. For sonnet generation using only 143 training sonnets, we identify and correct several failure modes in a baseline implementation, then compare beam search and NeuroLogic Decoding as inference-time extensions that enforce different notions of output quality without retraining.

These experiments also highlight limitations of ChrF as an evaluation metric for creative generation. Because ChrF measures character-level overlap, it may undervalue semantically faithful outputs with lexical variation. We therefore supplement ChrF with BERTScore and introduce a Sonnet Composite Score combining BERTScore F1, rhyme scheme accuracy, structural constraints, and type–token ratio to better capture poetic coherence and form.

2 Related Work

Sentiment classification with pretrained language models was established by the original GPT work [1], demonstrating that a linear head over the final token representation suffices for downstream classification.

Paraphrase detection has since extended this line of work to semantic equivalence, transitioning from feature-based engineering to contextualized representations at the scale of the Quora Question Pairs dataset. [2]. While bidirectional models like BERT have traditionally been preferred for classification [3], causal models like GPT-2 achieve competitive results under cloze-style reformulation [4], and parameter-efficient adaptation via LoRA [5] further reduces the cost of fine-tuning larger models.

For sonnet generation, prior work such as Deep-speare [6] has explored dedicated poetry models that jointly learn meter, rhyme, and language through task-specific architectures requiring substantial training data. Our approach instead follows the direction of constrained decoding at inference time. NeuroLogic Decoding [7] introduced a beam search procedure that enforces arbitrary lexical constraints via clause satisfaction, allowing structural requirements to be imposed on any pretrained language model without modification. We apply this framework to enforce the Shakespearean rhyme scheme using phonetic keys from the CMU Pronouncing Dictionary, extending the original lexical formulation to phonetic matching.

3 Approach

We implement GPT-2, comprising 12 transformer layers with hidden dimension $d = 768$, 12 attention heads, and 117M parameters. Each layer applies pre-layer normalization, masked multi-head self-attention, and a position-wise feed-forward network. Causal masking is composed with the padding mask additively in score space. We load pretrained weights from HuggingFace, splitting the fused Conv1D $[d, 3d]$ query, key, and value projection and transposing appropriately. The AdamW optimizer uses the efficient bias correction form $\alpha_t = \alpha \cdot \frac{\sqrt{1-\beta_2^t}}{1-\beta_1^t}$.

For sentiment classification, a linear projection maps the last non-padding token’s hidden state to the number of classes. We evaluate last-linear-layer and full-model fine-tuning regimes.

For paraphrase detection, we apply LoRA with low-rank updates $\Delta W = BA$ where $r \ll d$ across all linear layers in both attention and MLP blocks. We align the MLP expansion ratio to $4d$ for compatibility with pretrained weights and scale this configuration to GPT-2 Large. The task is formulated as a cloze-style classification problem, where a linear head maps the final token’s hidden state to the paraphrase label space.

For sonnet generation, we fine-tune GPT-2 small with four improvements over our initial implementation: a padding-aware loss mask, early stopping on validation loss, a newline token appended to the prompt, and removal of gradient clipping and label smoothing. At inference we compare two decoding strategies. Beam search ($k = 10$) applies a repetition penalty of 1.3 to all seen tokens except the newline token (token 198), which must recur across all 14 lines and is therefore excluded. NeuroLogic Decoding [7] enforces the ABAB CDCD EFEF GG rhyme scheme via soft phonetic constraints from the CMU Pronouncing Dictionary [8]: rhyme groups A and B are pre-anchored from the 3-line prompt, while groups C–G are anchored as each new group’s first line is generated. At each step a bonus $+\lambda$ is applied to rhyme-matching tokens and a penalty $-\lambda$ to mismatches, with beams pruned by retaining the top- α by log-probability union the top- β by constraint coverage. The end-of-sequence token is masked until 14 lines are complete, ensuring full sonnet generation.

4 Data

Sentiment: SST-5 contains 8,544 training and 1,101 dev examples with 5 sentiment classes (0–4). CFIMDB has 1,701 training and 245 dev examples with binary labels. Both datasets use GPT-2’s BPE tokenizer with the padding token set to EOS.

Paraphrase: Quora Question Pairs contains approximately 130k training and 8k dev examples. Text is lowercased and spaces are added before punctuation marks. Labels are the first BPE token of "yes" or "no".

Sonnet: 154 sonnets written by Shakespeare, each consisting of 14 lines with the rhyme scheme strictly following ABAB CDCD EFEF GG.

5 Evaluation method

Classification tasks are evaluated using accuracy and macro-averaged F1 from scikit-learn, computed on dev sets after each epoch. Best checkpoints are saved by dev accuracy. For paraphrase detection, labels are decoded from 0/1 back to BPE token IDs at inference. Sonnet generation is evaluated using ChrF [9] via sacrebleu against the Shakespeare original; since the 3-line prompt always matches exactly, only the 11 generated continuation lines meaningfully vary. We supplement with qualitative analysis across two sonnets comparing all three generation systems on thematic coherence, syntactic stability, rhyme scheme adherence, and structural integrity.

To extend our evaluation methodology, we analyze BERTScore to capture semantic similarity between generated and referenced sonnets. While ChrF measures character level overlap, it is limited in performance when applied to creative writing where a quality output may be scored low due to its variation and diversity in text. BERTScore addresses this limitation through contextual embeddings generated from a pretrained BERT model.

Let $x = (x_1, \dots, x_n)$ be the candidate tokens and $y = (y_1, \dots, y_m)$ be the reference tokens. Let $\mathbf{h}_i$ and $\mathbf{g}_j$ be their contextual embeddings. BERTScore computes token level cosine-similarities and aggregates them into precision P and recall R ,

$$P = \frac{1}{n} \sum_{i=1}^n \max_{j \in [1, m]} \cos(\mathbf{h}_i, \mathbf{g}_j) \quad (1)$$

$$R = \frac{1}{m} \sum_{j=1}^m \max_{i \in [1, n]} \cos(\mathbf{g}_j, \mathbf{h}_i) \quad (2)$$

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} \quad (3)$$

We additionally, define a Sonnet Composite Score combining BERTScore, structural adherence, rhyme scheme, and lexical diversity.

$$Score_i = 0.35B_i + 0.20S_i + 0.20R_i + 0.25C_i \quad (4)$$

where B_i is the BERTScore F1, S_i is a structure score measuring adherence to the 14-line format, R_i is a rhyme score measuring adherence to the ABAB CDCD EFEF GG scheme, and C_i is a type-token ratio measuring lexical diversity. Type-token ratio addresses a key limitation of ChrF: models that reuse common Shakespearean phrases can score highly despite producing repetitive output. Scores are averaged across all 12 held-out sonnets per system.

6 Experimental details

6.1 Sentiment Analysis

Both SST-5 and CFIMDB were evaluated under two fine-tuning regimes. For last-linear-layer fine-tuning, the GPT-2 backbone is frozen and only the classification head is trained at learning rate 10^{-3} . For full-model fine-tuning, all parameters are updated end-to-end at learning rate 10^{-5} . Both regimes use batch size 64 for SST-5 and 8 for CFIMDB, trained for 10 epochs with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-6}$).

6.2 Paraphrase Detection

Our experiments utilized the Quora Question Answering dataset for paraphrase detection. We compared standard full fine-tuning against Parameter-Efficient Fine-Tuning (PEFT) using Low-Rank Adaptation (LoRA). For our LoRA configurations, we explored targeting different components of the Transformer block: the Self-Attention layers (Query, Key, Value, and Attention Projection) and the Multi-Layer Perceptron (MLP) layers (Intermediate and Output Dense).

The experimental pipeline followed an iterative optimization strategy designed to maximize performance while minimizing computational expenses. We structured this progression into four phases:

- **Baseline Establishment:** Initial experiments were conducted using full fine-tuning on the GPT-2 Small model. After tuning hyperparameters, specifically weight decay and learning rate scheduling with a 10% warm-up, we extended this run to 10 epochs to establish a robust performance baseline.
- **Ablation of LoRA Targets:** To identify the most efficient adapter placement, we performed ablation studies on the GPT-2 Medium model over 3-epoch runs. We compared three configurations: targeting the self-attention blocks, targeting the MLP layers, and an "All-Linear" approach targeting both.
- **Scaling and Convergence:** Having identified the "All-Linear" configuration as the highest performing strategy, we extended its training to 5 epochs to observe convergence. This optimized setup was then scaled to the GPT-2 Large model, where we converged on peak performance within 3 epochs.
- **Inference-Time Sensitivity Tests:** Finally, using the optimized GPT-2 Large checkpoint, we conducted a series of prompt tuning experiments. These tests evaluated the model's sensitivity to prompt formatting (a missing quote typo) and the effectiveness of in-context learning through few-shot prompting.

To ensure resource efficiency throughout this progression, all training runs were checkpointed at every epoch, allowing for detailed analysis of learning curves and the ability to resume training from the most promising states.

All models were trained using the AdamW optimizer with a weight decay of 0.01 and a linear learning rate scheduler with a 10% warm-up phase. The starting learning rate was adjusted throughout experiments based on the type of finetuning and model size. Training was conducted primarily on a G4 GPU. A critical architectural detail discovered during implementation was the expansion ratio of the MLP block. While the given configurations used a $3d$ expansion, we aligned the model with the standard GPT-2 specification of $4d$ to ensure compatibility with pre-trained weights during LoRA weight updates. Furthermore, the model was trained using a cloze-style prompt: *"Question 1: 's1'\nQuestion 2: 's2\nAre these questions asking the same thing?\n"*.

6.3 Sonnet Generation

Our experiments evaluated four sonnet generation systems, comparing our initial implementation against three systems built on an improved fine-tuned checkpoint. All three improved systems share the same checkpoint and differ only in their inference-time decoding strategy, isolating the effect of decoding from training.

The experimental pipeline followed an iterative progression structured into three phases:

- **Baseline:** Our initial implementation uses top- p nucleus sampling at inference with no padding mask on the loss and no early stopping, saving the last checkpoint each epoch.
- **Training Fixes:** We applied four improvements to produce the GPT-2 Fine-tuned checkpoint: a padding-aware loss mask excluding padding tokens from cross-entropy; a 90/10 train/validation split with early stopping (patience=5) saving the best checkpoint by validation loss; a newline token (token 198) appended to the prompt; and removal of gradient clipping and label smoothing, which caused regression on this small dataset.
- **Inference-Time Extensions:** Using the same fine-tuned checkpoint without retraining, we compared two decoding strategies. GPT-2 Fine-tuned + Beam Search replaces top- p

sampling with deterministic k -best beam search, with a repetition penalty excluding the newline token. GPT-2 Fine-tuned + NeuroLogic applies constrained beam search enforcing the rhyme scheme via soft phonetic constraints, with EOS masked until 14 lines are complete.

All systems use AdamW with weight decay 0.01 and a linear learning rate scheduler with 10% warm-up, trained on an L4 GPU. A beam size ablation ($k \in \{5, 10, 20\}$) identified $k = 10$ as optimal for GPT-2 Fine-tuned + Beam Search¹

7 Results

7.1 Sentiment Analysis

Model	Dataset	LR	Batch Size	Dev Acc.
Last-linear-layer	SST-5	10^{-3}	64	0.452
Full-model	SST-5	10^{-5}	64	0.514
Last-linear-layer	CFIMDB	10^{-3}	8	0.886
Full-model	CFIMDB	10^{-5}	8	0.980

Table 1: Development accuracy under last-linear-layer vs. full-model fine-tuning.

Full-model fine-tuning substantially outperforms frozen-backbone training on both datasets. The improvement is more pronounced on CFIMDB (0.094 absolute) than SST-5 (0.062), likely because CFIMDB’s binary classification setting is better suited to end-to-end adaptation.

7.2 Paraphrase Detection

Performance metrics on the development (dev) set and final test set are summarized in the table below.

Model Size	Method	LR	Epochs	Dev Acc.	Test Acc.
GPT-2 Small	Full Fine-tuning	10^{-5}	10	0.881	–
GPT-2 Medium	LoRA (Attn)	10^{-4}	3	0.876	–
GPT-2 Medium	LoRA (MLP)	10^{-4}	3	0.873	–
GPT-2 Medium	LoRA (All-Linear)	10^{-4}	5	0.890	0.870
GPT-2 Large	LoRA (All-Linear)	5×10^{-5}	3	0.897	0.886

Table 2: Paraphrase Detection results showing the impact of model scaling and LoRA target selection.

The GPT-2 Medium all-linear configuration (targeting both Attention and MLP layers) achieved a test accuracy of **0.870**. Our highest-performing model, the GPT-2 Large all-linear configuration, achieved a test accuracy of **0.886** using the checkpoint from epoch 3.

7.3 Sonnet Generation

System	Decoding	Epochs	Dev ChrF	Test ChrF
Baseline	Top-p (T=1.2, p=0.9)	10	41.371	–
GPT-2 Fine-tuned	Top-p (T=1.2, p=0.9)	15	42.198	–
GPT-2 Fine-tuned + NeuroLogic	NeuroLogic ($k = 5, \lambda = 3.0, \alpha = 40, \beta = 2$)	15	41.446	41.283
GPT-2 Fine-tuned + Beam Search	Beam (k=10, rep=1.3)	15	42.329	42.379

Table 3: ChrF scores for sonnet generation systems.

All fine-tuned models use a learning rate of 1×10^{-5} , batch size 8; only the decoding strategy differs across systems.

¹Details on failed experiments, including subsequence data augmentation and LoRA on GPT-2 medium, are provided in the Appendix A.

Training fixes alone account for the largest single improvement, raising ChrF from 41.371 to 42.198 (+0.827 over the Starter Baseline). GPT-2 Fine-tuned + Beam Search achieves the highest ChrF at 42.329 on dev and 42.379 on test, a further +0.131 over GPT-2 Fine-tuned. GPT-2 Fine-tuned + NeuroLogic scores 41.446 on dev and 41.283 on test, below GPT-2 Fine-tuned despite enforcing the correct rhyme scheme, as rhyme constraints push beams toward phonetically-matched words that diverge from Shakespeare’s exact vocabulary.

System	BERTScore		Sonnet Composite	
	Dev	Test	Dev	Test
Baseline	0.898	-	0.67	-
GPT-2 Fine-tuned	0.901	-	0.685	-
GPT-2 Fine-tuned + NeuroLogic	0.902	0.901	0.680	0.680
GPT-2 Fine-tuned + Beam Search	0.896	0.895	0.703	0.711

Table 4: BERTScores and Sonnet Composite Scores for sonnet generation systems.

The BERTScores for our sonnet generation systems reveal the generated sonnets have very close semantic meaning to the referenced sonnets. The configurations of our various decoding methods are not rewarded in totality using BERTScores. Through our composite scores, we do see an improvement as we strengthen our systems, the highest composite score being from GPT-2 Fine-tuned + Beam Search with a score of .711.

8 Analysis

8.1 Sentiment Analysis

The gap between last-linear-layer and full-model fine-tuning is consistent across both datasets but varies in magnitude. On SST-5, the 5-class classification task benefits modestly from end-to-end adaptation (+0.062), suggesting that the frozen GPT-2 representations already capture sufficient sentiment signal for coarser distinctions. On CFIMDB, the improvement is larger (+0.094), likely because the longer, more polar reviews in CFIMDB require deeper adaptation of the contextual representations to capture document-level sentiment rather than phrase-level cues.

8.2 Paraphrase Detection

Our analysis revealed three primary insights regarding the behavior of LoRA-adapted models for classification tasks:

Target Layer Combinations: While LoRA on Attention layers (0.876) slightly outperformed LoRA on MLP layers (0.873) in isolation, combining both sets of targets provided a significant boost in performance, allowing the Medium model to surpass the Small model’s full fine-tuning baseline within fewer epochs.

Prompt and Syntax Sensitivity: The model exhibited extreme sensitivity to the exact syntax of the training prompt. During training, a closing quote was inadvertently omitted in the prompt template of the given code ("... *Question 2: "s2 ..."*). Reintroducing this quote ("... *Question 2: "s2" ..."*) during inference dropped the dev accuracy from 0.897 to 0.895. This indicates that the LoRA adapters have overfitted to the micro-syntax of the training distribution rather than purely learning semantic representations.

Failure of Few-Shot Prompting: Introducing few-shot examples to the fine-tuned GPT-2 Large model caused a catastrophic performance drop (0.897 to 0.646). We hypothesize this is because the fine-tuning process transformed the generative model into a specialized classifier. The fixed linear head was trained to detect the classification signal at a specific sequence position. As such, the added few-shot context shifted the target question much later in the sequence, effectively breaking the positional alignment the classification head expected.

8.3 Sonnet Generation

ChrF measures character 6-gram overlap against Shakespeare’s original text, rewarding lexical proximity rather than structural validity. This creates a systematic misalignment with sonnet quality:

GPT-2 Fine-tuned + Beam Search achieves the highest ChrF by maximizing log-probability and gravitating toward common Shakespearean phrases, while GPT-2 Fine-tuned + NeuroLogic enforces rhyme constraints that push beams toward phonetically matched but lexically divergent words. As a result, NeuroLogic often scores lower on ChrF despite producing structurally stronger sonnets.

Qualitative analysis across the held-out set reveals a consistent pattern across all three systems. We illustrate this with two representative prompts: Sonnet 5 and Sonnet 8. GPT-2 Fine-tuned + NeuroLogic produces the strongest continuations. In Sonnet 5 it maintains the vision metaphor introduced by the prompt (“*That they behold and see not what they see?*”), continuing with lines such as:

*“And when their sight shall be seen by thee / Then do I say that thy love hath no
part / In the world of my own eye”*

preserving the conceptual thread of visual truth and perception. In Sonnet 8 it develops the emotional restraint of “*My tongue-tied patience with too much disdain*” into reflection on beauty and memory, introducing Renaissance artistic imagery consistent with the poetic tradition.

GPT-2 Fine-tuned + Beam Search produces grammatically cleaner output but drifts toward rhetorical generalization. In Sonnet 8 it broadens the personal emotional tension into abstract statements about human relationships, while in Sonnet 5 it shifts toward repeated rhetorical questioning that weakens the narrative arc. GPT-2 Fine-tuned exhibits the most severe degradation, with syntactic collapse and semantic incoherence emerging in later lines, phrases such as:

“For all I base fall short, can I live them all back again” (Sonnet 5)
“As swine unless given shape change” (Sonnet 8)

illustrating the breakdown of both grammar and thematic coherence. This occurs despite sharing the same checkpoint as the other two systems, indicating that top- p sampling at this temperature introduces harmful variance that beam-based methods suppress.

The central finding is that ChrF fails to distinguish GPT-2 Fine-tuned + NeuroLogic’s qualitative superiority from GPT-2 Fine-tuned’s failure: both score below GPT-2 Fine-tuned + Beam Search despite substantial differences in output quality. This highlights the limitations of automatic evaluation metrics for creative generation tasks.

This limitation led us to examining our generated sonnets using BERTScore as well as generating our own composite score specific to sonnet structural requirements. While ChrF examines character overlap, BERTScore looks at contextual embeddings. We chose BERTScore because of ChrF’s failure to identify increased quality in our developed sonnets. Ultimately, the sonnet generation systems

9 Conclusion

In this work, we implemented GPT-2 from scratch and fine-tuned it across sentiment analysis, paraphrase detection, and sonnet generation. For sentiment analysis, full-model fine-tuning substantially outperforms frozen-backbone training, confirming the value of end-to-end adaptation even on small datasets. For paraphrase detection, applying LoRA across all linear layers and scaling to GPT-2 Large achieves 0.886 test accuracy, demonstrating that parameter-efficient fine-tuning of larger models outperforms full fine-tuning of smaller ones. Our analysis further reveals that fine-tuned classification heads are highly sensitive to prompt micro-syntax and do not benefit from few-shot prompting, as added context disrupts the positional alignment the classification head expects.

For sonnet generation, account for the largest single improvement in ChrF (+0.827) with beam search adding a further gain at inference time. NeuroLogic Decoding enforces structural constraints invisible to ChrF, producing more thematically coherent and rhyme-adherent sonnets at a modest metric cost. The central limitation of our sonnet evaluation is that ChrF rewards lexical proximity to Shakespeare’s vocabulary and cannot capture structural or thematic quality, a gap that human evaluation makes clear.

Team contributions

Alma Cooper: Implemented the Baseline for sonnet generation, the BERTScore generation, and custom Sonnet scoring method. Contributed to the Introduction, Related Work, Evaluation Methods, Sonnet BERTScore evaluation, Analysis and Results sections of this report.

Louis Weisdorf: Implemented the paraphrase detection module and ran the full experimental pipeline across fine-tuning configurations and prompt variations. Contributed to the Abstract, Related Work, Experimental Details, Results, Analysis, and Conclusion sections of the report.

Antra Nakhasi: Implemented the GPT-2 architecture and did the sentiment analysis experiments. Implemented the sonnet generation improvements and extensions including training fixes, constrained decoding, and inference-time ablations. Contributed to the Abstract, Introduction, Related Work, Approach, Experimental Details, Results, Analysis, and Conclusion sections of the report.

References

- [1] Alec Radford. Improving language understanding by generative pre-training. 2018.
- [2] Shankar Iyer, Nikhil Dandekar, and Kornel Csernai. First quora dataset release: Question pairs. *quoradata.quora.com*, 2017.
- [3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [4] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [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. *arXiv preprint arXiv:2106.09685*, 2021.
- [6] Jey Han Lau, Trevor Cohn, Timothy Baldwin, Julian Brooke, and Adam Hammond. Deep-speare: A joint neural model of poetic language, meter and rhyme. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1948–1958, 2018.
- [7] Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Neurologic decoding: (un)supervised neural text generation with predicate logic constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4288–4299, 2021.
- [8] Carnegie Mellon University. CMU pronouncing dictionary. <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>, 1993.
- [9] Maja Popović. chrF: character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, 2015.

Appendix

A Sonnet Generation Failed Experiments

LoRA on GPT-2 Medium. We applied LoRA ($r = 16$, $\alpha = 32$, targeting `c_attn`, `c_proj`, `c_fc`) to GPT-2 medium (345M parameters) with learning rate 3×10^{-4} , beam search ($k = 10$), and repetition penalty 1.3. This yielded a dev ChrF of 40.577, below the Starter Baseline (41.371). GPT-2 medium’s diverse pretraining distribution proved too strong to override with 143 sonnets at LoRA scale, with outputs containing formatting artifacts (`\`, `^`, `--`, backticks) from the pretraining distribution. Full fine-tuning of GPT-2 small remains a stronger approach at this data scale.

Subsequence Data Augmentation. To close the train/test distribution gap (the model sees 3-line prompts at test time but full sonnets at train time) we augmented the training set by adding versions of each sonnet starting from lines 2, 3, and 4, producing approximately 524 training examples. This yielded a dev ChrF of 42.244, numerically competitive but qualitatively degraded: subsequences lack structural position context, causing the model to learn mid-sonnet continuation without awareness of its position in the 14-line structure. The fourfold repetition of overlapping fragments also amplified copying behavior. This approach was reverted.

NeuroLogic Ablations. Several modifications to the NeuroLogic configuration were explored. Adding a line-length penalty (-1.0 for lines shorter than 5 tokens) suppressed the newline token, preventing `line_count` from reaching 14 and causing generation to run to `max_length`. A penalty decay schedule (`line_lam` increasing to 6.9 at line 13) overwhelmed the language model signal. A capped variant achieved dev ChrF of 42.160 but remained qualitatively degraded with dialogue drift and missing rhymes. All ablations were reverted to the clean configuration reported in the main paper.