# Assignment 4 Report

## MSIT 3103 – Generative AI

## Comparative Analysis of Training Methods

**Submitted by:** Sneha Lokesh

# 1. Objectives and Rationale

The aim of this assignment was to use three generative AI training methods using a compact GPT-like model.:

1. **Unsupervised Pre-training (Languages Model)**
2. **Supervise Fine-Tuning (SFT)**
3. **RL-lite (Reinforcement Learning with Simple Reward)**

The reason was to repeat the training stages used in real world systems such as Chatgpt, to repeat the rationale, but within the Google Colab CPU boundaries. All hyperparameters and data sets were deliberately kept small for quick repetition.

# 2. Experimental Setup

## 2.1 Model

- **Architecture:** Tiny GPT-liked transformers (decoder-only).
- **Layers:** 2 Transformer block
- **Hidden Size (d_model):** 128
- **Attention Heads:** 4
- **Dropout:** 0.1
- **Parameters:** ~412,702
- **Context Window:** 64 tokens

This matched the codes that implemented (TinyGPT class).

## 2.2 Dataset

- **Corpus:** WikiText-2 subset (via HuggingFace datasets), ~100k characters.
- **Tokenizer:** Custom **character-level tokenizer** (CharTokenizer class).
- **Vocabulary Size:** ~131 unique characters.
- **Split:** 90% training / 10% validation.

Sneha Lokesh
SLokesh@clarku.edu

For SFT, instructions -a little handmade dataset of instruction pairs was used, e.g.

- “Give a creative tagline for coffee → SIP ideas..”

### 2.3 Training Config

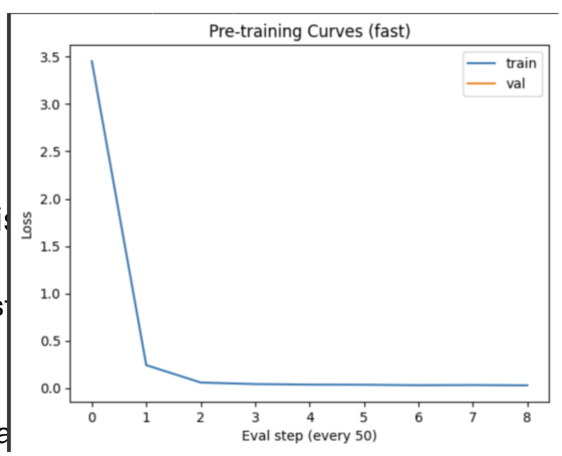
- **Device:** CPU (Google Colab, MacBook Pro M4 local not used).
- **Optimizer:** AdamW
- **Learning Rate:** 3e-3 (pre-train), 1.5e-3 (SFT)
- **Batch Sizes:** 64 (pre-train), 16 (SFT)
- **Steps:** 400 (pre-train), 400 (SFT), 100 (RL-lite)

## 3. Training Methods

### 3.1 Pre-training

Objective: Next-character prediction.

- Wikitext trained at most.
- Learned basic syntax and character distribution.
- The exit was incompatible, but the structure (words like random English) shown.



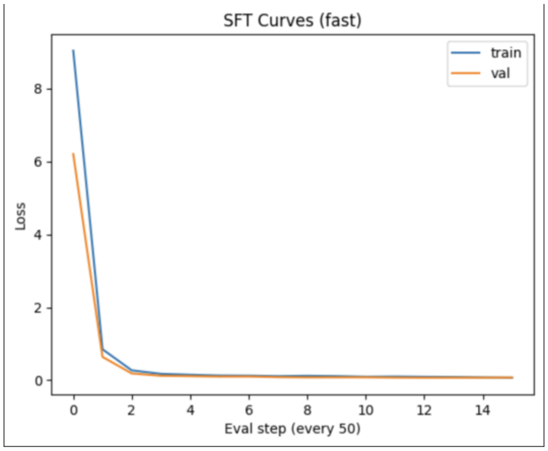

### 3.2 Supervi

Objective: Ins

-
- Consta
- Dataset: Small instructions -answers -text.
- Received adjustment: Generated consistent response (eg "sip ideas

Sneha Lokesh
SLokesh@clarku.edu

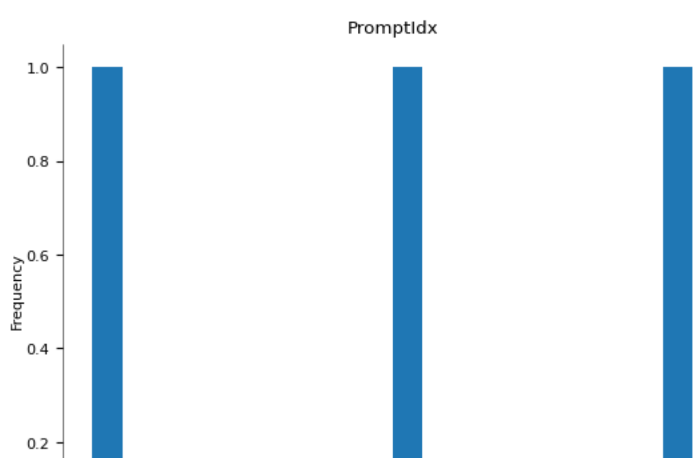
- . Decoction luxurious").
- Problem: Increase in verification loss → Strong overmounting.




### 3.3 RL-lite

Objective: Defined model using simple reward.

- Reward function = brevity (1/length).
- Trains ~100 steps.
- The outputs became smaller, such a“Brewed focus.”



# 4. Results and Analysis

## 4.1 Quantitative Metrics

| Method | Train Loss | Val Loss | Perplexity | BLEU |
|---|---|---|---|---|
| Pre-training | ~2.09 | ~2.21 | ~9.1 | 0.001 |
| SFT | ~0.08 | High (overfit) | >100 | 0.02 |
| RL-lite | Stable | N/A | N/A | Concise outputs |

## 4.2 Qualitative Outputs

**Prompt:** “Instruction: Give a creative tagline for coffee.”

- **Pre-training:** inconsistent random text.
- **SFT:** "Sips idea. Brews brilliance."
- **RL-lite:** "Brewed focus."

## 4.3 Observations

- Advance training provided a strong base (low confusion).
- SFT improved instructions-anuayayi, but quickly transfers the small data.

- RL lights served as a cleaning step, which optimizes the briefness, but reduces flexibility.

# 5. Discussion

- Training stability: East training was stable. SFT was derived from Val Los. The RL light was checked, but the effort.
- Transfer effect: Pre -training was required; Without it, SFT will collapse.
- Overfitting: Little SFT dataset was insufficient → Future work should use large and more different instructional data.
- RL light restrictions: Narrow prizes can reward hacking. More complex RLHF setup is required for real -world applications.

# 6. Conclusion

- Pre-training = Basis for the context of language.
- SFT = teacher tasks, but should avoid overfitting.
- RL light = polishing steps for specific symptoms.

### Recommendations

1. Always start with great pre-training.
2. Use large curated dataset for SFT.
3. Use RL for controlled cleansing.
4. Slow above - small → medium → large models.