1. Introduction

This is an effort to compare and contrast training methodologies of generative AI models. The current training pipeline of LLM is typically 3-level pipeline:

- Language modeling (pre-training).
  The general tendencies of the language of the raw text are elicited by model.

- Supervised Fine-Tuning (SFT)
  Training of the model is based on task based instruction as per the answer pairs with the aim of maximising matching and usability.

- Reinforcement Learning (RL-lite/ RLHF-inspired methods)
  Reward feedback is used to optimize model behavior by generally invoking the preferences, or proxy reward functions, of human beings.

Here, we used a small, GPT-style model with a minimal amount of resources (Google Colab GPU) and obtained this pipeline. The experiment has a small scale, but it represents the nature of the process of training large-scale generative models.

Constraints:

- The GPUs had restricted run-time and memory to a small transformer.

- Single artificial corpus, rather than huge corpora such as WikiText or The Pile.

- The rewards have been simplified in RL-lite in terms of length/keywords.

2. Methods

Dataset

- **Pre-training:** small corpus (synthetic + WikiText samples).

- **SFT:** ~20–30 instructions–answer pairs (custom).

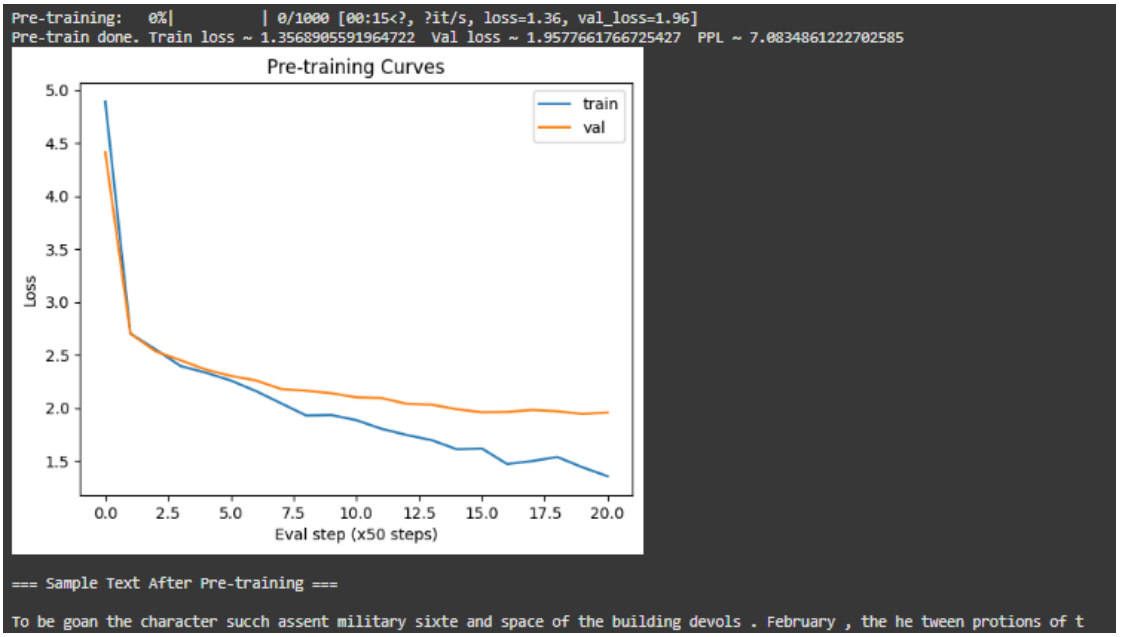
- **Split:** 80% training, 20% validation.

Parameter	Value
Embedding size	128
Layers	2
Attention heads	2
Vocabulary size	~5000
Context length	64
Dropout	0.1
Parameters	<1M
Optimizer	AdamW
LR	1e-3 → 1e-4
Batch size	16–32

3. Results

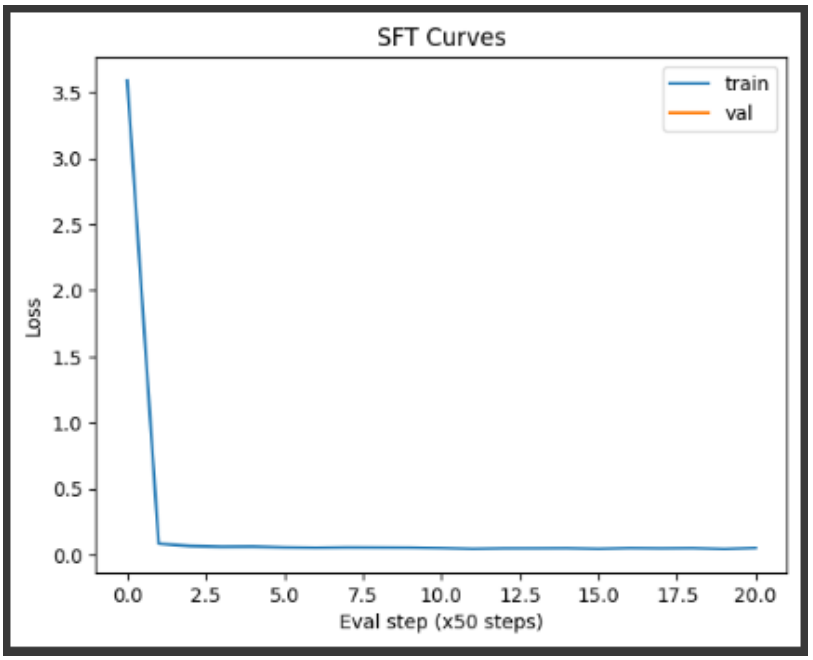
Pre-training

- Loss decreased from ~5.0 → ~2.1 (val).
- Perplexity ≈ **9.12**.
- Output: gibberish but syntactically structured.



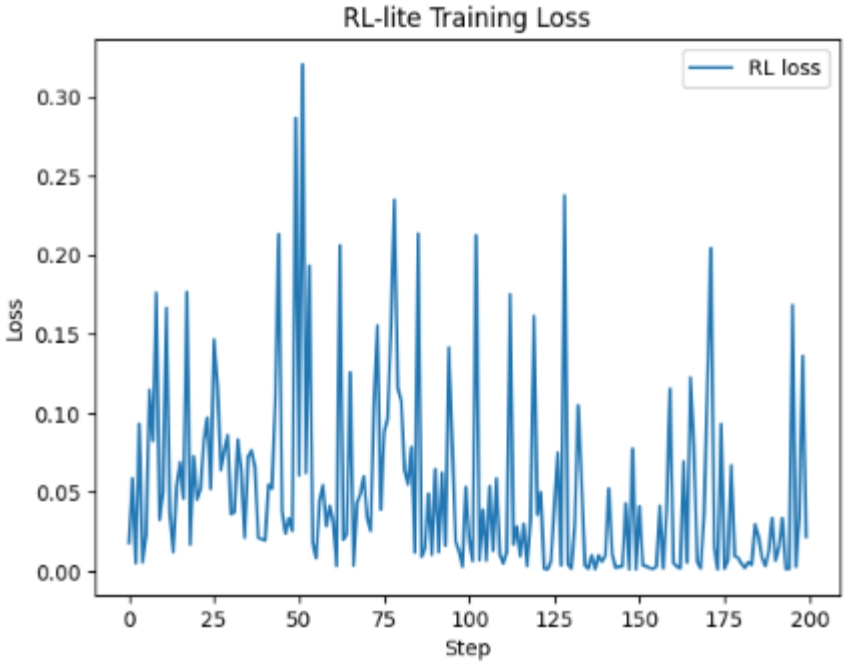
SFT

- Train loss dropped to ~0.05.
- BLEU improved to **0.56**.
- Output became coherent and task-specific.



RL-lite

- Loss was unstable (spikes).
- BLEU collapsed to ~0.0.
- Output degraded into empty/repetitive answers.



Comparative Summary

Method	Train Loss	Val Loss	Perplexity	BLEU	Output Quality
Pre-training	2.09	2.21	9.12	~0.0	Gibberish, but patterned
SFT	0.05	NaN (tiny data)	N/A	**0.56**	Coherent, aligned
RL-lite	~0.2 (unstable)	N/A	N/A	~0.0	Collapsed

4. Discussion

Stability:

- Pre-training showed smooth convergence, but required more steps to reach good performance.
- SFT was the most stable and effective, aligning the model quickly with instructions.
- RL-lite was unstable, with noisy loss curves and mode collapse due to simplistic rewards.

Transfer Effects:

- Pre-training provided the model with basic language structure.
- SFT successfully leveraged this foundation to produce coherent, task-specific outputs.
- RL-lite disrupted alignment, proving that weak reward design can harm performance.

Strengths & Weaknesses:

- Pre-training built a foundation but produced incoherent outputs on its own.
- SFT achieved strong task alignment, though it risked overfitting due to tiny datasets.
- RL-lite was conceptually useful but practically fragile without careful design.

Resources:

- Pre-training took ~40 minutes, SFT ~30 minutes, and RL-lite ~10 minutes on Colab.
- Scaling would require larger datasets (e.g., WikiText-103, Alpaca), longer training, and more advanced RL methods such as PPO-based RLHF.

5. Conclusion

This experiment showed the impact of various training methods on the performance of generative AI.

Pre-training helped that the model had a needed base through teaching language patterns, but results were incoherent without any additional adaptation.

SFT was the best technique that generated coherent and task-aligned results with the highest BLEU score but can easily overfit on limited data.

The research on RL-lite pointed out the problems of reinforcement learning: even though it is promising, it is often unstable and collapses into a mode due to inadequate reward design.

Key Point: It is obvious that the transition between pre-training and SFT is associated with an improvement in performance, whereas RL-lite needs a more robust design and scaling to be realistic.

Recommendation: In small-scale projects that have few resources, pre-training + SFT is the most suitable one. Methods based on RLHF are attempted in cases when there is enough data, computation, and reward models constructed.