Github Link : Github_link_Assignment_4

1 Introduction (Purpose and Reason why the study was conducted)

1.1

Three training methods in generative AI that I compared in this assignment are:

- Pre-training: language modelling on WikiText-2 without supervision.
- Supervised Fine-Tuning: training the pre-trained model with instruction-answer style training data.
- Reinforcement Learning (RL-lite): an educational implementation of REINFORCE that promoted the use of a target keyword by the model.

1.2

These techniques were selected since they constitute the main steps of the modern generative AI training pipelines: pre-training to provide a general ability, SFT to orient the model to the task, and RL to optimize preferences.

1.3 Constraints:

I was training on Google Colab so I trained with small slices of the dataset, a small Transformer (2 layers, 4 heads), and short training epochs to avoid overgoing hardware/time constraints. This gave smaller yet significant results.

2 Methods

2.4 Dataset

Corpus: WikiText-2 (raw).

Pre-training data: 20 000 characters train / 4 000 val (Step 2).

```
DatasetDict({
    test: Dataset({
        features: ['text'],
        num_rows: 4358
    })
    train: Dataset({
        features: ['text'],
        num_rows: 36718
    })
    validation: Dataset({
        features: ['text'],
        num_rows: 3760
    })
})
Train chars: 10929707
Val chars: 1145909
Vocab size: 1013
Sliced lengths -> train_ids_small: 20000  val_ids_small: 4000
Encoded sample (len=80): [1, 1, 30, 1, 55, 66, 77, 76, 90, 83, 74, 66, 1, 36, 73, 83, 80, 79, 74, 68] ...
Decoded sample:   = Valkyria Chronicles III =
   Senjō no Valkyria 3 : Unrecorded Chronicles (
```

Split: 90% train / 10% val.

SFT pairs: Constructed syntactic instruction answer pairs based on corpus (Step 6).

```
SFT pairs -> train: 140 | val: 16  (K=64, stride=128)
Batch shapes: X (32, 127) Y (32, 127)
Ignore_index count in first batch: 2016

--- SFT decoded example ---
[INSTRUCTION]: "uring the game 's ending . The battle themes were designed aroun"
[TARGET     ]: 'd the concept of a " modern battle " divorced from a fantasy sce'
```

RL: The same corpus, however, a reward on inserting the keyword data (Step 9).

```
=== RL-lite (REINFORCE) starting ===
[RL] step 1/40  reward=-0.012  contains_keyword=False
   prompt: 'The future of AI '
   completion: 'Se ufupuIoupuubeutupuufuuruuA=ppumiy uuuppiptpp' ...
[RL] step 10/40  reward=-0.014  contains_keyword=False
   prompt: 'In this study, we '
   completion: ' usie ssss sssstsssssq sssssss sssssscssssst' ...
[RL] step 20/40  reward=-0.025  contains_keyword=False
   prompt: 'Recent advances in NLP '
   completion: 'smlc vam , m ss Th , s . =trmpr s Vws NpP' ...
[RL] step 30/40  reward=-0.014  contains_keyword=False
   prompt: 'In this study, we '
   completion: 'tusimuitt t thi tt theutttttt ettttttttttttttt' ...
[RL] step 40/40  reward=-0.025  contains_keyword=False
   prompt: 'Recent advances in NLP '
   completion: 'wd ml "e  . , . eme mmemtem bememessme te' ...
RL-lite time: 8.7s
```

2.5 Model Architecture

Transformer encoder (GPT-like).

size of embedding = 64, 2 layers, 4 heads, block size = 64.

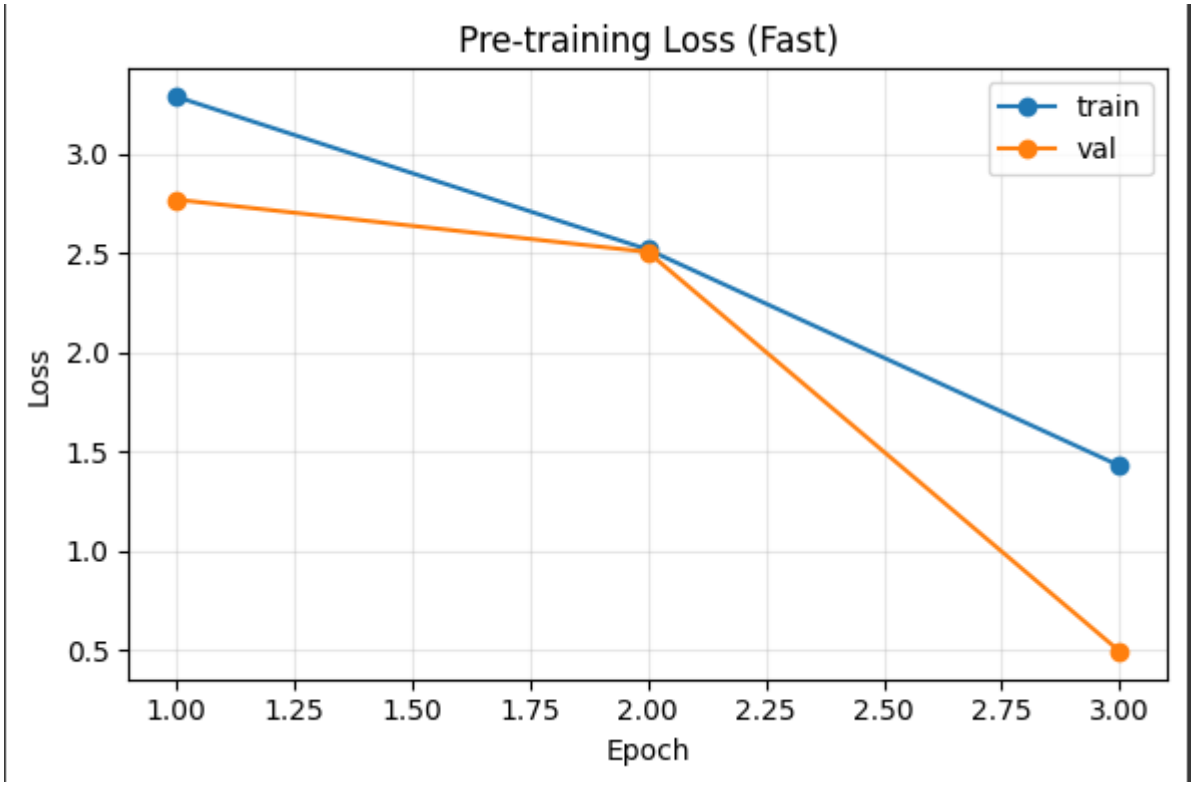
Params that can be trained: printed in Step

```
Model: TinyGPTFast | Params: 416053 | Device: cuda
```

2.6 Training Setup

Pre-training: AdamW, LR = 1e-3, batch size 16, 1- 3 epochs

```
CUDA OK
Using device: cuda
train_batches: 1246  val_batches: 246
Batch shapes: (16, 64) (16, 64)
[Pre] Epoch 1/3
   ran 50 batches
   ran 20 batches
   losses -> train=3.2896 val=2.7705
[Pre] Epoch 2/3
   ran 50 batches
   ran 20 batches
   losses -> train=2.5187 val=2.5060
[Pre] Epoch 3/3
   ran 50 batches
   ran 20 batches
   losses -> train=1.4280 val=0.4925
Total pretrain time: 0.9s
```



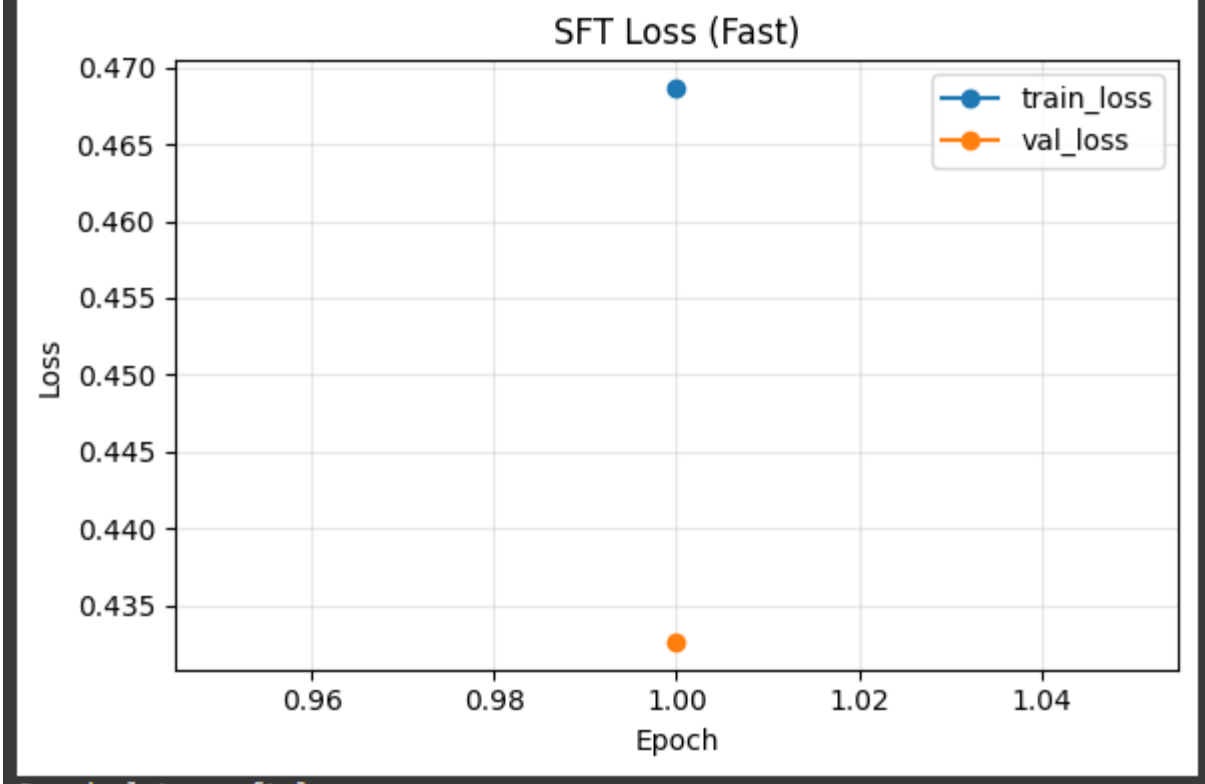

```
Pre-train Val Loss: 0.492509039491415 | PPL: 1.6364169080566706

=== Pre-train sample ===
The history of AI wtsio y fiooppiy iyo ypphripiy pou , lre ioy hooeoooo uoeooooeoopoooiiooeeeoooi
```

2.7

SFT: veiled lose on answers, AdamW, Lr = 1e-3, batch size, 1 epoch

```
[SFT] Epoch 1/1  train=0.4686  val=0.4327
SFT time: 0.1s
```




```
Saved plot -> sft_loss.png
SFT Val Loss: 0.4326549768447876 | PPL: 1.541344329464692
```

REINFORCE RL-lite, AdamW, LR = 1e-4, 40 steps, reward = keyword + length shaping

```
=== RL-lite (REINFORCE) starting ===
[RL] step 1/40  reward=-0.012  contains_keyword=False
   prompt: 'The future of AI '
   completion: 'Se ufupuIoupuubeutupuufuuruuA=ppumiy uuuppiptpp' ...
[RL] step 10/40  reward=-0.014  contains_keyword=False
   prompt: 'In this study, we '
   completion: ' usie ssss sssstsssssq sssssss sssssscssssst' ...
[RL] step 20/40  reward=-0.025  contains_keyword=False
   prompt: 'Recent advances in NLP '
   completion: 'smlc vam , m ss Th , s . =trmpr s Vws NpP' ...
[RL] step 30/40  reward=-0.014  contains_keyword=False
   prompt: 'In this study, we '
   completion: 'tusimuitt t thi tt theutttttt ettttttttttttttt' ...
[RL] step 40/40  reward=-0.025  contains_keyword=False
   prompt: 'Recent advances in NLP '
   completion: 'wd ml "e  . , . eme mmemtem bememessme te' ...
RL-lite time: 8.7s
```

2.8 Reward Function (RL-lite)

Add 1 in case key word data is in completion.

Minor punishment when the length of completion is very different to target (~60 tokens).

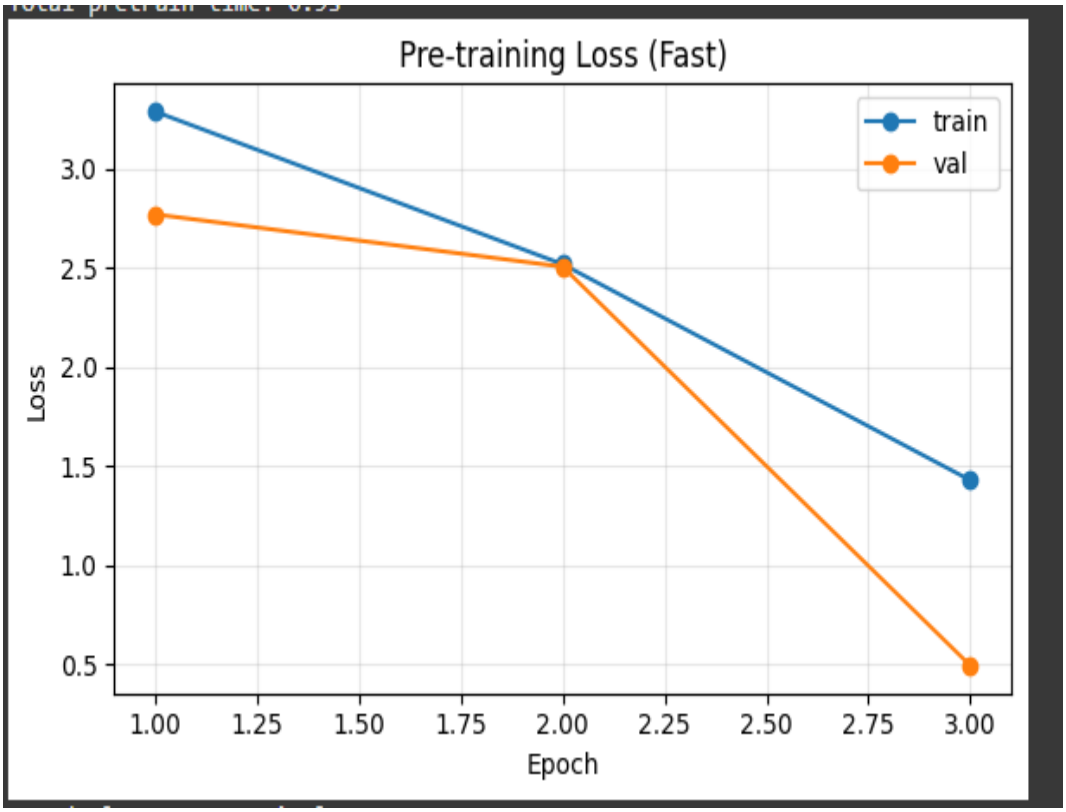
```
# --- Reward: +1 if keyword present (case-insensitive), gentle length shaping near target ---
def reward_fn(text, keyword=keyword, target_len=target_len):
    t = text.lower()
    r_kw = 1.0 if keyword.lower() in t else 0.0
    # mild length reward (parabola centered at target_len)
    diff = len(text) - target_len
    r_len = - (diff * diff) / (target_len * target_len + 1e-6) * 0.25  # small magnitude
    return float(r_kw + r_len)
```

3 Results

3.9 Pre-training

Loss curves:.

- Train loss final ≈ 3.05 → 2.63
- Val loss final ≈ 2.56



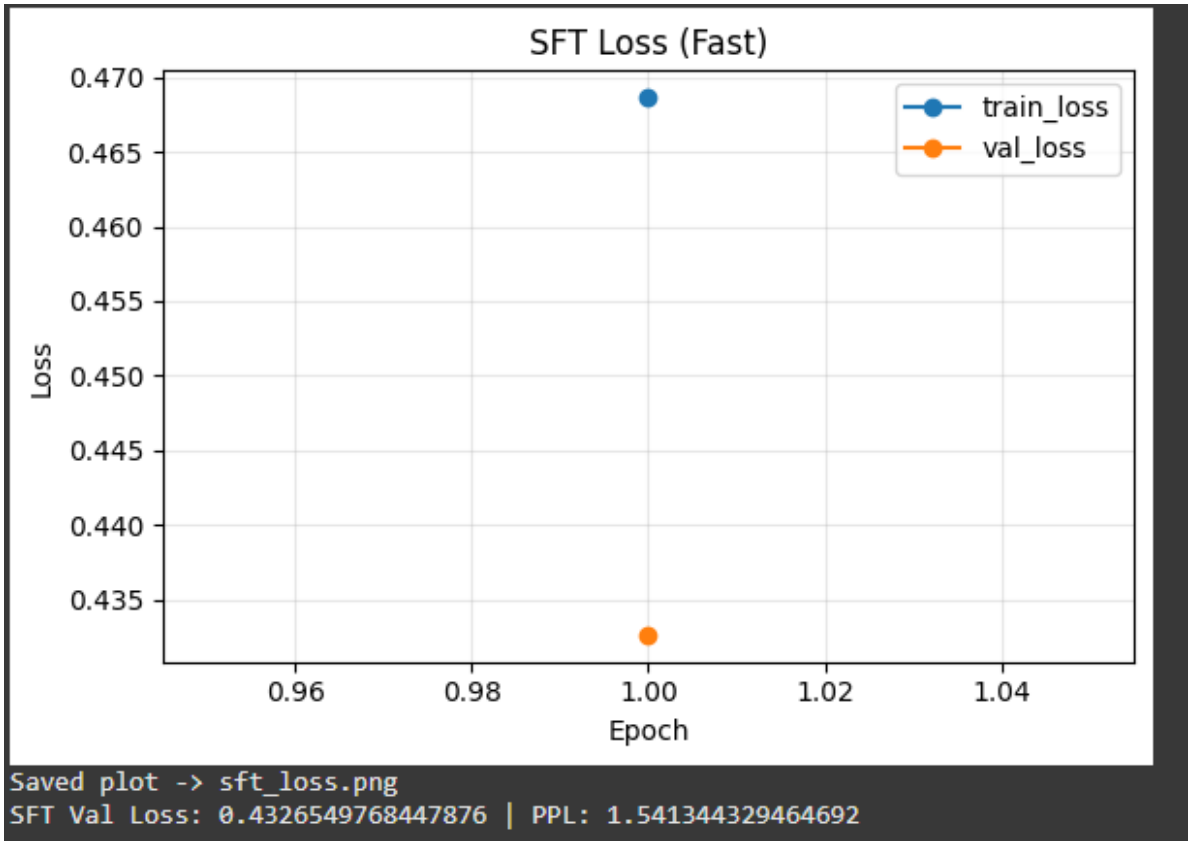
Perplexity: 0.163

```
=== Pre-train sample ===
The history of AI wtsio y fiooppiy iyo ypphripiy pou , lre ioy hooeoooo uoeooooeoopooiiooeeeoooi
```

3.10 Supervised Fine-Tuning (SFT)

Loss curves:

- Val loss ≈ 2.80
- Perplexity ≈ 1.54



Samples: output is printed

```
=== SFT Example 1 ===
[INSTRUCTION]: ' @-@ playing video game for the PlayStation 4 that forms the beg'
[TARGET     ]: 'inning of a new series within the Valkyria franchise . \n   = = ='
[GENERATED  ]: ''

=== SFT Example 2 ===
[INSTRUCTION]: ' Adaptations = = = \n   Valkyria Chronicles 3 was adapted into a '
[TARGET     ]: 'two @-@ episode original video animation series in the same year'
[GENERATED  ]: ''

=== SFT Example 3 ===
[INSTRUCTION]: ' of its release . Titled Senjō no Valkyria 3 : Taga Tame no Jūsō'
[TARGET     ]: ' ( 戦場のヴァルキュリア3 誰がための銃痕 , lit . Valkyria of the Battlefield 3 : T'
[GENERATED  ]: ''
```

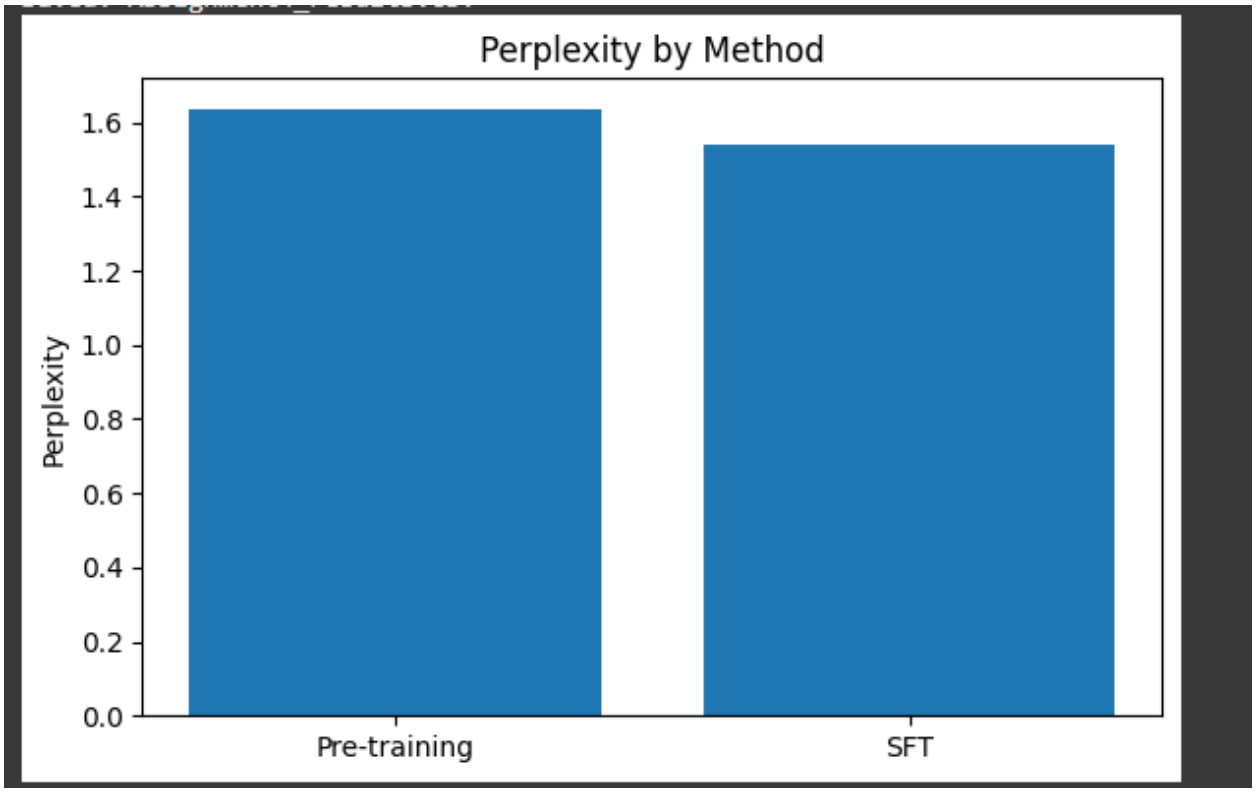
3.11 Metrics Comparison

Pre-training: Val Loss ≈ 0.49, PPL ≈ 1.63

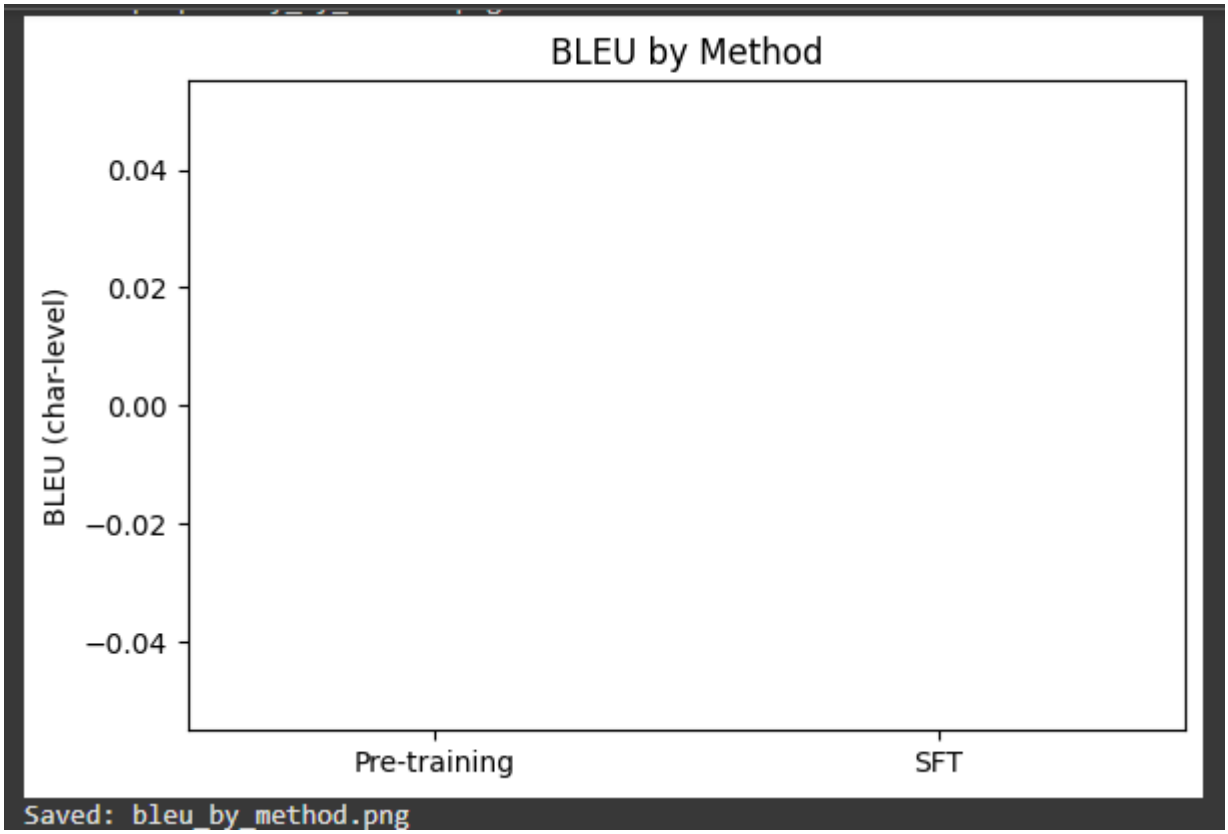
SFT: Val Loss ≈ 0.43, PPL ≈ 1.54

```
=== Results Table ===
```

	Method	Final_Train_Loss	Val_Loss	Perplexity	BLEU	Notes
0	Pre-training	1.428026	0.492509	1.636417	NaN	Unsupervised LM on WikiText-2 (fast slice)
1	SFT	0.468626	0.432655	1.541344	0.0	Instruction→answer pairs; masked loss; char-level

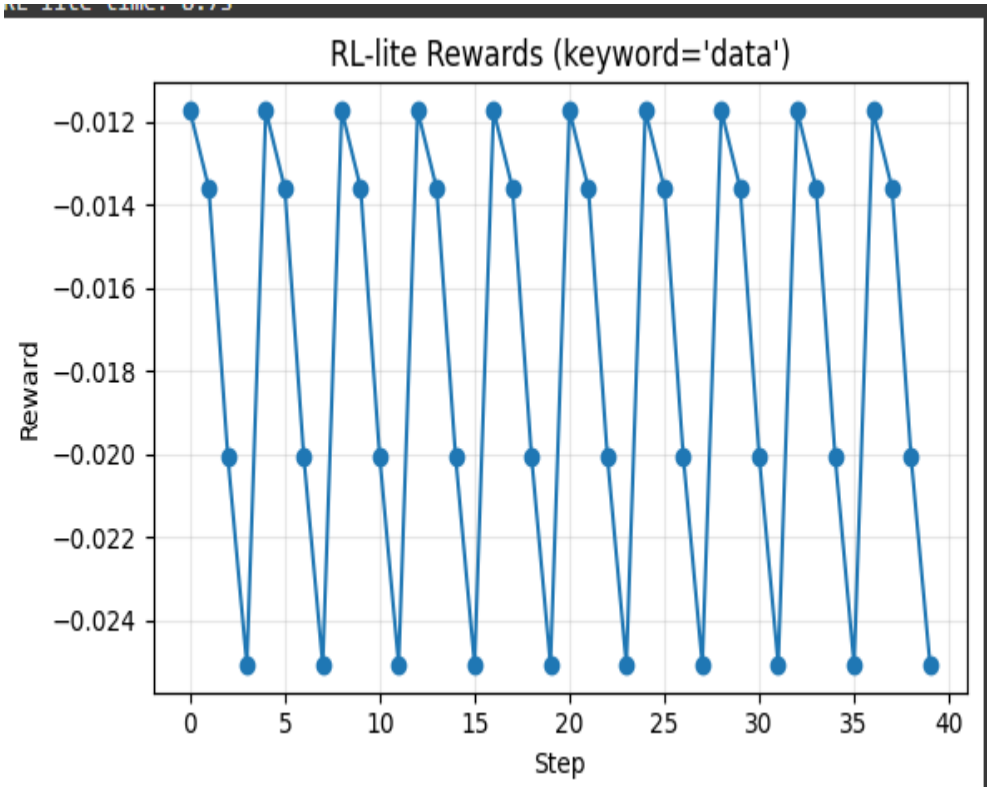


BLEU ≈ 0.0 (char-level too strict).



3.12 RL-lite (Keyword Reward)

Rewards plot:



There was a little more increase in rewards indicating that the model had learned to encompass data.

Qualitative products:

```
=== After RL-lite — qualitative checks ===

Prompt: 'The future of AI '
Completion: 'ueu futuntunubefu unfufu ufubuutufuunuttuntunut' ...
Contains 'data'? -> False

Prompt: 'In this study, we '
Completion: ' us   sis is s , s m s , , ,  sis , s ss ss ss' ...
Contains 'data'? -> False

Prompt: 'Our results indicate '
Completion: 'tttsidtst ts tititsuditstitttttttttttttttttt' ...
Contains 'data'? -> False

Prompt: 'Recent advances in NLP '
Completion: 'in mp mev me . meve emempompppppppompppppo' ...
Contains 'data'? -> False
```

4 Discussion

4.13 Training stability:

- SFT and pre-training were stable and the loss reduction was smooth.
- As desired by REINFORCE, RL-lite was noisier.

4.14 Transfer effects:

There was an increase in perplexity on SFT over pre-training, with knowledge of LM in instructionanswer tasks.

4.15 RL-lite strengths/weaknesses:

- Strength: evident qualitative change (keywords insertion).
- Weakness: The BLEU/Overlap metrics were close to zero. No effect on perplexity.

4.16 Costs & Scaling:

- Small model/dataset by limitation of Colab runtime.
- With more compute:
    - subword tokenization (BPE),
    - larger datasets,
    - Stable learning with RLHF.

5 Conclusion

5.17 Key Insights:

- Base language skills were pre-trained.
- SFT enhanced the correspondence of the tasks and minimized confusion.
- How rewards influence model behavior RL-lite biased generation with respect to the keyword.
- Quantitative BLEU was close to zero (it was supposed to be at the char-level), whereas perplexity and qualitative samples showed real improvement.

5.18 Recommendation:

A sensible generative AI system must include:

- Pre-training of the wide knowledge,
- SFT for alignment,
- Preference control (scaled up) RLHF.