1 Question 1

We omit the biases, the normalization layers and the language model head.

- **Parameters in an Attention Head:**
  - Each attention head in a transformer attention layer has three sets of weight matrices: W_q, W_k, and W_v for query, key, and value projections.
  - For each set, the number of parameters: (hidden size x hidden size), where hidden size is the hidden dimension of the model.

  It means there are 3 x (hidden size x hidden size) parameters for each attention head.
- **Parameters in Feedforward Neural Network:**
  - Each transformer layer has a feedforward neural network with two linear layers.
  - For each linear layer, the number of parameters is (hidden size x hidden size) + hidden size, or only (hidden size x hidden size).
- **Multiplying by the Number of Layers:** We multiply the number of parameters per the total number of layers in our BERT model.
- **Adding Embedding Layer Parameters:** BERT models also have an embedding layer for input tokens. The number of parameters for this embedding layer is (vocab size x hidden size).

Summing Up All Parameters, we get

$$\begin{aligned}\textbf{Nb of parameters} = \text{Total nb of layers} \times \Big[&\text{Nb of attention head} \times (3 \times (\text{hidden size} \times \text{hidden size})) \\ &+ (\text{hidden size} \times \text{hidden size})\Big] + (\text{vocab size} \times \text{hidden size})\end{aligned} \tag{1}$$

In our case, we have

- input size $= 32,000$
- hidden size $= 512$
- Nb of attention head $= 1$
- Total nb of layers $= 4$
- output size $= 32,000$

then

$$\begin{aligned}\textbf{Nb of parameters} &= 4 \times \Big[1 \times (3 \times (512 \times 512) + (512 \times 512)\Big] + (32,000 \times 512) \\ &= \textbf{20,578,304}\end{aligned} \tag{2}$$

2 Question 2

Parameter used in LoraConfig:

- **r=16** is the rank for the decomposition matrices.
- **lora_alpha=32** is the scale factor of the learned weights.
- **target_modules=["query_key_value"]** are the layers targetted by LoRA.
- **lora_dropout=0.05** is the percentage of neurons ignored at each training iteration to prevent overfitting.
- **bias="none"** tells if the bias parameters should be trained.
- **task_type="CAUSAL_LM"** to captures the cause-and-effect structure present in language.

3 Task 3: Tensorboard Visualisation

Here are the accuracies of the six models on the valid and test sets.

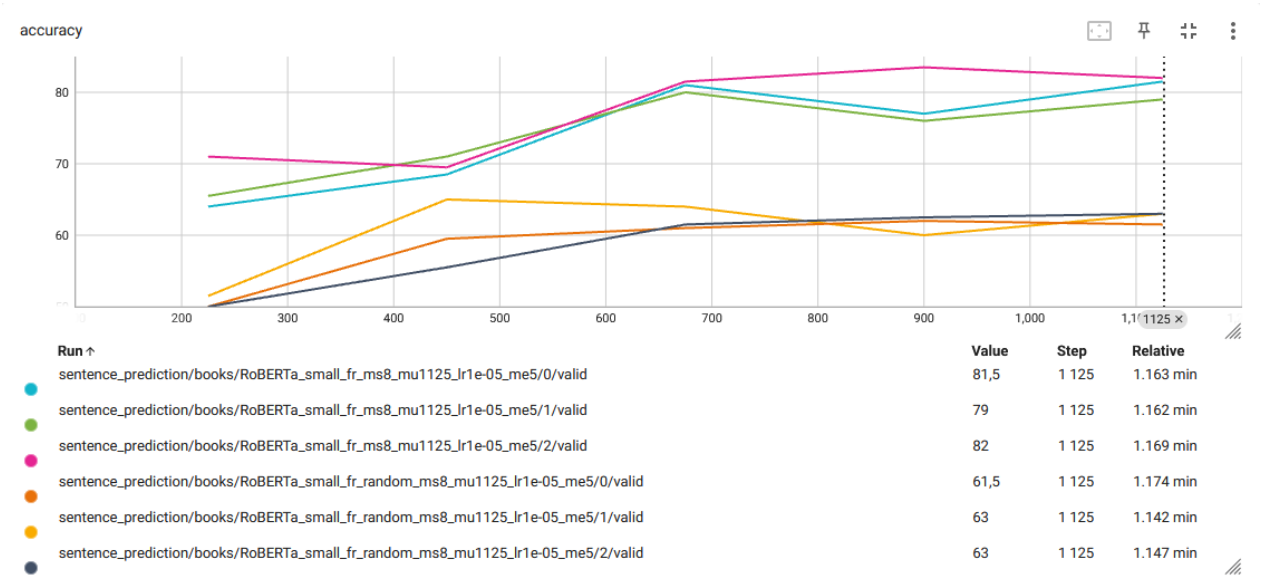

Figure 1: Accuracies of the six models on the valid set.

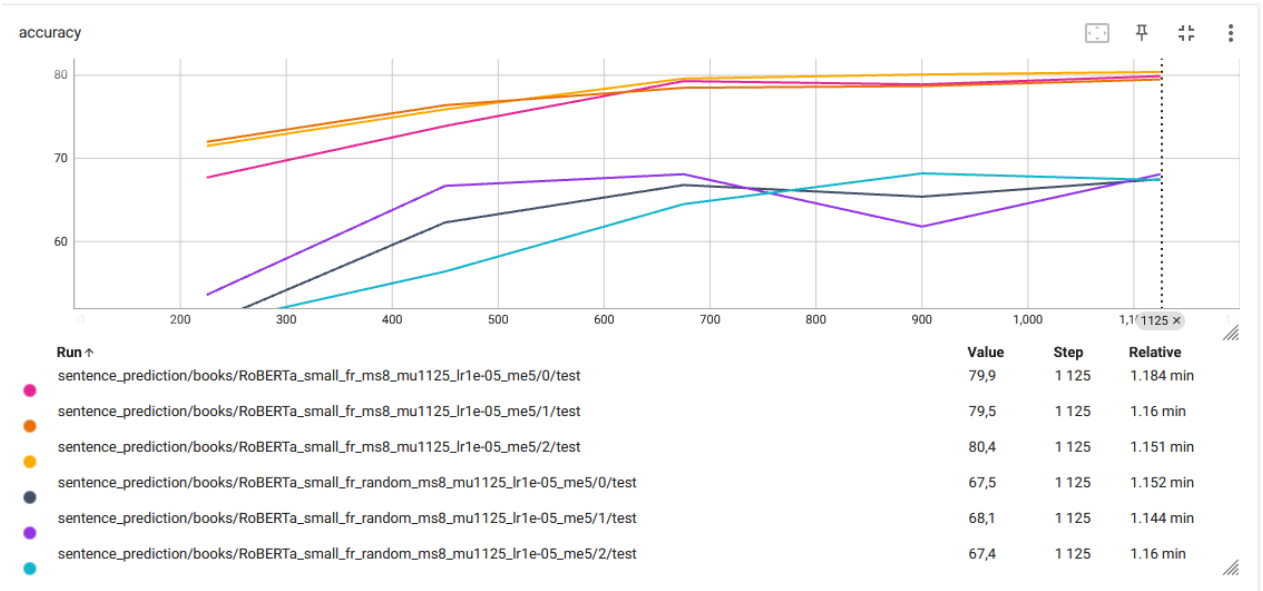

Figure 2: Accuracies of the six models on the test set.

We observe that,

- For the fine tuned roberta:
  - the best validation accuracy of seed 1 is 81.5 at step 1125 which corresponds to 79.9 in test accuracy.
  - the best validation accuracy of seed 2 is 80 at step 675 which corresponds to 78.5 in test accuracy.
  - the best validation accuracy of seed 3 is 83.5 at step 900 which corresponds to 80.1 in test accuracy.
- For the random roberta:
  - the best validation accuracy of seed 1 is 62 at step 900 which corresponds to 65.4 in test accuracy.
  - the best validation accuracy of seed 2 is 65 at step 450 which corresponds to 66.7 in test accuracy.
  - the best validation accuracy of seed 3 is 63 at step 1125 which corresponds to 67.4 in test accuracy.

We clearly see the advantages of fine-tuning. For a given number of step, the fine-tuned model will always be about 10% better than the random one.