MERRHEIM Maïssane
maissane.merrheim@telecom-sudparis.eu

1 Question 1

Let's denote x, the embedding dimension.
The model is composed of 4 transformer encoders that are composed of the following layers :

- Embedding layer : $Embedding_{params} = n_{dictionnary} \times x = 32000 \times 512 = 16384000$
- Positional embedding : $Positional_{params} = max_{tokens} \times x = 258 \times 512 = 132096$
- 4 Multi Head Attention layer: $\times MHA_{params} = 4 \times (4 \times x^2) = 4 \times (4 \times (512^2)) = 4194304$
- $4\times$ 2 fully connected layers : $4 \times 2 \times FCN_{params} = 4 \times (2 \times x^2) = 4 \times (2 \times 512^2) = 2097152$

Therefore , the total number of parameters is :

$$\begin{aligned} Total_{params} &= Embedding_{params} + Positional_{params} + 4 \times MHA_{params} + 4 \times 2 \times FCN_{params} \\ &= (n_{dictionnary} \times x) + (max_{tokens} \times x) + 4 \times (4 \times x^2) + 4 \times (2 \times x^2) \\ &= 16384000 + 132096 + 4194304 + 2097152 \\ &= 22807552 \end{aligned}$$

$$Total_{params} = 22807552$$

2 Task 3

In this task, I first fine-tuned a pretrained Roberta model and then trained a Roberta model from scratch, for 3 different seeds, using the framework fairseq. The results of this task are below :

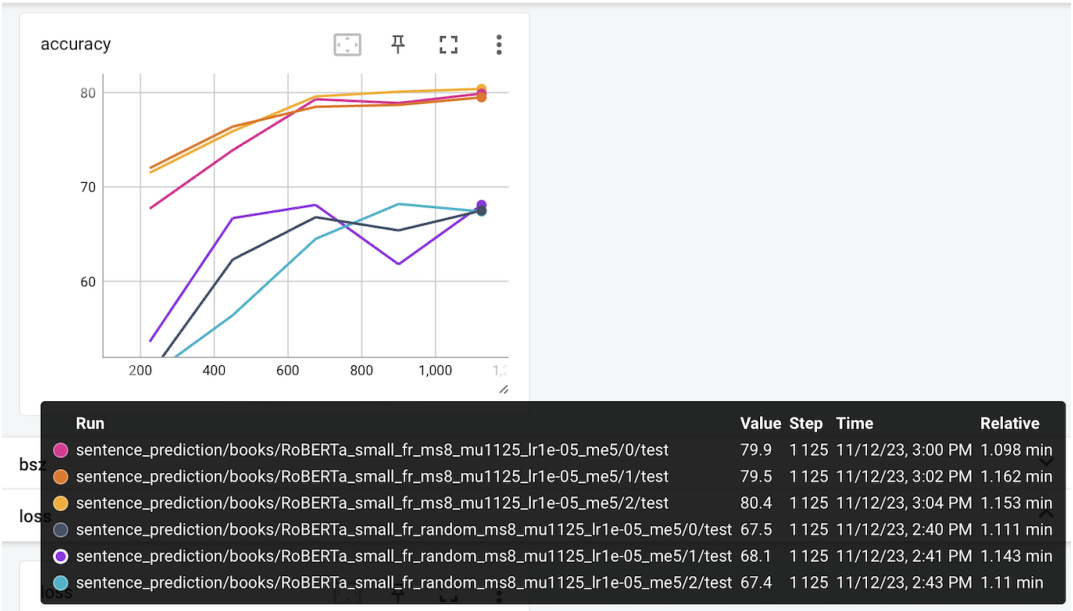

Figure 1: Accuracy across the seeds

For each seed and for each model, I plotted the accuracy on the test data. On figure 1, regardless of the seed used, the accuracy for the pretrained model, whatever the seed is , is significantly better that the non petrained model.

What's more, the seed has no meaningful impact on the the accuracy of the model throughout the epochs for the pretrained model. However, for the random model, the accuracy doesn't progress uniformly across the different seeds. In fact, for seed 1 , the accuracy increases faster than the other seeds, and reaches the final accuracy value on step 700, but then decreases brutally on step 900. For seed 2 and 3 , the accuracies evolves slowly but without abrupt changes. The final accuracy of the pretrained model is around 80%.

Models	Validation accuracy	STEP	Test accuracy
Pretrained Model (Seed 1)	81.5	1125	79.9
Pretrained Model (Seed 2)	80	675	78.5
Pretrained Model (Seed 3)	83.5	900	80.1
Random Model (Seed 1)	62	900	65.4
Random Model (Seed 2)	65	450	66.7
Random Model (Seed 3)	63	1125	67.4

Table 1: For each seed, the checkpoint with best validation accuracy

Models	Average test accuracy	Standard deviation
Pretrained Model	79.93	0.135
Random Model	67.66	0.0956

Table 2: Average accuracy and Standard deviation on the test set

3 Task 5

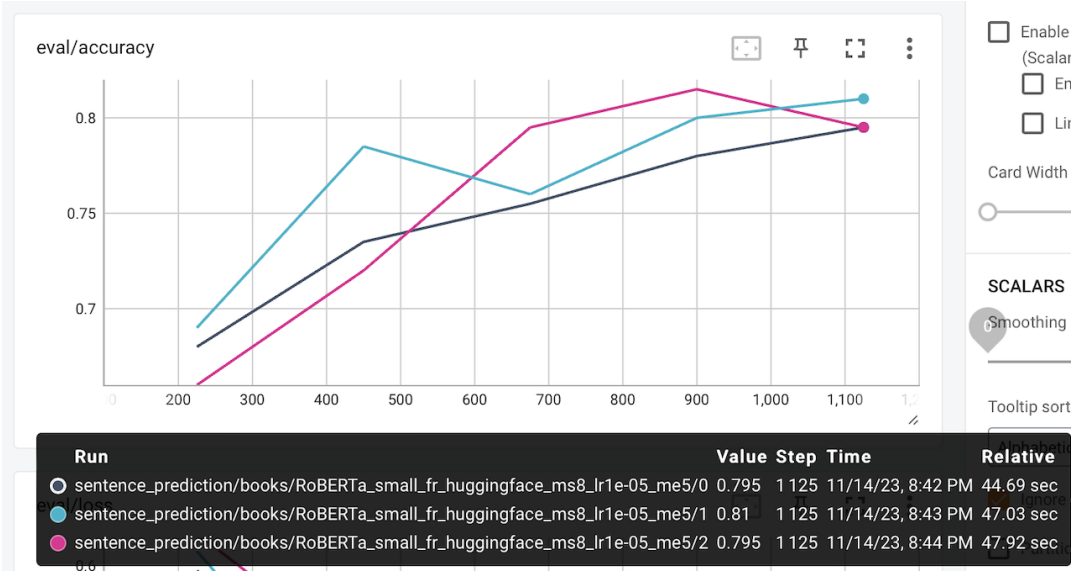

Figure 2: Accuracy across the seeds (HuggingFace) for the pretrained model

4 Question 3

LoRA model has different parameters and hyper parameters :

- $rank = 8$ to maintain smaller checkpoint sizes and to avoid artificially inflating our checkpoint files.
- $lora_{alpha} = 16$ which scales the learned weights.
- $lora_{dropout} = 0.05$
- $base_{learning-rate} = 1e-4$
- $target_{modules} = [\text{"}a_{proj}\text{"},\text{"}v_{proj}\text{"},\text{"}K_{proj}\text{"},\text{"}o_{proj}\text{"},\text{"}gate_{proj}\text{"},\text{"}up_{proj}\text{"},\text{"}down_{proj}\text{"},\text{"}embed_{tokens}\text{"},\text{"}m_{head}\text{"}]$ which corresponds to all dense layers. In LoRA original paper, it has been proven that targeting other layers than attention layers, improve the performance of the model when fine-tuning.