Task 8: Model Performance Discussion

**Q1. What are the challenges in training sequence-to-sequence models?**

Training sequence-to-sequence models presents several challenges that I encountered during this assignment. One major issue was **tokenizer consistency** — ensuring that the input and target texts were correctly tokenized with proper start (<start>) and end (<end>) tokens was crucial for accurate predictions. Any mistake in tokenization led to incorrect translations. Another challenge was the **long training time**; when I train whole dataset then it takes a lot of time approx. 45 hours per epoch even with a reduced dataset of 10,000 samples, each epoch took over 20 minutes on a CPU-based system, which made experimentation slow and time-consuming.

Additionally, model complexity increases with larger vocabularies and longer sequences, which can lead to higher memory usage and slower computation. The **size of the model**, due to embedding layers and LSTM units, also added to the difficulty of managing resources effectively. I also faced **prediction related issue:** initially, the model produced the same output for every input due to incorrect decoding logic and improper handling of token indices. Understanding and correcting the **evaluate and decode functions** required careful debugging. Another challenge was that after closing the notebook, the trained model was lost unless explicitly saved, resulting in the need to retrain it.

**Q2. What does a “bad” translation look like? Why might it happen?**

A “bad” translation typically appears as a sentence that is grammatically incorrect, semantically unrelated to the input, or filled with repeated, irrelevant, or nonsensical words. In my assignment, I observed examples where the output included unrelated or random French words that had no logical connection with the English input. In some cases, the model even repeated the same word multiple times or ignored the actual meaning of the sentence entirely.

This often happens due to insufficient training, tokenizer mismatches, or improper handling of special tokens like <start> and <end>. Additionally, if the model is underfitted or trained on a small sample size, it may not learn the correct word alignments or grammar structures. Poor decoding logic (such as not properly breaking the loop at <end> token) can also cause the model to generate unnecessarily long or irrelevant outputs. Furthermore, **lack of attention mechanisms** can lead the decoder to focus on the wrong parts of the input sentence, resulting in inaccurate translations.

**Q3. How can the model be improved further?**

To improve the performance of the sequence-to-sequence model, several enhancements can be made. Firstly, training the model on a larger and more diverse dataset would allow it to learn richer vocabulary and more complex sentence structures. Adding an atte**ntion mechanism** can significantly improve translation quality by helping the decoder focus on the most relevant parts of the input at each step. Using pre-trained word embedding**s** like GloVe or fastText may also provide better semantic understanding.

On the technical side, using **GPU acceleration** (e.g., through Google Colab or a local CUDA setup) can reduce training time and enable the training of deeper models or more epochs. Implementing **beam search** instead of greedy decoding during inference can also result in more accurate and fluent translations. Additionally, performing **regularization** techniques such as dropout, and fine-tuning **hyperparameters** (like batch size, learning rate, number of LSTM units) could help avoid overfitting and improve generalization. Finally, saving and reusing trained models using checkpoints ensures that training doesn’t have to restart from scratch.