Assignment 3: Implementing English-to-French Machine Translation Using RNN or Transformer

Objective:

Implement an English-to-French Neural Machine Translation (NMT) model using either Recurrent Neural Networks (RNN, LSTM, or GRU) or the Transformer architecture. Explore the practical application of deep learning in sequence-to-sequence tasks such as translation.

Instructions:

You are required to build a machine translation system that translates sentences from English to French. You can choose between two model architectures for this task:
1. RNN (including LSTM/GRU)
2. Transformer

You may use deep learning frameworks such as TensorFlow or Keras for implementing your model.

Tasks and Requirements:

1. Data Preparation:

- Download the English-to-French dataset :
https://drive.google.com/drive/folders/19BxJgbm1qOlPDTr9ZNsvx3s63WyqTYlG?usp=drive_link

- Preprocess the data by tokenizing, converting to lowercase, padding sequences, etc.
- Split the dataset into training, validation, and test sets.

2. Model Implementation (Choose One Architecture):

Option 1: RNN-based Model (LSTM/GRU):

- Implement an encoder-decoder architecture using RNNs, LSTMs, or GRUs.
- You should use word embeddings (either pre-trained like one-hot, word2vec, GloVe or trainable embeddings).
- Include techniques like teacher forcing during training.

Option 2: Transformer Model:

- Implement a Transformer-based model for machine translation.

3. Training the Model:

- Train the model on the English-to-French dataset.
- Choose an appropriate loss function such as cross-entropy loss.
- Use performance metrics like accuracy or BLEU score to monitor the model's performance during training.

4. Evaluation:

- Evaluate the model on a test set.
- Compute the BLEU score to assess the translation quality.

5. Report:

- Submit a brief report covering:
- A description of the chosen architecture (RNN or Transformer) and how it was implemented.
- Training/validation performance plots (loss, accuracy, BLEU score).
- Analysis of results, including any challenges encountered.
- A reflection on how different architectural choices (RNN/Transformer) affect translation quality.

6. Code Submission:

- Submit your code (properly commented) as a zip file. Ensure it is structured, modular, and easy to follow.

Grading Criteria:

The assignment will be graded based on the following criteria:

1. Data Preparation (10 points):
   - Correctly preprocesses the dataset (tokenization, padding, etc.).
   - Splits the data properly into training, validation, and test sets.

2. Model Implementation (30 points):
   - Correct implementation of either the RNN or Transformer model.
   - Proper usage of embeddings and/or attention mechanisms.
   - Architecture is well-structured and adheres to the guidelines.

3. Training & Evaluation (20 points):
   - Successful model training with appropriate loss functions and metrics.
   - Evaluation on the test set using the BLEU score.

4. Report (20 points):
   - Use IEEE conference format (LaTeX).

- Clearly explains the model, training process, and results.
- Includes performance plots and analysis.
- Discusses architectural choices and their impact on results.

5. Code Quality (20 points):
   - Well-organized, modular code with clear comments and documentation.
   - Code is easy to follow and executes without errors.