Project1: Neural Machine Translation – Urdu to Roman Urdu (15 Abs)

Objective

Build a sequence-to-sequence model using a bidirectional LSTM (BiLSTM) encoder-decoder to translate Urdu text into its Roman Urdu transliteration. The goal is to experiment with data from urdu_ghazals_rekhta and push the limits of what BiLSTM-based NMT can achieve for low-resource, poetic text.

Dataset

You will use the *urdu_ghazals_rekhta* dataset:
https://github.com/amir9ume/urdu_ghazals_rekhta

The dataset includes poetic works (Ghazals) in Urdu script, English transliteration, and Hindi script. You will extract the pairs you need: Urdu (source) → Roman Urdu (target). You may need to preprocess or build conversion rules if Roman Urdu is not directly present (e.g. derive it using transliteration rules).

Tasks

1. Preprocessing

- Clean the Urdu text: normalize characters, remove extraneous punctuation as needed.
- Define or collect rules for converting Urdu into Roman Urdu (if Roman Urdu is not directly given in the dataset).
- Tokenization: choose proper tokenization strategy for both source (Urdu) and target (Roman Urdu). Consider subword methods (e.g. Byte-Pair Encoding, WordPiece) if helpful.

2. Model Architecture

- Build a seq2seq model with a BiLSTM encoder and an LSTM decoder.
- Use 2 layers in the encoder and 4 layers in the decoder.

3. Training & Hyperparameters

- Define training, validation, and test splits (50%, 25%, and 25% respectively)
- Train the model with appropriate loss (e.g. cross-entropy) and optimizer (e.g. Adam). You must code in PyTorch.

4. Evolution / Experimentation Parameters

Students are required to conduct at least three experiments by varying one or more of the following parameters:

Parameter	Suggested Values / Ranges

Embedding dimension	128, 256, 512
Hidden size of LSTM layers	256, 512
Number of BiLSTM encoder layers	1, 2, 3, and 4
Number of decoder LSTM layers	2, 3, and 4
Dropout rate	0.1, 0.3, 0.5
Learning rate	1e-3, 5e-4, 1e-4
Batch size	32, 64, 128

5. Evaluation

- Use BLEU and perplexity score as your primary metric.
- Additionally, use character error rate (CER) or edit distance (Levenshtein).
- Provide qualitative examples: show translations from your model vs. the ground truth.

6. Submission & Guidelines

- Respect academic integrity — absolutely no plagiarism.
- Late submissions will NOT be accepted.
- You may use external GPU resources (Kaggle), but document which you used.
- If you seek help from instructor or TA, maintain professionalism and provide clear questions.

7. Deliverables

- All code (training, evaluation, preprocessing) in a well-organized repository.
- Write a Blog post and a LinkedIn report. Tag me in your LinkedIn post, this will give you more visibility.
- Deploy Final trained model and put a streamlit cover on it and make it live.

Challenge Questions (Bonus)

- Can you augment the dataset (e.g. via back-transliteration, noise injection) to improve performance?
- Try replacing BiLSTM + LSTM with xLSTM.