Project Brief — NMTS Seq2Seq Machine Model

1. Environment Setup

- Install essential libraries: PyTorch (for model training), NLTK (for BLEU scoring), tqdm (progress bars), Levenshtein (for CER calculation), matplotlib (for visualization).
- Ensure GPU runtime (e.g., Google Colab or local CUDA setup) for faster training.
- Download necessary resources like NLTK tokenizers.

---

2. Dataset Loading

- Import parallel **Urdu → Roman text pairs**.
- Clean the dataset by removing:
  - Empty or invalid lines.
  - Unwanted formatting or corrupted samples.
- Print dataset statistics (train/validation/test split sizes).

---

3. Text Preprocessing

- Normalize Urdu text:
  - Standardize Unicode characters.
  - Remove diacritics and extra symbols.
  - Handle punctuation and spacing.
- Ensure both source (Urdu) and target (Roman) texts are consistent and aligned.

---

4. Tokenizer Training (BPE)

- Train **Byte Pair Encoding (BPE) tokenizers** separately for Urdu and Roman text.
- Build vocabularies for both languages.
- Add special tokens:
  - <pad> → padding

    - <sos> → start of sentence
    - <eos> → end of sentence
    - <unk> → unknown token
- Encode/decode functions allow switching between text and token IDs.

---

5. Dataset & DataLoader

- Convert sentences into token ID sequences using the tokenizers.
- Apply **padding** so that all sequences in a batch have the same length.
- Use DataLoader with a custom collate function to prepare batches for training.
- Split into **train, validation, and test sets**.

---

6. Model Architecture (Seq2Seq)

- **Encoder**: Bidirectional LSTM that processes Urdu tokens and captures context.
- **Decoder**: Unidirectional LSTM that generates Roman text step by step.
- **Bridge layers**: Linear layers to connect encoder's final hidden states with decoder's initial states.
- **Embedding layers**: Convert token IDs into dense vectors before feeding to LSTM.
- **Output projection layer**: Maps decoder outputs to target vocabulary logits.

---

7. Training Loop

- Train the model using **teacher forcing** (decoder gets the correct previous token during training).
- Loss function: CrossEntropy (ignores padding tokens).
- Optimizer: Adam (with learning rate tuning).
- Run multiple experiments with different **hyperparameters** (embedding size, hidden size, layers, dropout, learning rate, batch size).
- After each epoch:

- Compute training loss.
- Evaluate on validation set.

---

8. Evaluation Metrics

- **BLEU Score** → measures similarity between generated and reference Roman text.
- **CER (Character Error Rate)** → edit distance between prediction and reference.
- **PPL (Perplexity)** → measures fluency/confidence of the model.
- Choose the best model based on these metrics.

---

9. Experiment Management

- Each experiment's best checkpoint is saved in a checkpoints/ folder.
- Results (BLEU, CER, PPL, hyperparameters) are logged into experiment_results.json.
- GPU memory is cleared between experiments to avoid crashes.

---

10. Data Augmentation (Optional)

- Add noise (random character insert, delete, or replace) to increase data variety.
- Augmented samples help the model generalize better.

---

11. Inference (Prediction Phase)

- Load the saved checkpoint of the best model.
- Input: Urdu sentence.
- Steps:
    1. Encode Urdu text into tokens.
    2. Pass through encoder → generate hidden states.
    3. Decoder generates Roman tokens step-by-step until <eos> is reached.

4. Convert Roman tokens back into text.

- Output: Predicted Romanized sentence.

---

12. Outputs of the Project

- **Checkpoints** of trained models for each experiment.
- **Experiment results file** with evaluation metrics.
- **Graphs/plots** of training and validation performance.
- **Romanized text predictions** for test sentences.

Screenshots:

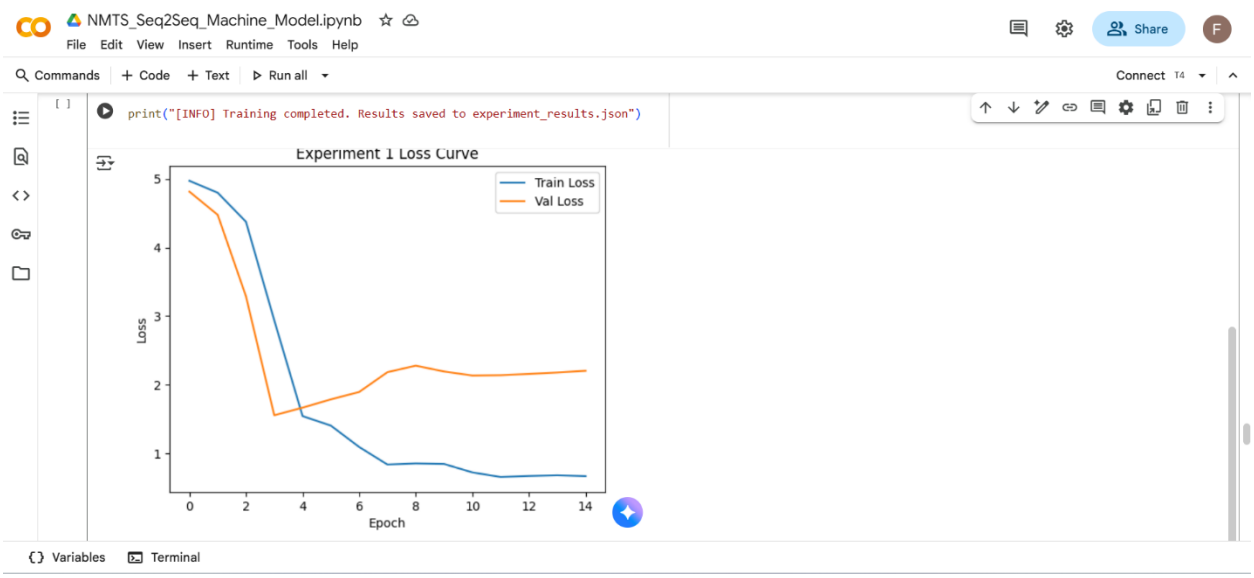
```
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (4.67.1)
Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages (3.9.1)
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from nltk) (8.2.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from nltk) (1.5.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.12/dist-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from nltk) (4.67.1)
[INFO] Libraries loaded successfully.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
[INFO] Vocab sizes: src=151, tgt=142
[INFO] Train=657, Val=328, Test=329
```

```
[INFO] Data split -> Train: 2, Val: 1, Test: 2
[INFO] Special tokens added to vocab (if missing).
[DEBUG] Source vocab size: 154, Target vocab size: 145
[INFO] Collate function defined successfully.
```

```
################################################################################
EXPERIMENT 1/3
################################################################################
Config: {'embed_dim': 128, 'hidden_dim': 256, 'enc_layers': 2, 'dec_layers': 4, 'dropout': 0.3, 'lr': 0.001, 'model': 'Seq2Seq', 'batch_size': 3
Epoch  1/15 | Train 4.9723 | Val 4.8145 | BLEU 0.0000 | CER 0.2273 | PPL 123.2908
/usr/local/lib/python3.12/dist-packages/nltk/translate/bleu_score.py:577: UserWarning:
The hypothesis contains 0 counts of 4-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
Epoch  2/15 | Train 4.7989 | Val 4.4767 | BLEU 0.0000 | CER 0.2273 | PPL 87.9451
Epoch  3/15 | Train 4.3752 | Val 3.2891 | BLEU 0.0000 | CER 0.2273 | PPL 26.8179
Epoch  4/15 | Train 2.9383 | Val 1.5544 | BLEU 0.0000 | CER 0.2273 | PPL 4.7322
Epoch  5/15 | Train 1.5421 | Val 1.6651 | BLEU 0.0000 | CER 0.2273 | PPL 5.2859
Epoch  6/15 | Train 1.4035 | Val 1.7861 | BLEU 0.0000 | CER 0.2273 | PPL 5.9664
Epoch  7/15 | Train 1.0921 | Val 1.8937 | BLEU 0.0000 | CER 0.2273 | PPL 6.6437
Epoch  8/15 | Train 0.8354 | Val 2.1826 | BLEU 0.0000 | CER 0.3636 | PPL 8.8690
Epoch  9/15 | Train 0.8520 | Val 2.2773 | BLEU 0.0000 | CER 0.5455 | PPL 9.7505
Epoch 10/15 | Train 0.8444 | Val 2.1923 | BLEU 0.0000 | CER 0.2[illegible] | PPL 8.9561
Epoch 11/15 | Train 0.7214 | Val 2.1335 | BLEU 0.0000 | CER 0.2[illegible] | PPL 8.4446
Epoch 12/15 | Train 0.6542 | Val 2.1382 | BLEU 0.0000 | CER 0.2273 | PPL 8.4843
```

```
Epoch  2/15 | Train 4.7989 | Val 4.4767 | BLEU 0.0000 | CER 0.2273 | PPL 87.9451
Epoch  3/15 | Train 4.3752 | Val 3.2891 | BLEU 0.0000 | CER 0.2273 | PPL 26.8179
Epoch  4/15 | Train 2.9383 | Val 1.5544 | BLEU 0.0000 | CER 0.2273 | PPL 4.7322
Epoch  5/15 | Train 1.5421 | Val 1.6651 | BLEU 0.0000 | CER 0.2273 | PPL 5.2859
Epoch  6/15 | Train 1.4035 | Val 1.7861 | BLEU 0.0000 | CER 0.2273 | PPL 5.9664
Epoch  7/15 | Train 1.0921 | Val 1.8937 | BLEU 0.0000 | CER 0.2273 | PPL 6.6437
Epoch  8/15 | Train 0.8354 | Val 2.1826 | BLEU 0.0000 | CER 0.3636 | PPL 8.8690
Epoch  9/15 | Train 0.8520 | Val 2.2773 | BLEU 0.0000 | CER 0.5455 | PPL 9.7505
Epoch 10/15 | Train 0.8444 | Val 2.1923 | BLEU 0.0000 | CER 0.2273 | PPL 8.9561
Epoch 11/15 | Train 0.7214 | Val 2.1335 | BLEU 0.0000 | CER 0.2273 | PPL 8.4446
Epoch 12/15 | Train 0.6542 | Val 2.1382 | BLEU 0.0000 | CER 0.2273 | PPL 8.4843
Epoch 13/15 | Train 0.6691 | Val 2.1567 | BLEU 0.0000 | CER 0.2273 | PPL 8.6427
Epoch 14/15 | Train 0.6797 | Val 2.1776 | BLEU 0.0000 | CER 0.2273 | PPL 8.8248
Epoch 15/15 | Train 0.6668 | Val 2.2026 | BLEU 0.0000 | CER 0.2273 | PPL 9.0481
/usr/local/lib/python3.12/dist-packages/nltk/translate/bleu_score.py:577: UserWarning:
The hypothesis contains 0 counts of 3-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
```



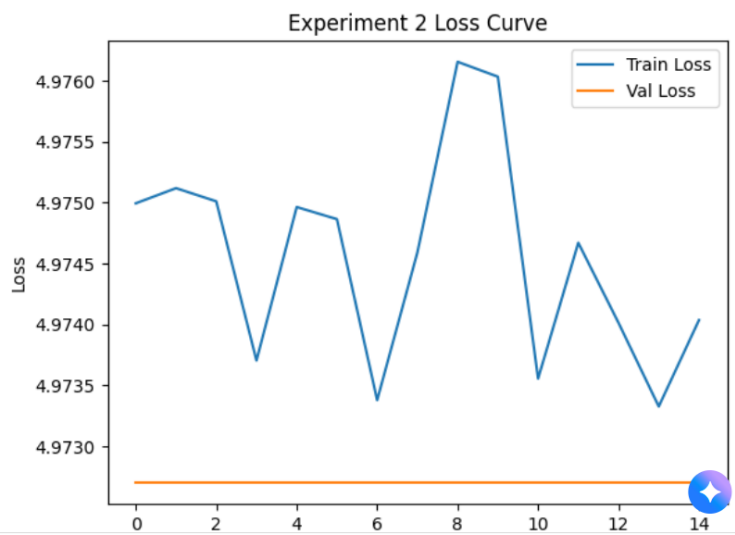
```
/usr/local/lib/python3.12/dist-packages/nltk/translate/bleu_score.py:577: UserWarning:
The hypothesis contains 0 counts of 2-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)

Experiment 1 Summary:
Test Loss=3.2136, BLEU=0.0000, CER=0.9000, Perplexity=24.8672

################################################################################
EXPERIMENT 2/3
################################################################################
Config: {'embed_dim': 256, 'hidden_dim': 512, 'enc_layers': 2, 'dec_layers': 4, 'dropout': 0.3, 'lr': 0.001, 'model': 'Seq2Seq', 'batch_size': 64}
Epoch  1/15 | Train 4.9750 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
Epoch  2/15 | Train 4.9751 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
Epoch  3/15 | Train 4.9750 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
Epoch  4/15 | Train 4.9737 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
Epoch  5/15 | Train 4.9750 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
Epoch  6/15 | Train 4.9749 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
Epoch  7/15 | Train 4.9734 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
Epoch  8/15 | Train 4.9746 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
Epoch  9/15 | Train 4.9762 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
Epoch 10/15 | Train 4.9760 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
Epoch 11/15 | Train 4.9736 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
Epoch 12/15 | Train 4.9747 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
Epoch 13/15 | Train 4.9740 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
Epoch 14/15 | Train 4.9733 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
```


```
Epoch 11/15 | Train 4.9736 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
Epoch 12/15 | Train 4.9747 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
Epoch 13/15 | Train 4.9740 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
Epoch 14/15 | Train 4.9733 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
```


```
Epoch 15/15 | Train 4.9740 | Val 4.9727 | BLEU 0.0000 | CER 0.9091 | PPL 144.4165
```



NMTS_Seq2Seq_Machine_Model.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all Connect T4

```
Experiment 2 Summary:
Test Loss=4.9691, BLEU=0.0000, CER=1.0000, Perplexity=143.8923

################################################################################
EXPERIMENT 3/3
################################################################################
Config: {'embed_dim': 256, 'hidden_dim': 256, 'enc_layers': 3, 'dec_layers': 3, 'dropout': 0.5, 'lr': 0.0005, 'model': 'xLSTM', 'batch_size': 128}
Epoch  1/15 | Train 5.0213 | Val 5.0139 | BLEU 0.0000 | CER 0.8182 | PPL 150.4888
Epoch  2/15 | Train 5.0246 | Val 5.0139 | BLEU 0.0000 | CER 0.8182 | PPL 150.4888
Epoch  3/15 | Train 5.0218 | Val 5.0139 | BLEU 0.0000 | CER 0.8182 | PPL 150.4888
Epoch  4/15 | Train 5.0166 | Val 5.0139 | BLEU 0.0000 | CER 0.8182 | PPL 150.4888
Epoch  5/15 | Train 5.0167 | Val 5.0139 | BLEU 0.0000 | CER 0.8182 | PPL 150.4888
Epoch  6/15 | Train 5.0189 | Val 5.0139 | BLEU 0.0000 | CER 0.8182 | PPL 150.4888
Epoch  7/15 | Train 5.0287 | Val 5.0139 | BLEU 0.0000 | CER 0.8182 | PPL 150.4888
Epoch  8/15 | Train 5.0162 | Val 5.0139 | BLEU 0.0000 | CER 0.8182 | PPL 150.4888
Epoch  9/15 | Train 5.0267 | Val 5.0139 | BLEU 0.0000 | CER 0.8182 | PPL 150.4888
Epoch 10/15 | Train 5.0290 | Val 5.0139 | BLEU 0.0000 | CER 0.8182 | PPL 150.4888
Epoch 11/15 | Train 5.0255 | Val 5.0139 | BLEU 0.0000 | CER 0.8182 | PPL 150.4888
Epoch 12/15 | Train 5.0149 | Val 5.0139 | BLEU 0.0000 | CER 0.8182 | PPL 150.4888
Epoch 13/15 | Train 5.0264 | Val 5.0139 | BLEU 0.0000 | CER 0.8182 | PPL 150.4888
Epoch 14/15 | Train 5.0240 | Val 5.0139 | BLEU 0.0000 | CER 0.8182 | PPL 150.4888
Epoch 15/15 | Train 5.0218 | Val 5.0139 | BLEU 0.0000 | CER 0.8182 | PPL 150.4888
```

```
Best Config: {'embed_dim': 128, 'hidden_dim': 256, 'enc_layers': 2, 'dec_layers': 4, 'dropout': 0.3, 'lr': 0.001, 'model': 'Seq2Seq', 'batch_size': 32}
Best BLEU Score: 0.0000
Best CER Score: 0.9000

Example 1:
Source (Urdu):      محبت ایک احساس ہے
Reference (Roman): mohabbat ek ehsaas hai
Generated (Roman): <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk>
CER: 2.2273
------------------------------------------------------------

Example 2:
Source (Urdu):      زندگی ایک سفر ہے
Reference (Roman): zindagi ek safar hai
Generated (Roman): <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk>
CER: 2.4000
------------------------------------------------------------
```

```
Example 1:
Source (Urdu):      پرندےزادیسےاڑتےہیاورکچ
Reference (Roman): ardnde azaadd ase udte hain aur kudch
Generated (Roman): parinde azaadi se udte hain aur kuchuh
CER: 0.1892
------------------------------------------------------------

Example 2:
Source (Urdu):      بادااسمامیہی
Reference (Roman): baadal aasmaan mein hain
Generated (Roman): baadal aasmaan mein hainaur kuchuu
CER: 0.4167
------------------------------------------------------------

Example 3:
Source (Urdu):      پولباغمیکلتےہی
Reference (Roman): phool bagh mein khilte hain
Generated (Roman): phool bagh mein khilte hainur kuchuh
CER: 0.3333
------------------------------------------------------------

Example 4:
Source (Urdu):      دلمیاکباتچیہے
Reference (Roman): dil mein ek baat chhipi hai
Generated (Roman): dil mein ek baat chhipi haiuuu
CER: 0.1111
------------------------------------------------------------
```

End of the Document