21k-4589 21k-3451 21k-3445

Language Translator (using LSTM and Transformer Models)

GitHub - AashuKumar3451/Language-Translator

Objective

To build deep learning models for automatic language translation between Urdu to French and French to Urdu using Urdu-English and English-French language pairs using both sequence-to-sequence LSTM and Transformer architectures. The project includes model training, evaluation, and inference setup for real-time translation.

Problem Statement

Languages like Urdu have limited resources for machine translation due to sparse datasets and structural differences from languages like English. This makes it difficult to build accurate translation systems. The goal is to apply modern neural network techniques to address these challenges and build models that can translate with reasonable accuracy across diverse languages.

Methodology

1. Data Preparation
- Parallel corpora for Urdu-English and English-French were used.
- Sentences were tokenized, padded, and filtered for maximum length.
- Vocabulary size and sequence lengths were defined for both source and target languages.

2. Model 1: Urdu to English Translation (Seq2Seq LSTM)
- Bidirectional LSTM Encoder processes tokenized Urdu input.
- Decoder is a unidirectional LSTM initialized with encoder states.
- Dense softmax layer predicts the next word in the English sequence.

3. Model 2: English to Urdu Translation (Seq2Seq LSTM)
- Mirrors Model 1 architecture in reverse direction.
- Embedding and LSTM layers adjusted for English input and Urdu output.

4. Model 3: English to French Translation (Transformer)
- Implements attention-based Transformer model for long-range dependencies.
- TensorFlow's high-level APIs used for defining encoder-decoder stacks.
- Custom tokenizers handle text vectorization and detokenization.

5. Training and Optimization
- Models were trained using sparse categorical crossentropy loss.
- Adam optimizer used for all models.
- **EarlyStopping** was implemented to halt training when validation loss stopped improving (patience=3).
- Additional callbacks like ModelCheckpoint and ReduceLROnPlateau were used to enhance convergence.

6. Inference
- Separate encoder and decoder models created for LSTM-based inference.
- For Transformer, input and output are generated using greedy decoding.
- Tokenizers are saved and reused to ensure preprocessing consistency.

Results

Example Translation:

```
Urdu: تم کیا کھا رہے ہو

English: eating

French: vous êtes en train de manger
```

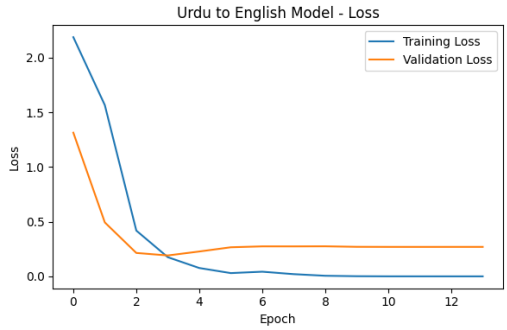
```
Urdu → English BLEU-1 Score: 0.0498
English → French BLEU-1 Score: 1.0
```

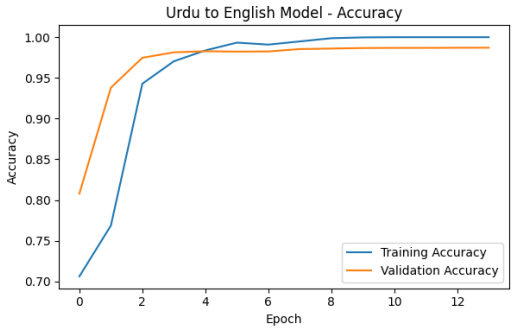
Training time:

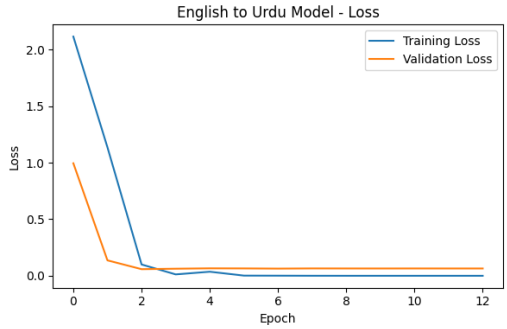
```
2500/2500 ━━━━━━━━━━━━━━━━━━━━ 142s 42ms/step - loss: 0.0541 - val_loss: 1.1872
Epoch 17/25
2500/2500 ━━━━━━━━━━━━━━━━━━━━ 111s 44ms/step - loss: 0.0541 - val_loss: 1.1874
Epoch 18/25
2500/2500 ━━━━━━━━━━━━━━━━━━━━ 142s 45ms/step - loss: 0.0540 - val_loss: 1.1820
Epoch 19/25
2500/2500 ━━━━━━━━━━━━━━━━━━━━ 141s 44ms/step - loss: 0.0537 - val_loss: 1.2014
Epoch 20/25
2500/2500 ━━━━━━━━━━━━━━━━━━━━ 142s 44ms/step - loss: 0.0530 - val_loss: 1.1877
Epoch 21/25
2500/2500 ━━━━━━━━━━━━━━━━━━━━ 142s 44ms/step - loss: 0.0532 - val_loss: 1.1894
Epoch 22/25
2500/2500 ━━━━━━━━━━━━━━━━━━━━ 111s 44ms/step - loss: 0.0531 - val_loss: 1.2040
Epoch 23/25
2500/2500 ━━━━━━━━━━━━━━━━━━━━ 142s 44ms/step - loss: 0.0519 - val_loss: 1.1994
Epoch 24/25
2500/2500 ━━━━━━━━━━━━━━━━━━━━ 142s 45ms/step - loss: 0.0529 - val_loss: 1.2047
Epoch 25/25
2500/2500 ━━━━━━━━━━━━━━━━━━━━ 142s 44ms/step - loss: 0.0519 - val_loss: 1.2086
```

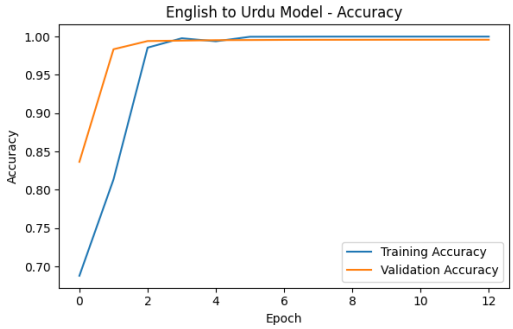
Each model took around an hour to train so 4 models took 4 hours.

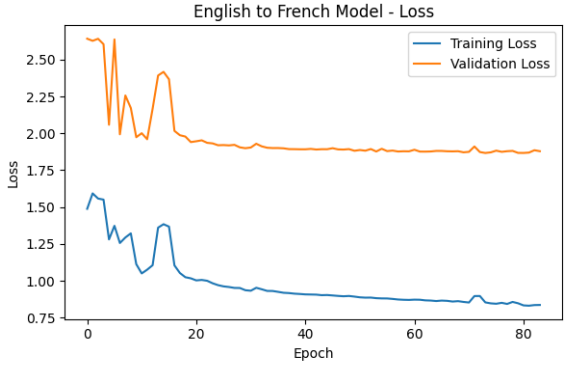
Training Loss and Accuracy:

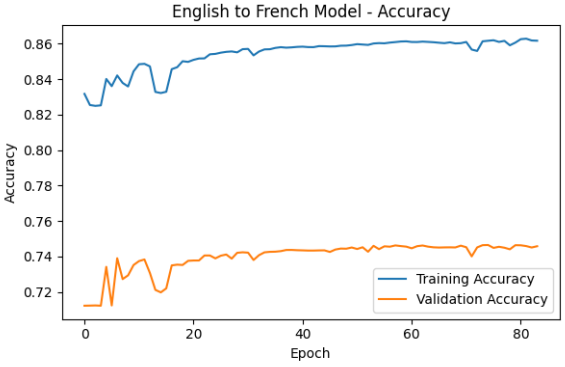


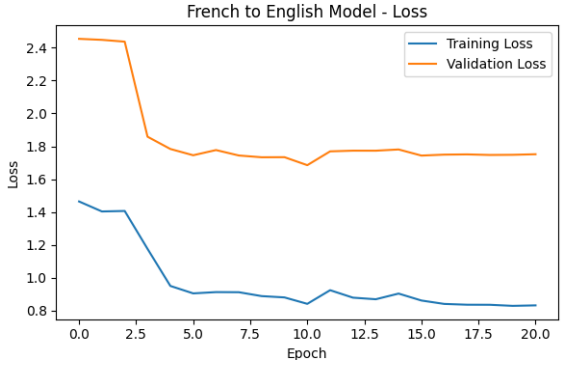


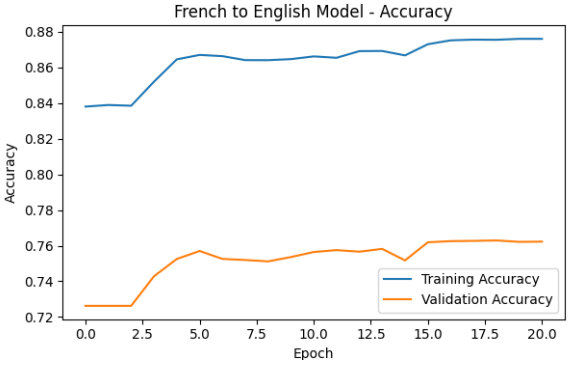












References

- https://arxiv.org/abs/1409.3215
- https://arxiv.org/abs/1706.03762

-Github Repo
GitHub - AashuKumar3451/Language-Translator