Report

Implementation of three different types of language models using PyTorch:

1. Neural Network-based Language Model (5-gram context)
2. RNN-based Language Model using LSTM
3. Transformer Decoder-based Language Model

Each model is trained and evaluated on the `Auguste_Maquet` corpus, using `GloVe 100d` pre-trained embeddings. The performance of the models is measured using perplexity scores on both training and test sets.

Preprocessing

All text was converted to lowercase for preprocessing, and trailing spaces were removed. Contractions were expanded using the `contractions` library. The corpus was then tokenized into sentences and words using NLTK tokenizers. Non-alphanumeric characters were removed, and sentences with fewer than five words were discarded. Words not found in the embeddings or those with low frequency were replaced by the `<unk>` token.

The data is split in 70:20:10 ratio for training, validation and testing respectively. The number of sentences for each model are as follows :

```
Train Sentences:  22243
Validation Sentences:  3177
Test Sentences:  6357
```

1. NNLM

```
batch_size = 64
input_dim = 500
hidden_dim = 300
output_dim = vacabulary size
num_epochs = 7
activation = Tanh
optimizer = Adam
criterion = CrossEntropyLoss
```

A batch size of 64 indicates that 64 sentences are processed together in each batch, with each sentence being broken down into potential 5-grams. To calculate the perplexity of a sentence, its 5 grams are passed through the model as a batch, and the exponential of the cross-entropy loss is computed.

Training:

```
pavani@bhashini-ilmt:/home/pavani/pav$ python3 NNLM.py
Epoch: 1/7, Train Loss: 6.081919306202478, Val Loss: 5.528446244315133
Epoch: 2/7, Train Loss: 5.439140212943521, Val Loss: 5.318350608081072
Epoch: 3/7, Train Loss: 5.2138656192419, Val Loss: 5.218830556166808
Epoch: 4/7, Train Loss: 5.0444868508358045, Val Loss: 5.163486988265487
Epoch: 5/7, Train Loss: 4.900416853368151, Val Loss: 5.130061676900572
Epoch: 6/7, Train Loss: 4.770868291110788, Val Loss: 5.114818382191112
Epoch: 7/7, Train Loss: 4.652179655502634, Val Loss: 5.108029805070423
Test Loss: 5.110942731845269
```

Perplexity

- `Train` - 99.63
- `Validation` - 224.59
- `Test` - 237.34

2. LSTM

```
batch_size = 32
input_dim = 100
hidden_dim = 300
output_dim = vacabulary size
num_epochs = 10
optimizer = Adam
criterion = nn.CrossEntropyLoss(
n_layers = 1
learning rate = 0.001
```

Padding length is taken according to the max_len of the sentence in the batch.

Training:

```
Epoch: 1, Training Loss: 2.087303280145272, Validation Loss: 1.8233987504243852
Epoch: 2, Training Loss: 1.7240240510510303, Validation Loss: 1.7056891357898711
Epoch: 3, Training Loss: 1.616623860409205, Validation Loss: 1.638766617178917
Epoch: 4, Training Loss: 1.5647099361508743, Validation Loss: 1.604404907822609
Epoch: 5, Training Loss: 1.5211364083077716, Validation Loss: 1.5836947166919708
Epoch: 6, Training Loss: 1.489096099874754, Validation Loss: 1.570010203719139
Epoch: 7, Training Loss: 1.4265068593895298, Validation Loss: 1.5598342907428742
Epoch: 8, Training Loss: 1.4225279263210022, Validation Loss: 1.5537383258342743
Epoch: 9, Training Loss: 1.3758377086842197, Validation Loss: 1.5506564682722093
Epoch: 10, Training Loss: 1.3685397188468227, Validation Loss: 1.5511965662240983
Test Loss:  1.5307415190653586
```

Perplexity

- `Train` - 79.24
- `Validation` - 175.57
- `Test` - 173.49

3.Decoder

```
batch_size = 32
input_dim = 100
output_dim = vocabulary size
num_epochs = 5
n_heads = 4
ff_dim = 300
n_layers = 1
dropout = 0.1
learning rate = 0.001
optimizer = Adam
Criterion = Cross Entropy Loss
```

Padding length is taken according to the max_len of the sentence in the whole training data.

Training:

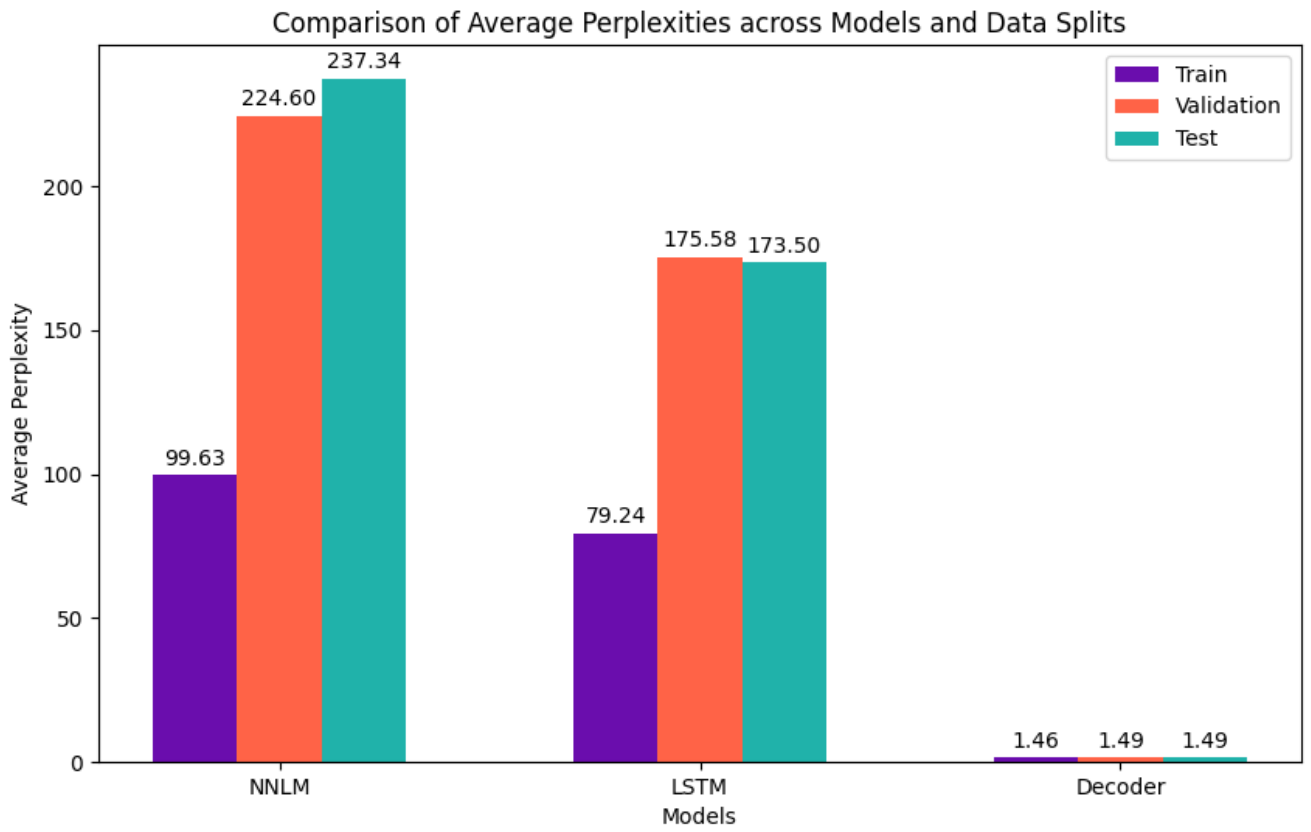
```
Epoch: 1, Training Loss: 0.6198541731841263, Validation Loss: 0.4057701727747917
Epoch: 2, Training Loss: 0.40385365092206277, Validation Loss: 0.37108911722898485
Epoch: 3, Training Loss: 0.37726960607386867, Validation Loss: 0.3515765930712223
Epoch: 4, Training Loss: 0.3600940875409321, Validation Loss: 0.33979363888502123
Epoch: 5, Training Loss: 0.34700692010422546, Validation Loss: 0.3299164465069771
Test Loss:  0.33457435382970013
```

Perplexity

- `Train` - 1.45
- `Validation` - 1.49

- `Test` - 1.48

Analysis



The performance hierarchy shows that the Transformer Decoder significantly outperforms both the LSTM and NNLM. The NNLM, limited by its 5-word context window, struggles to capture long-term dependencies, leading to poor generalization beyond short-term patterns. LSTMs, with their gated structure, can model longer-term dependencies better by maintaining an "infinite" context window. However, they still face difficulties in capturing very long-range dependencies, evident in the gap between training and test/validation perplexities. In contrast, with its self-attention mechanism, the Transformer Decoder efficiently handles both short- and long-term dependencies, resulting in consistently lower perplexities across datasets.