Assignment 1

Siya Puttagunta

2021101062

The deep learning framework used throughout the assignment is **Pytorch**. All the models are trained on `Auguste_Maquet.txt` . The pre-trained embeddings used for this assignment can be found at `glove.6B.100d.txt`. Perplexity is calculated by taking exponential of loss.

Pre-processing

- The dataset is split into train, val and test in the ratio 70:20:10.
- `nltk` is used to tokenise the corpus into sentences and they are later tokenised into words.
- Only alpha-numeric characters are retained and characters such as `\n`, `-` and `_` are replaced with a space.
- A vocab (`word2idx`) is created that stores the vocabulary of training data.
- Words neither present in `word2idx` nor in pre-trained glove embeddings are replaced with a `<unk>` token.
- A start token `<s>` and end token `</s>` are added at the start and ending of each sentence.
- Sentences whose length is less than or equal to 5 are removed.

Q1 (NNLM)

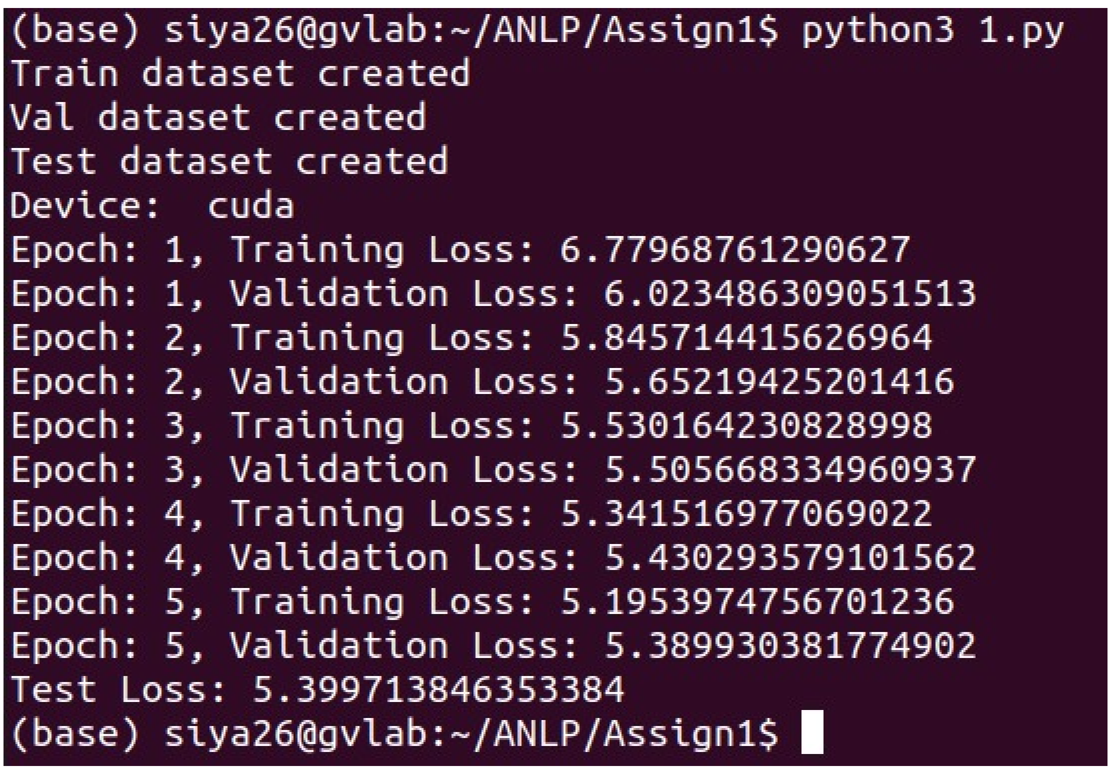
NNLM is used to predict the next word, given a 5-gram context in a sentence. 5-grams are processed batch-wise during training. While this allows it to capture short-term dependencies effectively, the model struggles with long-term dependencies that extend beyond the 5-word limit.

- Input Dimension: 100 $*$ 5 (5-grams, each is 100-dimensional)
- Hidden Dimension: 300
- Output Dimension: Size of vocabulary (`len(word2idx)`)
- Loss: Cross-Entropy

Hyperparameters

- No. of epochs: 5
- Batch size: 256
- Learning rate: 0.001
- Optimizer: Adam

Results



```
(base) siya26@gvlab:~/ANLP/Assign1$ python3 1.py
Train dataset created
Val dataset created
Test dataset created
Device:  cuda
Epoch: 1, Training Loss: 6.77968761290627
Epoch: 1, Validation Loss: 6.023486309051513
Epoch: 2, Training Loss: 5.845714415626964
Epoch: 2, Validation Loss: 5.65219425201416
Epoch: 3, Training Loss: 5.530164230828998
Epoch: 3, Validation Loss: 5.505668334960937
Epoch: 4, Training Loss: 5.341516977069022
Epoch: 4, Validation Loss: 5.430293579101562
Epoch: 5, Training Loss: 5.195397456701236
Epoch: 5, Validation Loss: 5.389930381774902
Test Loss: 5.399713846353384
(base) siya26@gvlab:~/ANLP/Assign1$
```

Average Perplexity

- Train: 187.75

- Val: 302.68
- Test: 296.72

Q2

LSTM (RNN-based language model) is used to predict the next word in a sentence. Sentences are padded to maximum length within their respective batch. It is designed to handle long-term dependencies using gates (input, forget, and output). Unlike NNLMs with a fixed context window, LSTMs can theoretically retain information over extended time periods, thus, giving LSTMs an advantage in modeling long-term dependencies.

- Input Dimension: 100
- Hidden Dimension: 300
- Output Dimension: Size of vocabulary (`len(word2idx)`)
- Loss: Cross-Entropy

Hyperparameters

- No. of layers: 1
- No. of epochs: 10
- Batch size: 64
- Learning rate: 0.001
- Optimizer: Adam

Results

```
(base) siya26@gvlab:~/ANLP/Assign1$ python3 2.py
Train dataset created
Val dataset created
Test dataset created
Device:  cuda
Epoch: 1, Training Loss: 1.873058889617865
Epoch: 1, Validation Loss: 1.6557632797956467
Epoch: 2, Training Loss: 1.5687857835457242
Epoch: 2, Validation Loss: 1.5354381865262985
Epoch: 3, Training Loss: 1.4763922722175205
Epoch: 3, Validation Loss: 1.4755360227823258
Epoch: 4, Training Loss: 1.4227475393777607
Epoch: 4, Validation Loss: 1.4379252678155898
Epoch: 5, Training Loss: 1.3830743140872868
Epoch: 5, Validation Loss: 1.4134117394685746
Epoch: 6, Training Loss: 1.3504960028261974
Epoch: 6, Validation Loss: 1.3968141055107117
Epoch: 7, Training Loss: 1.3226967219306134
Epoch: 7, Validation Loss: 1.3863496017456054
Epoch: 8, Training Loss: 1.2981586555639903
Epoch: 8, Validation Loss: 1.378506795167923
Epoch: 9, Training Loss: 1.276326623113676
Epoch: 9, Validation Loss: 1.3722308892011643
Epoch: 10, Training Loss: 1.2558116796372951
Epoch: 10, Validation Loss: 1.3671016162633896
Test Loss: 1.3078943884372711
(base) siya26@gvlab:~/ANLP/Assign1$
```

Average Perplexity

- Train: 143.53
- Val: 212.64
- Test: 211.98

Q3

Decoder is used to predict the next word in a sentence. Sentences are padded to maximum length within the whole corpus. The Decoder outperforms NNLM and LSTM due to its self-attention mechanism, which efficiently handles both short-term and long-term dependencies.

- Input Dimension: 100
- Hidden Dimension: 300
- Output Dimension: Size of vocabulary (`len(word2idx)`)
- Loss: Cross-Entropy

Hyperparameters

- No. of heads: 4
- No. of layers: 1
- Dropout: 0.1
- Feedforward dimension: 512
- No. of epochs: 5
- Batch size: 128
- Learning rate: 0.001
- Optimizer: Adam

Results

```
(base) siya26@gvlab:~/ANLP/Assign1$ python3 3.py
Train dataset created
Val dataset created
Test dataset created
Device:  cuda
Epoch: 1, Training Loss: 1.0741858884863469
Epoch: 1, Validation Loss: 0.8091203427314758
Epoch: 2, Training Loss: 0.5544420706814733
Epoch: 2, Validation Loss: 0.7610957515239716
Epoch: 3, Training Loss: 0.5158843201124805
Epoch: 3, Validation Loss: 0.7060149741172791
Epoch: 4, Training Loss: 0.4856249501650361
Epoch: 4, Validation Loss: 0.6696408116817474
Epoch: 5, Training Loss: 0.4601122897931899
Epoch: 5, Validation Loss: 0.6433716714382172
Test Loss: 0.5027886772155762
(base) siya26@gvlab:~/ANLP/Assign1$
```

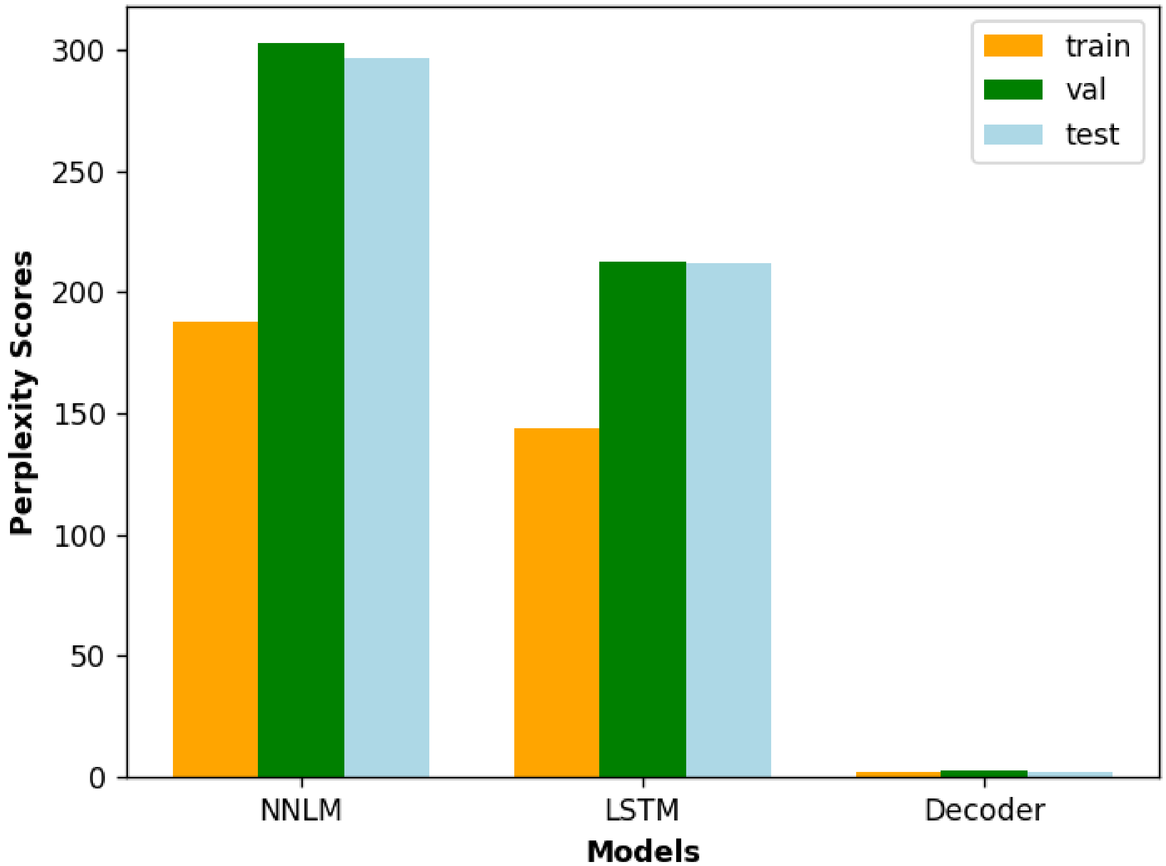
Average Perplexity

- Train: 1.74
- Val: 2.52
- Test: 1.98

Analysis

After observing the perplexity scores, we can conclude the performance of the models as follows:

Decoder >> LSTM > NNLM


train
val
test
Perplexity Scores
300
250
200
150
100
50
0
NNLM
LSTM
Decoder
Models