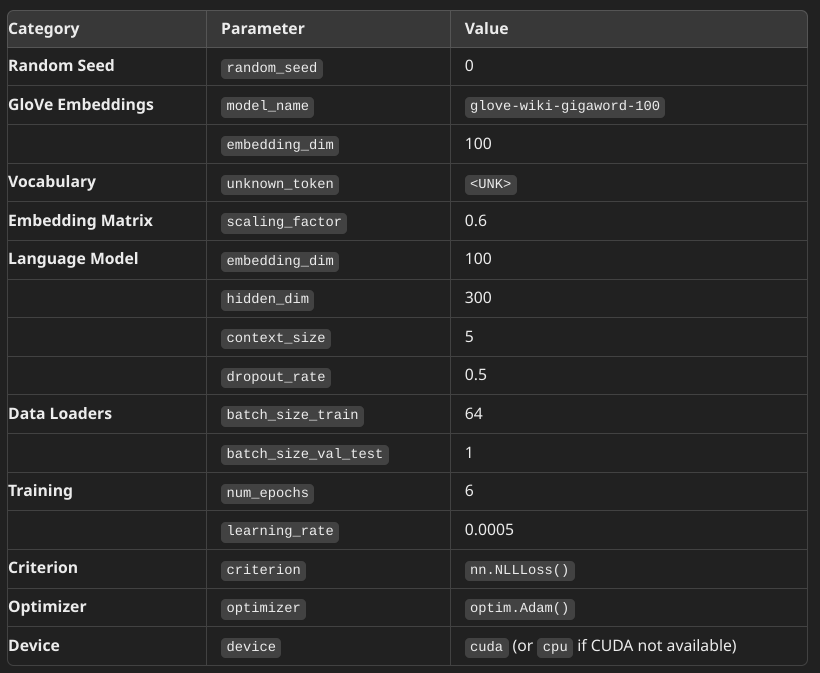
CS7.501 | Advanced NLP| Assignment - 1

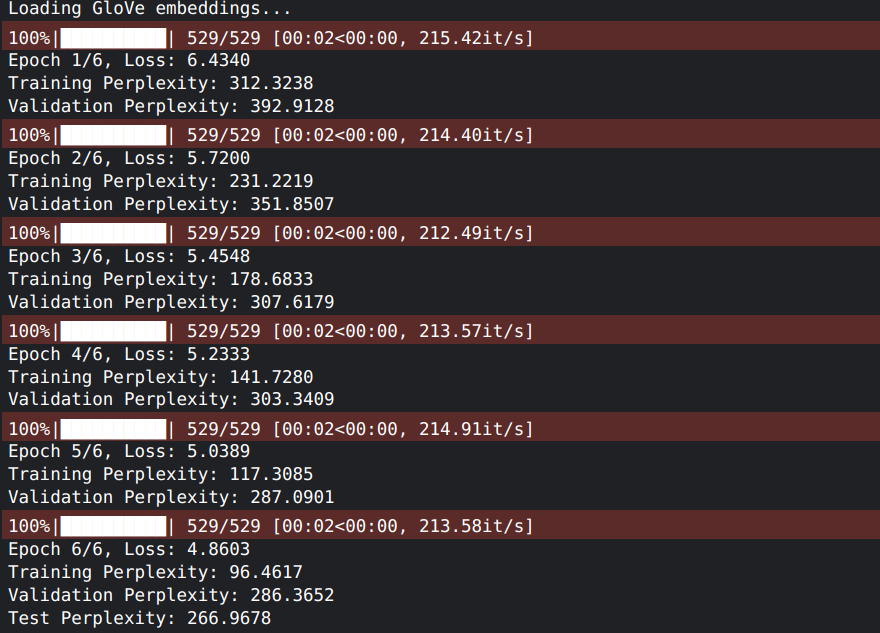
Shiva Shankar Gande

2023202005

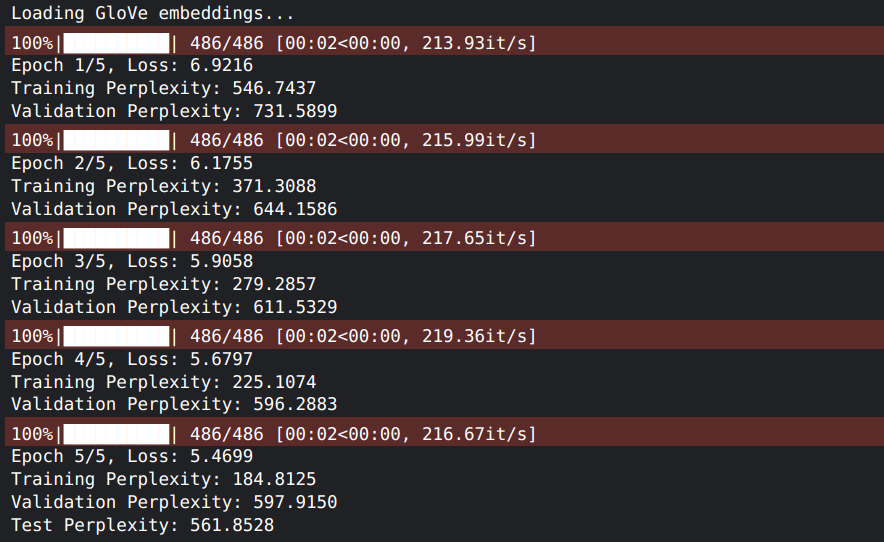
## **1 Neural Network Language Model**



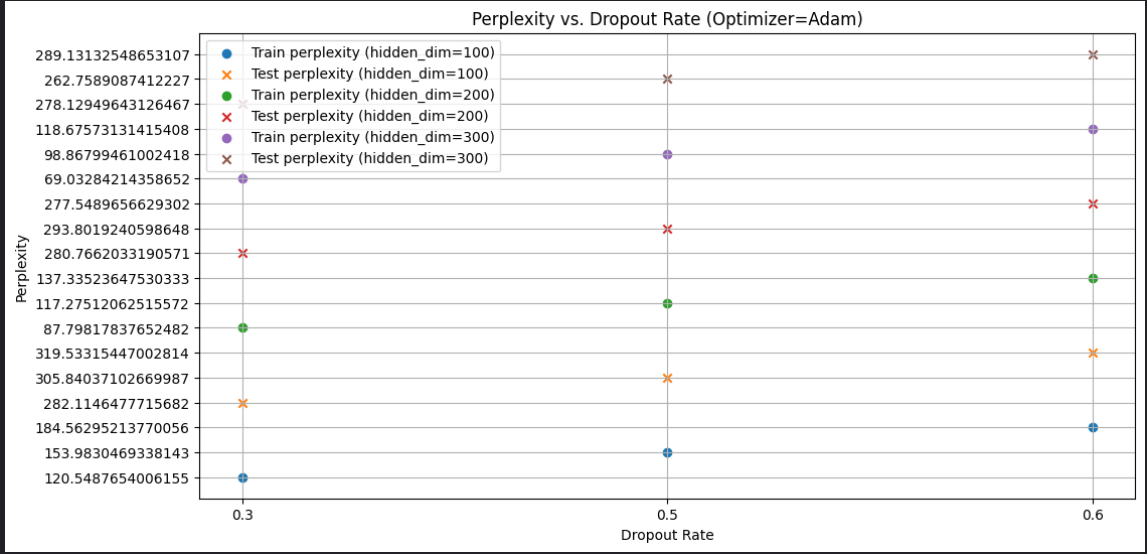
Without removing punctuations

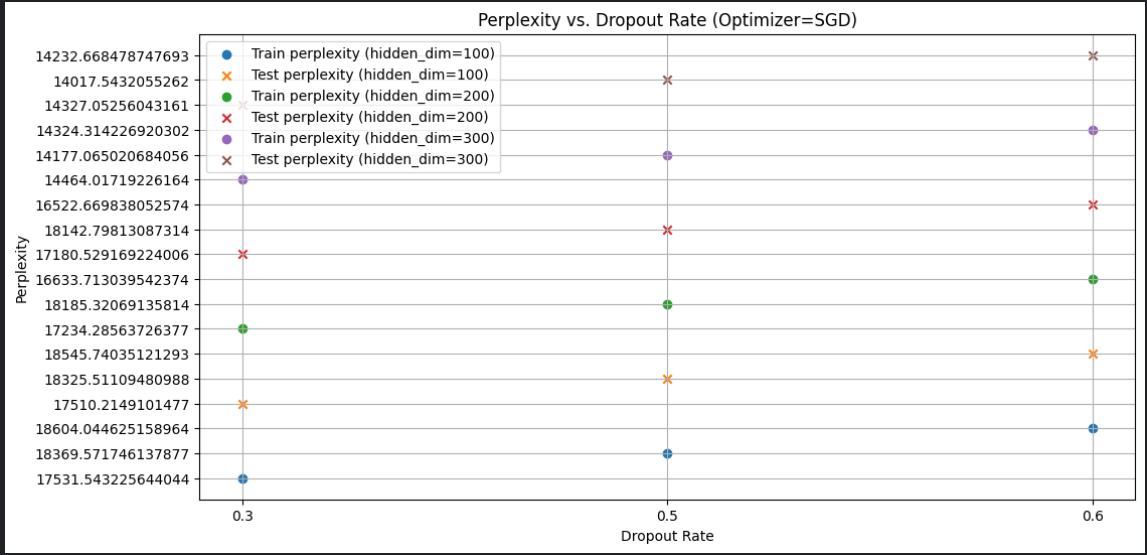


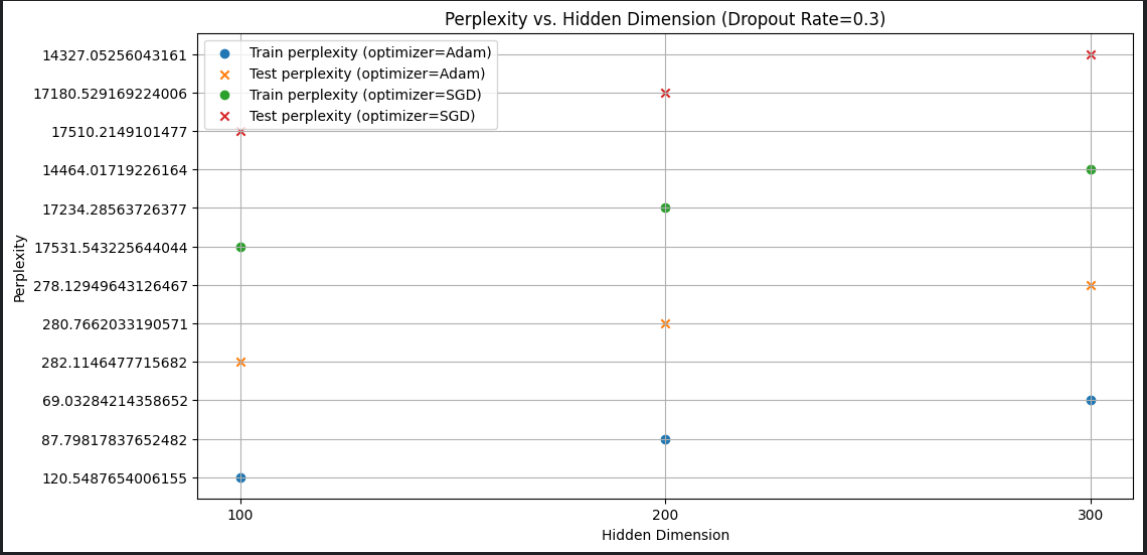
With removing punctuations

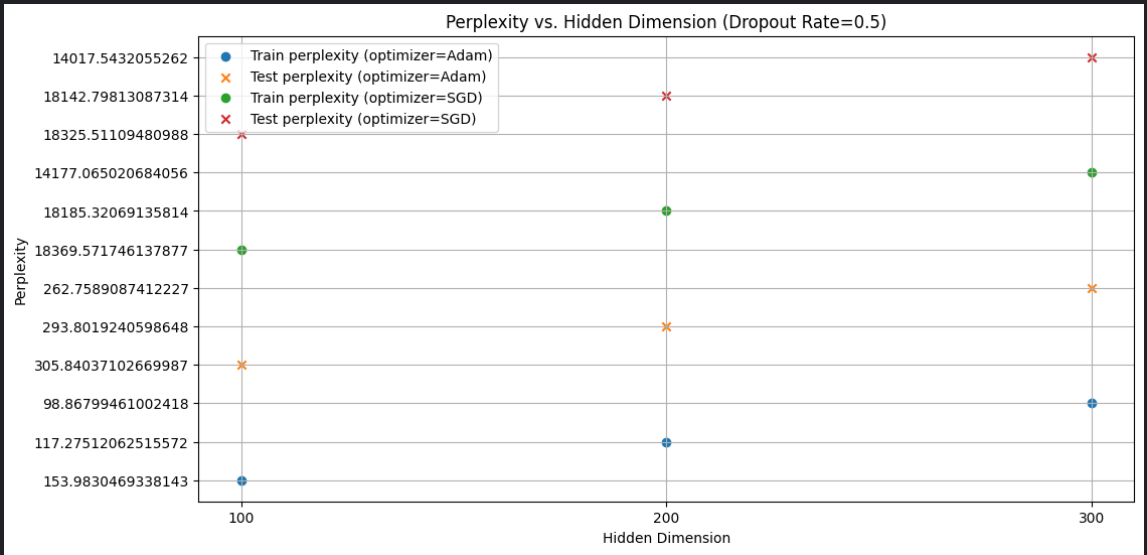


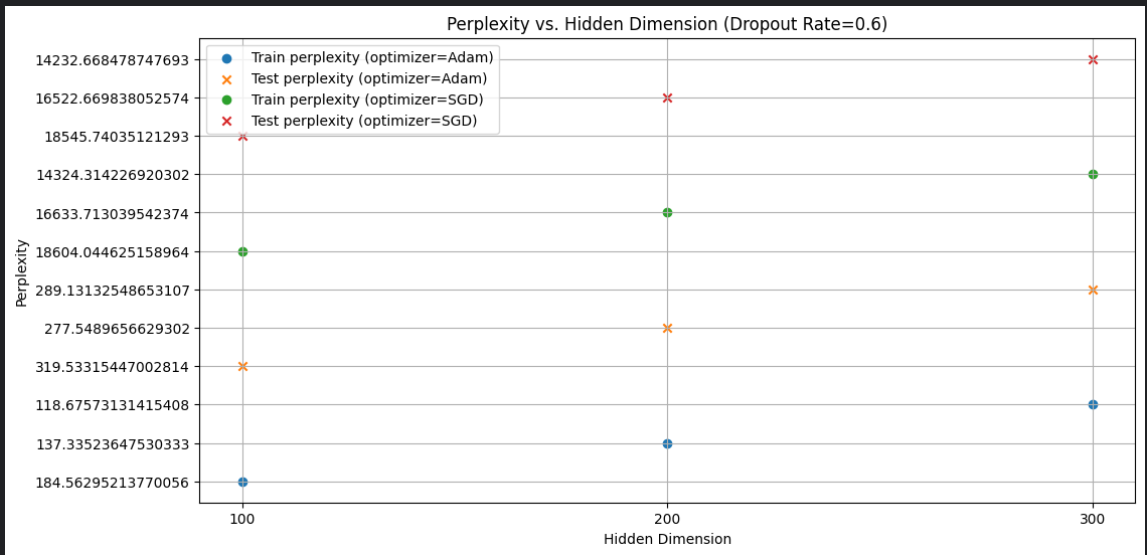
| Hyperparameter tuning   dropout\_rates = [0.3, 0.5, 0.6]  hidden\_dims = [100, 200, 300]  optimizers = {  'Adam': optim.Adam,  'SGD': optim.SGD  }  Loading GloVe embeddings... Testing with dropout\_rate=0.3, hidden\_dim=100, optimizer=Adam Epoch 1/6, Training Loss: 6.600046329807274 Epoch 2/6, Training Loss: 5.767457309958359 Epoch 3/6, Training Loss: 5.516558956194956 Epoch 4/6, Training Loss: 5.320902709830127 Epoch 5/6, Training Loss: 5.159758554277965 Epoch 6/6, Training Loss: 5.016563756141424 Perplexities - Train: 120.5487654006155, Val: 303.8639801481259, Test: 282.1146477715682 Testing with dropout\_rate=0.3, hidden\_dim=100, optimizer=SGD Epoch 1/6, Training Loss: 9.987481632708544 Epoch 2/6, Training Loss: 9.966470156689125 Epoch 3/6, Training Loss: 9.944000297028559 Epoch 4/6, Training Loss: 9.915775954025609 Epoch 5/6, Training Loss: 9.87676782383625 Epoch 6/6, Training Loss: 9.817169153142006 Perplexities - Train: 17531.543225644044, Val: 17624.202125025873, Test: 17510.2149101477 Testing with dropout\_rate=0.3, hidden\_dim=200, optimizer=Adam Epoch 1/6, Training Loss: 6.409498654063661 Epoch 2/6, Training Loss: 5.622801234286009 Epoch 3/6, Training Loss: 5.332954812644714 Epoch 4/6, Training Loss: 5.105601349691134 Epoch 5/6, Training Loss: 4.908686667423489 Epoch 6/6, Training Loss: 4.722426972181165 Perplexities - Train: 87.79817837652482, Val: 302.47431913479915, Test: 280.7662033190571 Testing with dropout\_rate=0.3, hidden\_dim=200, optimizer=SGD Epoch 1/6, Training Loss: 9.987233916678806 Epoch 2/6, Training Loss: 9.967827401122289 Epoch 3/6, Training Loss: 9.946007439328154 Epoch 4/6, Training Loss: 9.91559750912276 Epoch 5/6, Training Loss: 9.874117623241542 Epoch 6/6, Training Loss: 9.806262480559974 Perplexities - Train: 17234.28563726377, Val: 17326.84012949907, Test: 17180.529169224006 Testing with dropout\_rate=0.3, hidden\_dim=300, optimizer=Adam Epoch 1/6, Training Loss: 6.332263338716653 Epoch 2/6, Training Loss: 5.562985953166015 Epoch 3/6, Training Loss: 5.24669962221389 Epoch 4/6, Training Loss: 4.9800647707629375 Epoch 5/6, Training Loss: 4.740334660671162 Epoch 6/6, Training Loss: 4.528862446916572 Perplexities - Train: 69.03284214358652, Val: 304.26140909645994, Test: 278.12949643126467 Testing with dropout\_rate=0.3, hidden\_dim=300, optimizer=SGD Epoch 1/6, Training Loss: 9.97963221986285 Epoch 2/6, Training Loss: 9.953434303746489 Epoch 3/6, Training Loss: 9.921343379851223 Epoch 4/6, Training Loss: 9.875864750867873 Epoch 5/6, Training Loss: 9.807097346700646 Epoch 6/6, Training Loss: 9.681067280669549 Perplexities - Train: 14464.01719226164, Val: 14503.714944499987, Test: 14327.05256043161 Testing with dropout\_rate=0.5, hidden\_dim=100, optimizer=Adam Epoch 1/6, Training Loss: 6.75803931486498 Epoch 2/6, Training Loss: 5.925973081865026 Epoch 3/6, Training Loss: 5.72138099889179 Epoch 4/6, Training Loss: 5.565585460676374 Epoch 5/6, Training Loss: 5.426720477847663 Epoch 6/6, Training Loss: 5.309835049817646 Perplexities - Train: 153.9830469338143, Val: 332.4238658286466, Test: 305.84037102669987 Testing with dropout\_rate=0.5, hidden\_dim=100, optimizer=SGD Epoch 1/6, Training Loss: 9.987181039683655 Epoch 2/6, Training Loss: 9.969839935622995 Epoch 3/6, Training Loss: 9.950622709251062 Epoch 4/6, Training Loss: 9.927961090347706 Epoch 5/6, Training Loss: 9.898010010554096 Epoch 6/6, Training Loss: 9.855182620866426 Perplexities - Train: 18369.571746137877, Val: 18448.849636537896, Test: 18325.51109480988 Testing with dropout\_rate=0.5, hidden\_dim=200, optimizer=Adam Epoch 1/6, Training Loss: 6.548276049117889 Epoch 2/6, Training Loss: 5.811249007087122 Epoch 3/6, Training Loss: 5.567389812877701 Epoch 4/6, Training Loss: 5.354703682061275 Epoch 5/6, Training Loss: 5.175198895663139 Epoch 6/6, Training Loss: 5.02356567886883 Perplexities - Train: 117.27512062515572, Val: 315.95460293982615, Test: 293.8019240598648 Testing with dropout\_rate=0.5, hidden\_dim=200, optimizer=SGD Epoch 1/6, Training Loss: 10.000177319370941 Epoch 2/6, Training Loss: 9.979815910707428 Epoch 3/6, Training Loss: 9.959387696915602 Epoch 4/6, Training Loss: 9.933215921647134 Epoch 5/6, Training Loss: 9.899817611867547 Epoch 6/6, Training Loss: 9.848393470468258 Perplexities - Train: 18185.32069135814, Val: 18190.887855261488, Test: 18142.79813087314 Testing with dropout\_rate=0.5, hidden\_dim=300, optimizer=Adam Epoch 1/6, Training Loss: 6.434789221859351 Epoch 2/6, Training Loss: 5.71839049105753 Epoch 3/6, Training Loss: 5.463090479141349 Epoch 4/6, Training Loss: 5.243058755181565 Epoch 5/6, Training Loss: 5.06012604932096 Epoch 6/6, Training Loss: 4.873657890029786 Perplexities - Train: 98.86799461002418, Val: 285.7722867824029, Test: 262.7589087412227 Testing with dropout\_rate=0.5, hidden\_dim=300, optimizer=SGD Epoch 1/6, Training Loss: 9.986726173796919 Epoch 2/6, Training Loss: 9.957774633773186 Epoch 3/6, Training Loss: 9.924773318414868 Epoch 4/6, Training Loss: 9.877445147084586 Epoch 5/6, Training Loss: 9.80373771638612 Epoch 6/6, Training Loss: 9.673692627805647 Perplexities - Train: 14177.065020684056, Val: 14241.246892685176, Test: 14017.5432055262 Testing with dropout\_rate=0.6, hidden\_dim=100, optimizer=Adam Epoch 1/6, Training Loss: 6.926044169224449 Epoch 2/6, Training Loss: 6.032343579026504 Epoch 3/6, Training Loss: 5.84534536970413 Epoch 4/6, Training Loss: 5.707752443056263 Epoch 5/6, Training Loss: 5.591725616121287 Epoch 6/6, Training Loss: 5.481440338778893 Perplexities - Train: 184.56295213770056, Val: 333.6098351136401, Test: 319.53315447002814 Testing with dropout\_rate=0.6, hidden\_dim=100, optimizer=SGD Epoch 1/6, Training Loss: 9.989177842674152 Epoch 2/6, Training Loss: 9.973100799469432 Epoch 3/6, Training Loss: 9.956651562224668 Epoch 4/6, Training Loss: 9.93450682126871 Epoch 5/6, Training Loss: 9.906129370517196 Epoch 6/6, Training Loss: 9.867442151678924 Perplexities - Train: 18604.044625158964, Val: 18645.095970655766, Test: 18545.74035121293 Testing with dropout\_rate=0.6, hidden\_dim=200, optimizer=Adam Epoch 1/6, Training Loss: 6.621975268936473 Epoch 2/6, Training Loss: 5.890680602471626 Epoch 3/6, Training Loss: 5.681389439623308 Epoch 4/6, Training Loss: 5.489405424357786 Epoch 5/6, Training Loss: 5.332885224415758 Epoch 6/6, Training Loss: 5.188972839695125 Perplexities - Train: 137.33523647530333, Val: 299.6685674957917, Test: 277.5489656629302 Testing with dropout\_rate=0.6, hidden\_dim=200, optimizer=SGD Epoch 1/6, Training Loss: 9.975854239791396 Epoch 2/6, Training Loss: 9.954204902703054 Epoch 3/6, Training Loss: 9.929406823059818 Epoch 4/6, Training Loss: 9.898176892169953 Epoch 5/6, Training Loss: 9.853009580508662 Epoch 6/6, Training Loss: 9.781647579346028 Perplexities - Train: 16633.713039542374, Val: 16705.654900731955, Test: 16522.669838052574 Testing with dropout\_rate=0.6, hidden\_dim=300, optimizer=Adam Epoch 1/6, Training Loss: 6.5212276369541184 Epoch 2/6, Training Loss: 5.820869155086298 Epoch 3/6, Training Loss: 5.591036766799218 Epoch 4/6, Training Loss: 5.396946949500402 Epoch 5/6, Training Loss: 5.2156815056937855 Epoch 6/6, Training Loss: 5.054351044670972 Perplexities - Train: 118.67573131415408, Val: 314.3033004767873, Test: 289.13132548653107 Testing with dropout\_rate=0.6, hidden\_dim=300, optimizer=SGD Epoch 1/6, Training Loss: 9.976056820198592 Epoch 2/6, Training Loss: 9.948962996583099 Epoch 3/6, Training Loss: 9.917361500218952 Epoch 4/6, Training Loss: 9.872046566043352 Epoch 5/6, Training Loss: 9.804056232894892 Epoch 6/6, Training Loss: 9.680542918797629 Perplexities - Train: 14324.314226920302, Val: 14384.81291466275, Test: 14232.668478747693 |
| --- |











Best hyperparameters

## **2 RNN-based Language Model**



| class LanguageModelDataset(Dataset):  def \_\_init\_\_(self, sentences, word\_to\_idx, seq\_length=40):  self.sentences = sentences  self.word\_to\_idx = word\_to\_idx  self.seq\_length = seq\_length  self.data = self.prepare\_data()   def prepare\_data(self):  data = []  pad\_idx = self.word\_to\_idx["<PAD>"]  for sentence in self.sentences:  if len(sentence) > 1: # Ensure the sentence has at least 2 words  indexed\_sentence = [self.word\_to\_idx.get(word, self.word\_to\_idx["<UNK>"]) for word in sentence]    # Pad or truncate the sentence to seq\_length + 1  if len(indexed\_sentence) < self.seq\_length + 1:  indexed\_sentence = indexed\_sentence + [pad\_idx] \* (self.seq\_length + 1 - len(indexed\_sentence))  else:  indexed\_sentence = indexed\_sentence[:self.seq\_length + 1]    # Create input-target pairs  seq = indexed\_sentence[:self.seq\_length]  target = indexed\_sentence[1:self.seq\_length + 1]    data.append((seq, target))  return data   def \_\_len\_\_(self):  return len(self.data)   def \_\_getitem\_\_(self, idx):  seq, target = self.data[idx]  return torch.tensor(seq), torch.tensor(target) |
| --- |

**Class: LanguageModelDataset**

* **Purpose**: Prepares and manages a dataset of text sequences for training a language model.

**Constructor: \_\_init\_\_**

* **Purpose**: Initializes the dataset with sentences, vocabulary mappings, and sequence length. Calls prepare\_data to process the sentences.

**Method: prepare\_data**

* **Purpose**: Converts sentences into sequences of indices. Pads or truncates sequences to a fixed length. Creates input-target pairs for model training.

**Method: \_\_len\_\_**

* **Purpose**: Returns the number of data pairs in the dataset.

**Method: \_\_getitem\_\_**

* **Purpose**: Retrieves the input and target tensors for a given index.



| **Loading GloVe embeddings... 100%|██████████| 529/529 [00:27<00:00, 18.90it/s] Epoch 1/10, Loss: 6.1560 Epoch 1 Training Perplexity: 265.1210 Epoch 1 Validation Perplexity: 264.4561 100%|██████████| 529/529 [00:25<00:00, 21.10it/s] Epoch 2/10, Loss: 5.4908 Epoch 2 Training Perplexity: 184.1699 Epoch 2 Validation Perplexity: 188.8351 100%|██████████| 529/529 [00:26<00:00, 20.32it/s] Epoch 3/10, Loss: 5.2389 Epoch 3 Training Perplexity: 149.0239 Epoch 3 Validation Perplexity: 157.2684 100%|██████████| 529/529 [00:25<00:00, 20.56it/s] Epoch 4/10, Loss: 5.0770 Epoch 4 Training Perplexity: 127.6362 Epoch 4 Validation Perplexity: 140.6669 100%|██████████| 529/529 [00:25<00:00, 20.57it/s] Epoch 5/10, Loss: 4.9525 Epoch 5 Training Perplexity: 111.8711 Epoch 5 Validation Perplexity: 129.1297 100%|██████████| 529/529 [00:25<00:00, 20.52it/s] Epoch 6/10, Loss: 4.8479 Epoch 6 Training Perplexity: 100.2902 Epoch 6 Validation Perplexity: 121.4492 100%|██████████| 529/529 [00:25<00:00, 20.47it/s] Epoch 7/10, Loss: 4.7583 Epoch 7 Training Perplexity: 90.9603 Epoch 7 Validation Perplexity: 116.0041 100%|██████████| 529/529 [00:25<00:00, 20.47it/s] Epoch 8/10, Loss: 4.6794 Epoch 8 Training Perplexity: 83.3873 Epoch 8 Validation Perplexity: 112.2057 100%|██████████| 529/529 [00:25<00:00, 20.45it/s] Epoch 9/10, Loss: 4.6077 Epoch 9 Training Perplexity: 76.8515 Epoch 9 Validation Perplexity: 109.0440 100%|██████████| 529/529 [00:25<00:00, 20.44it/s] Epoch 10/10, Loss: 4.5425 Epoch 10 Training Perplexity: 71.4666 Epoch 10 Validation Perplexity: 107.8051**  Final Test Perplexity: 105.9117 |
| --- |

## 3 Transformer Decoder based Language Model:

**For the 1st version:**

| class LanguageModelDataset(Dataset):  def \_\_init\_\_(self, sentences, word\_to\_idx, context\_size=5, pad\_idx=0):  self.sentences = [s for s in sentences if len(s) >= context\_size + 1]  self.context\_size = context\_size  self.pad\_idx = pad\_idx   def \_\_len\_\_(self):  return len(self.sentences)   def \_\_getitem\_\_(self, idx):  sentence = self.sentences[idx]  context = sentence[:self.context\_size]  target = sentence[1:self.context\_size+1]  return torch.tensor(context, dtype=torch.long), torch.tensor(target, dtype=torch.long) |
| --- |

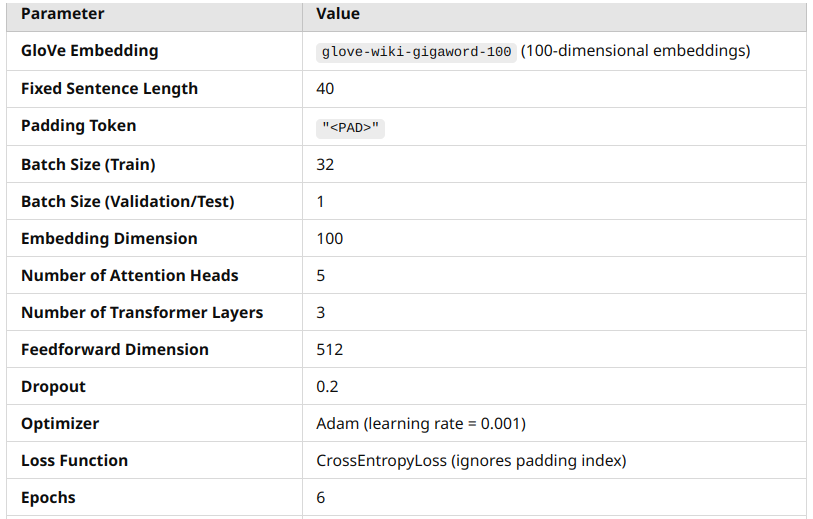
| Epoch 1/4: 100%|██████████| 1058/1058 [00:19<00:00, 54.20it/s] Epoch 1/4, Train Loss: 3.6795, Train Perplexity: 8.5655, Val Perplexity: 15.8006 Epoch 2/4: 100%|██████████| 1058/1058 [00:19<00:00, 55.51it/s] Epoch 2/4, Train Loss: 2.2453, Train Perplexity: 4.2638, Val Perplexity: 11.6119 Epoch 3/4: 100%|██████████| 1058/1058 [00:19<00:00, 54.36it/s] Epoch 3/4, Train Loss: 1.7493, Train Perplexity: 3.1631, Val Perplexity: 11.3980 Epoch 4/4: 100%|██████████| 1058/1058 [00:18<00:00, 56.11it/s] Epoch 4/4, Train Loss: 1.4849, Train Perplexity: 2.8677, Val Perplexity: 12.3838 Test Perplexity: 12.2444 |
| --- |

**For the 2nd version:**

| class LanguageModelDataset(Dataset):  def \_\_init\_\_(self, sentences, word\_to\_idx, pad\_idx=0):  self.sentences = [s for s in sentences if len(s) > 1] # Ensure sentences have more than 1 token  self.pad\_idx = pad\_idx   def \_\_len\_\_(self):  return len(self.sentences)   def \_\_getitem\_\_(self, idx):  sentence = self.sentences[idx]  context = sentence[:-1] # All tokens except the last one  target = sentence[1:] # All tokens except the first one  return torch.tensor(context, dtype=torch.long), torch.tensor(target, dtype=torch.long) |
| --- |

| Epoch 1: Train Loss = 2.5270, Train Perplexity = 1.7918, Val Perplexity = 2.3060  Epoch 2/4: 100%|██████████| 1058/1058 [07:05**<**00**:**00**,** 2.49**it/s] Epoch** 2**: Train Loss =** 0.5398**, Train Perplexity =** 1.2171**, Val Perplexity =** 1.7159 **Epoch** 3**/**4**:** 100**%|██████████|** 1058**/**1058 **[**07**:**05**<**00**:**00**,** 2.49**it/s] Epoch** 3**: Train Loss =** 0.2488**, Train Perplexity =** 1.1259**, Val Perplexity =** 1.6482 **Epoch** 4**/**4**:** 100**%|██████████|** 1058**/**1058 **[**07**:**05**<**00**:**00**,** 2.49**it/s] Epoch** 4**: Train Loss =** 0.1617**, Train Perplexity =** 1.0958**, Val Perplexity =** 1.6486  **Test Perplexity: 1.6095** |
| --- |

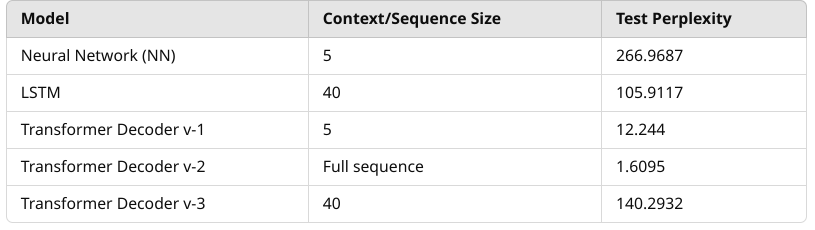
**For the 3rd version:**

****

| class LanguageModelDataset(Dataset):  def \_\_init\_\_(self, sentences, word\_to\_idx, fixed\_length=40, pad\_idx=0):  self.sentences = [s for s in sentences if len(s) > 0] # Filter out empty sentences  self.word\_to\_idx = word\_to\_idx  self.fixed\_length = fixed\_length  self.pad\_idx = pad\_idx   def \_\_len\_\_(self):  return len(self.sentences)   def \_\_getitem\_\_(self, idx):  sentence = self.sentences[idx]  # Convert words to indices  indices = [self.word\_to\_idx.get(word, self.word\_to\_idx["<UNK>"]) for word in sentence]   # Pad or truncate to fixed length  if len(indices) < self.fixed\_length:  indices += [self.pad\_idx] \* (self.fixed\_length - len(indices)) # Padding  else:  indices = indices[:self.fixed\_length] # Truncation   # Prepare context and target  context = indices[:-1] # All but the last word  target = indices[1:] # All but the first word  return torch.tensor(context, dtype=torch.long), torch.tensor(target, dtype=torch.long) |
| --- |

| Epoch 1/6: 100%|██████████| 1058/1058 [00:20<00:00, 51.55it/s] Epoch 1/6, Train Loss: 6.5932, Train Perplexity: 164.8817, Val Perplexity: 190.0352 Epoch 2/6: 100%|██████████| 1058/1058 [00:19<00:00, 53.33it/s] Epoch 2/6, Train Loss: 5.2095, Train Perplexity: 120.3677, Val Perplexity: 164.1827 Epoch 3/6: 100%|██████████| 1058/1058 [00:20<00:00, 52.41it/s] Epoch 3/6, Train Loss: 4.9907, Train Perplexity: 101.9890, Val Perplexity: 155.4045 Epoch 4/6: 100%|██████████| 1058/1058 [00:19<00:00, 53.14it/s] Epoch 4/6, Train Loss: 4.8598, Train Perplexity: 91.9084, Val Perplexity: 149.6447 Epoch 5/6: 100%|██████████| 1058/1058 [00:20<00:00, 52.83it/s] Epoch 5/6, Train Loss: 4.7642, Train Perplexity: 83.6704, Val Perplexity: 145.8721 Epoch 6/6: 100%|██████████| 1058/1058 [00:19<00:00, 52.93it/s] Epoch 6/6, Train Loss: 4.6898, Train Perplexity: 76.8503, Val Perplexity: 144.0227 Test Perplexity: 140.2932 |
| --- |

## Analysis



### **Analysis:**

1. **Neural Network (NN)** with a small context size of 5 yields the **highest perplexity (266.9687)**, indicating it struggles to capture long-range dependencies and perform well in language modeling tasks, especially with limited context.
2. **LSTM** shows a **significant improvement (105.9117 perplexity)** when using a sequence length of 40, reflecting its ability to capture longer-term dependencies better than the NN. LSTMs are known for handling sequential data well by maintaining a memory of previous words.
3. **Transformer Decoder v-1**, using a small context size of 5, performs **far better than the NN and LSTM** with a perplexity of **12.244**. This suggests that even with limited context, transformers outperform traditional architectures in handling dependencies.
4. **Transformer Decoder v-2**, which uses the full sequence as context, achieves a perplexity of **1.6095**—the best result among all models. This highlights the transformer’s ability to leverage the entire sequence effectively for predicting the next token, making it highly efficient.
5. **Transformer Decoder v-3**, with a sequence length of 40, shows a **higher perplexity (140.2932)** compared to the v-1 and v-2 versions. This could be due to limitations in capturing sequence information optimally with the specific setup and paarmeters.