Fall-2024
Assignment 3

Course Title: CS-4045 Deep Learning for Perception

Section: D

Name: 21I-2502 Mansoor Ali

Department of Computer Science

National University of Computing & Emerging Sciences

Islamabad, Pakistan

Fill-in-the-Blank Prediction using LSTM Networks

1. Data Preprocessing Details

For this project, I used the RACE dataset to train an LSTM-based model to predict missing words in a sentence. This required careful preprocessing of the data, as handling natural language and setting up the model to learn effectively were both critical.

The preprocessing involved several steps:

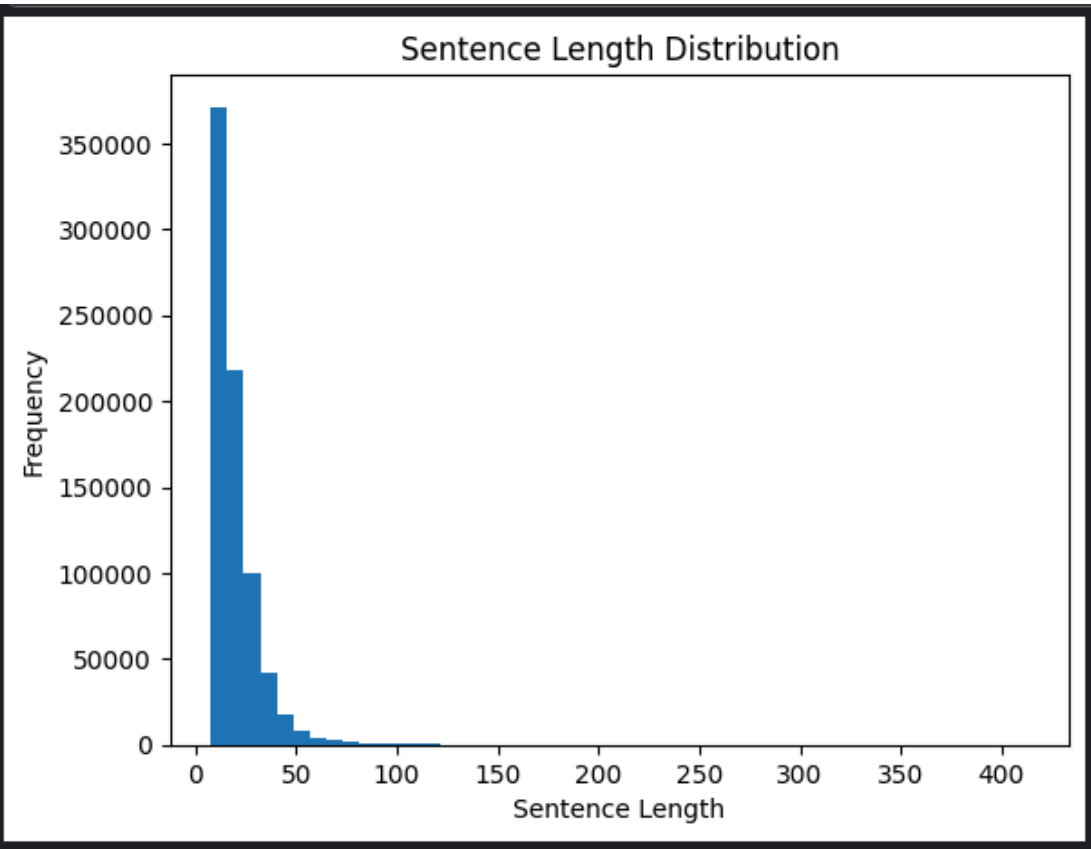
1. **Sentence Tokenization**: First, I split the text into individual sentences. This was essential for creating training samples, as each sentence would later have one word removed and replaced with a blank for prediction purposes.
2. **Word Tokenization and Sentence Length Filtering**: After splitting each sentence, I tokenized it into words and removed sentences shorter than eight words to ensure enough context for meaningful predictions.
3. **Creating Blanks**: I designed the model to predict one word removed from each sentence. To balance context on both sides, I selected the missing word from the first half of each sentence. This strategy ensured that both "Part A" (words before the blank) and "Part B" (words after the blank) contained a reasonable number of tokens, facilitating the model's learning process.
4. **Removing Stop Words and Punctuation**: To avoid uninformative blanks, I skipped over common stop words and punctuation when selecting the missing word. Only relevant, meaningful words were chosen to be blanked out.

5. **Input Preparation**: I split each sentence into two parts for the forward and backward LSTM models:
    - **Part A**: The words before the missing word.
    - **Part B (Reversed)**: The words following the missing word, but reversed to suit the backward LSTM. These two parts formed the inputs for the models, and padding was applied to ensure uniform input length.
6. **Vocabulary Creation**: I created a vocabulary with a size limit of 5000, including special tokens (<PAD> and <UNK>), to standardize the inputs and manage memory effectively.

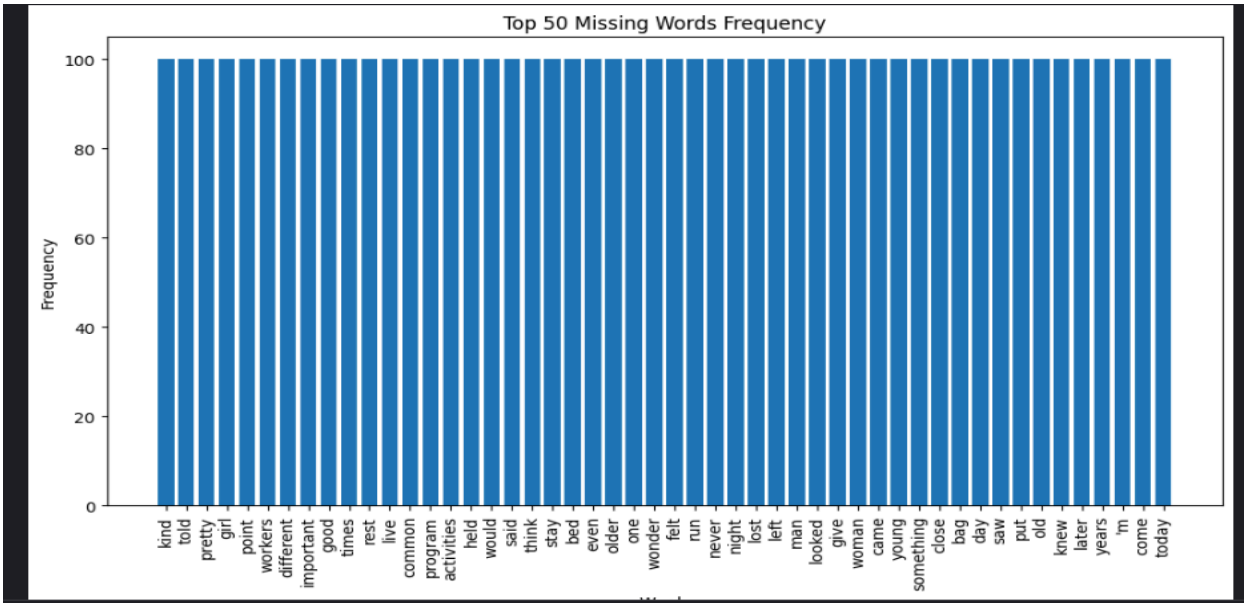
Throughout this process, I aimed to balance the model's capacity to learn from each sample with constraints on data quality and vocabulary size. This attention to preprocessing was essential for optimizing the model's training and performance.

Outputs:

```
--------------------------------------------------
Original Sentence: It seems to me that there is a trend towards wanting "perfection" , and that is an ideal that just does not exist in
reality.
Sentence without Blank: It to me that there is a trend towards wanting `` perfection '' , and that is an ideal that just does not exist
in reality .
Missing Word: seems
Part A: It
Part B Reversed: . reality in exist not does just that ideal an is that and , '' perfection `` wanting towards trend a is there that me
to
--------------------------------------------------
Original Sentence: I asked, expecting them to tell me that they would need a   or family friend to help them out.
Sentence without Blank: I asked , them to tell me that they would need a or family friend to help them out .
Missing Word: expecting
Part A: I asked ,
Part B Reversed: . out them help to friend family or a need would they that me tell to them
--------------------------------------------------
Original Sentence: I was pretty alarmed by that response.
Sentence without Blank: I was alarmed by that response .
Missing Word: pretty
Part A: I was
Part B Reversed: . response that by alarmed
--------------------------------------------------
Original Sentence: "They break your legs, put in special extending screws, and slowly expand the gap between the two ends of the bone as
it re-grows, you can get at least 5 cm taller!"
Sentence without Blank: `` They break your legs , put in special extending screws , and slowly the gap between the two ends of the bone
as it re-grows , you can get at least 5 cm taller ! ''
Missing Word: expand
Part A: `` They break your legs , put in special extending screws , and slowly
Part B Reversed: '' ! taller cm 5 least at get can you , re-grows it as bone the of ends two the between gap the
--------------------------------------------------
```




```
Original data size: 769132
Balanced data size: 497775
Final data size: 276793
```



2. Model Architecture

To tackle this task, I experimented with a forward LSTM model, a backward LSTM model, and ultimately a combined model that took advantage of bidirectional learning.

1. **Embedding Layer**: Each model began with an embedding layer, initialized using pre-trained GloVe embeddings. This layer converted words into dense vector representations, which helped the model understand relationships between words in a richer context. The embedding dimension was set to 100, matching the GloVe vectors.
2. **LSTM Layers**: I used bidirectional LSTM layers for both the forward and backward parts:
    - **Forward LSTM**: This model processed Part A of the sentence as a regular LSTM sequence.
    - **Backward LSTM**: This model processed the reversed Part B as input, focusing on learning the sequence from the opposite direction.
3. **Attention Layers**: After the LSTM layers, I applied attention mechanisms to focus the model on the most relevant parts of each sequence. Attention layers helped highlight significant words, ensuring the model could make better predictions based on context.
4. **Concatenation and Dense Layers**: I combined the attention-processed outputs of both LSTM parts using a concatenation layer. This output was then fed into dense layers with dropout regularization and L2 regularization to prevent overfitting. The final output layer used softmax activation to predict the missing word from the vocabulary.

5. **Hyperparameters**:
    - LSTM units: 256
    - Dropout: 0.3
    - L2 regularization strength: 0.01
    - Batch size: 64
    - Learning rate: 1e-3

The model architecture was carefully tuned to balance complexity and efficiency, with regularization techniques added to enhance generalization.

3. Training Details

Training the model required a structured approach to achieve a balance between underfitting and overfitting. Below is a summary of the key details and challenges I encountered:

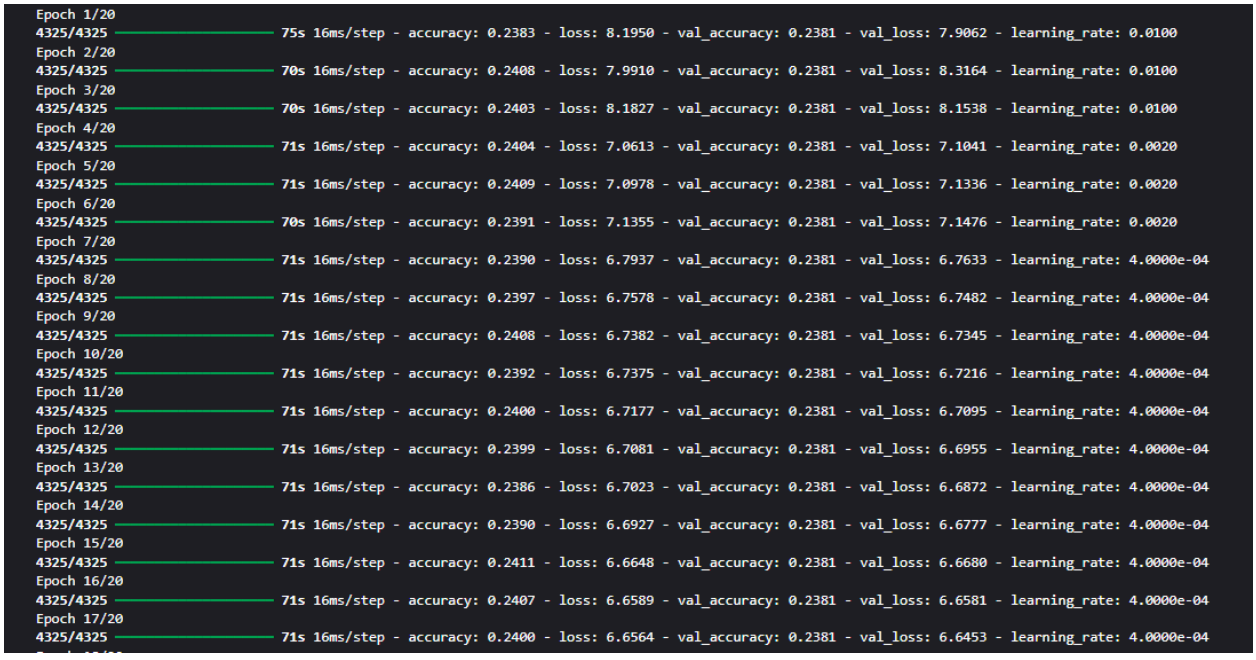
1. **Training and Validation Performance**: The model's accuracy improved slightly over time, but the plateau in validation accuracy indicated the model struggled to generalize beyond a certain point. I observed that despite extensive tuning, both the training and validation accuracy hovered around 24%.
2. **Hyperparameter Tuning**: I experimented with various learning rates, batch sizes, and regularization techniques. Initially, the learning rate was too high, which led to fluctuating results. Lowering it to 1e-5 helped stabilize training. Additionally, I incorporated L2 regularization to prevent overfitting.
3. **Challenges with Predictions**: Throughout training, I noted that the model had a tendency to predict generic words like "example," "anything," or <UNK> when it was uncertain. To address this, I filtered

out <PAD> and <UNK> tokens during prediction. However, improving the model's nuanced understanding of context remained challenging.

4. **Early Stopping and Learning Rate Reduction**: To prevent overfitting, I used early stopping based on validation loss, coupled with a learning rate reduction when the validation loss plateaued. These techniques were helpful in stabilizing training.

The training process was iterative, with each round of adjustments revealing new insights. The model's final validation accuracy of ~24% indicates that while it captured some patterns, more improvements could be made.

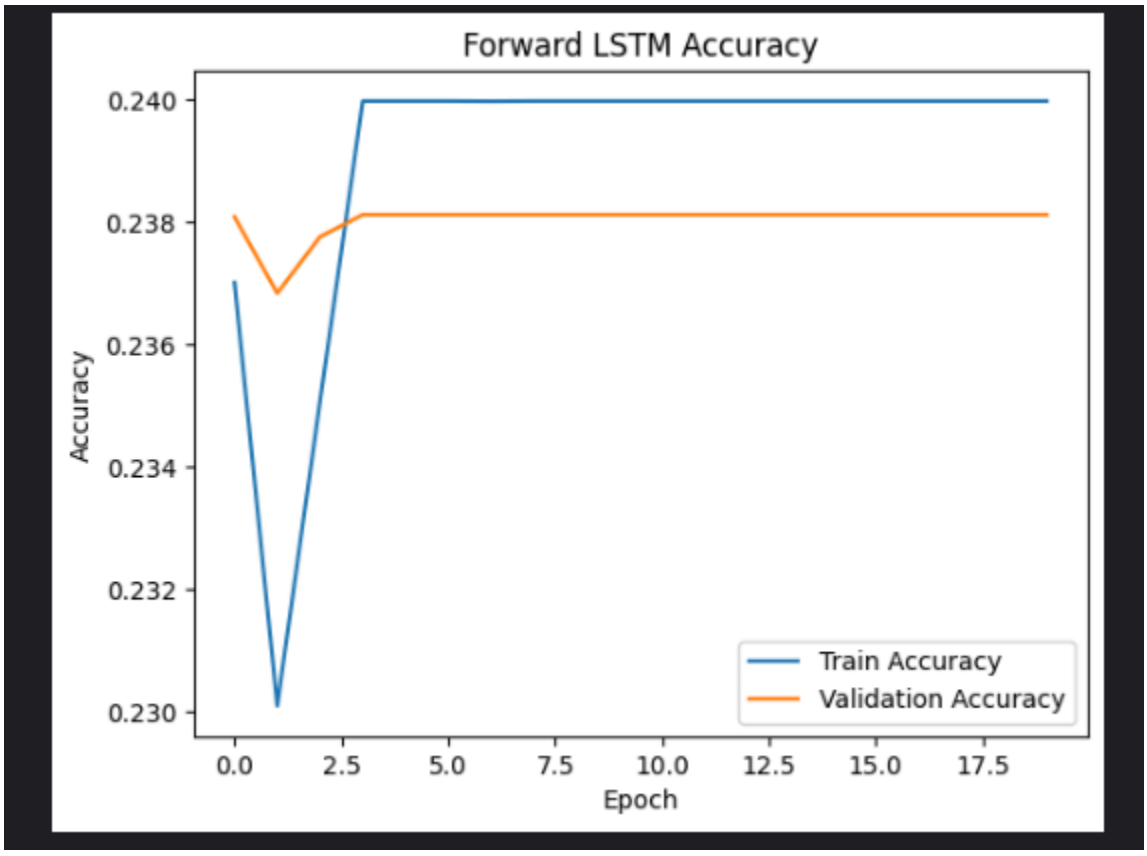
Outputs:



4. Observations

Reflecting on the training and evaluation of this model, I gained insights into both the strengths and limitations of using LSTMs for missing word prediction:

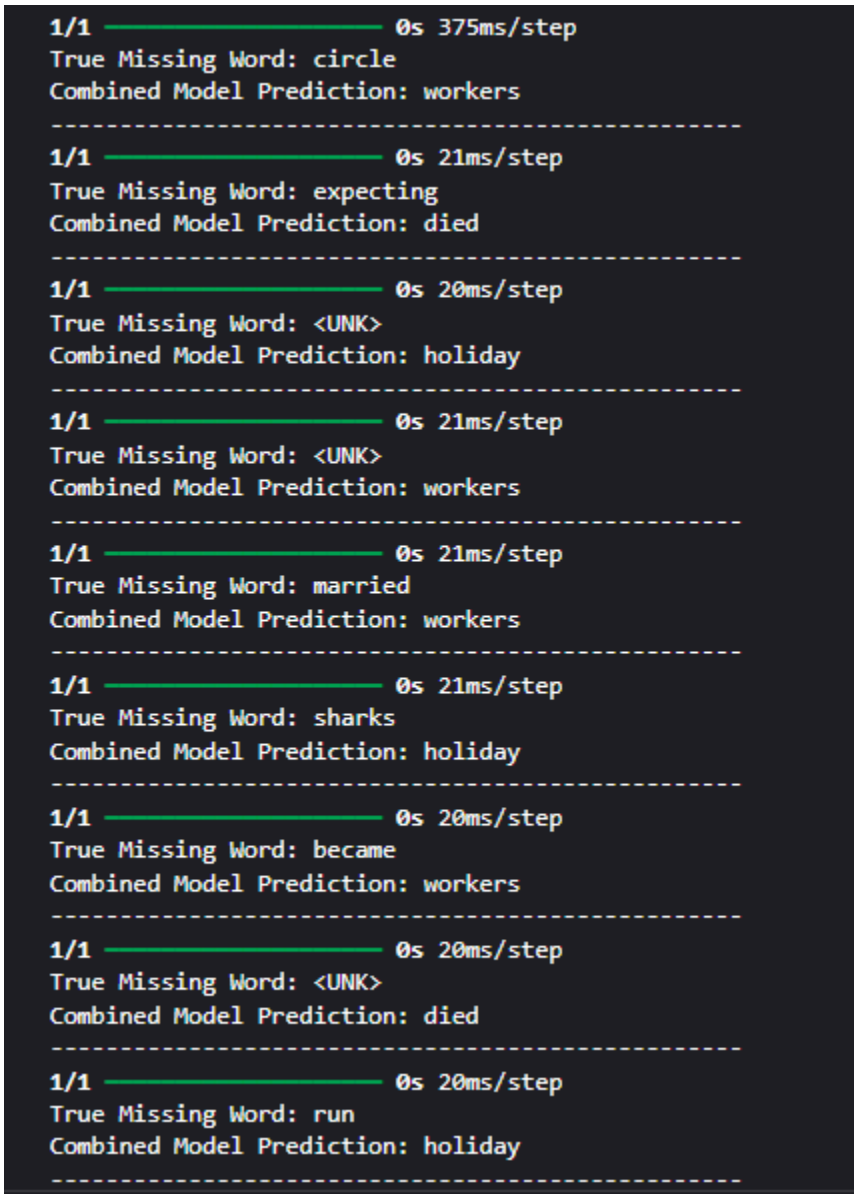
1. **Accuracy**: The combined model's accuracy plateaued early, suggesting that the architecture might need further tuning or that additional context from each sentence is required for accurate predictions. The final validation accuracy of approximately 24% suggests that while the model learned some patterns, it could not consistently predict missing words accurately.



2. **Differences in Forward and Backward Predictions**: The combined model brought together the insights from both forward and backward

LSTM components, yet I observed that certain generic predictions, like "example" for missing words like "versions," remained common. This suggests that the forward and backward predictions complemented each other, but they were limited in handling subtle contextual differences.

3. **Predictions:**



4. **Challenges Faced**:

- **Data Imbalance and Contextual Complexity**: Creating balanced data with meaningful missing words was a challenge. Some blanks were chosen from the last part of sentences, which led to unbalanced parts A and B, impacting prediction accuracy.
- **Vocabulary Constraints**: The vocabulary size constraint meant that many words were mapped to <UNK>, making it hard for the model to differentiate subtle contextual cues.
- **Limited Improvement with L2 Regularization**: While L2 regularization helped avoid overfitting to an extent, it didn't significantly boost performance. The model's accuracy still remained relatively low, suggesting that alternative regularization methods or a different architecture might be needed.

5. **Recommendations for Future Work**:
   - **Explore Transformer Models**: Given the context-dependent nature of this task, transformer models like BERT could capture the nuanced relationships between words better than LSTMs.
   - **Enhanced Data Augmentation**: Further exploration into balanced data preprocessing, possibly with different blanking strategies, could improve learning.
   - **Fine-Tune Pre-Trained Models**: Leveraging pre-trained models on larger vocabularies and fine-tuning them could help improve performance.

Overall, this assignment provided valuable insights into the complexities of word prediction using LSTMs. Although the final results were modest, the process illuminated key aspects of data preprocessing, model architecture, and training challenges. The combined model, while limited in accuracy, laid a foundation for further improvements in future iterations.