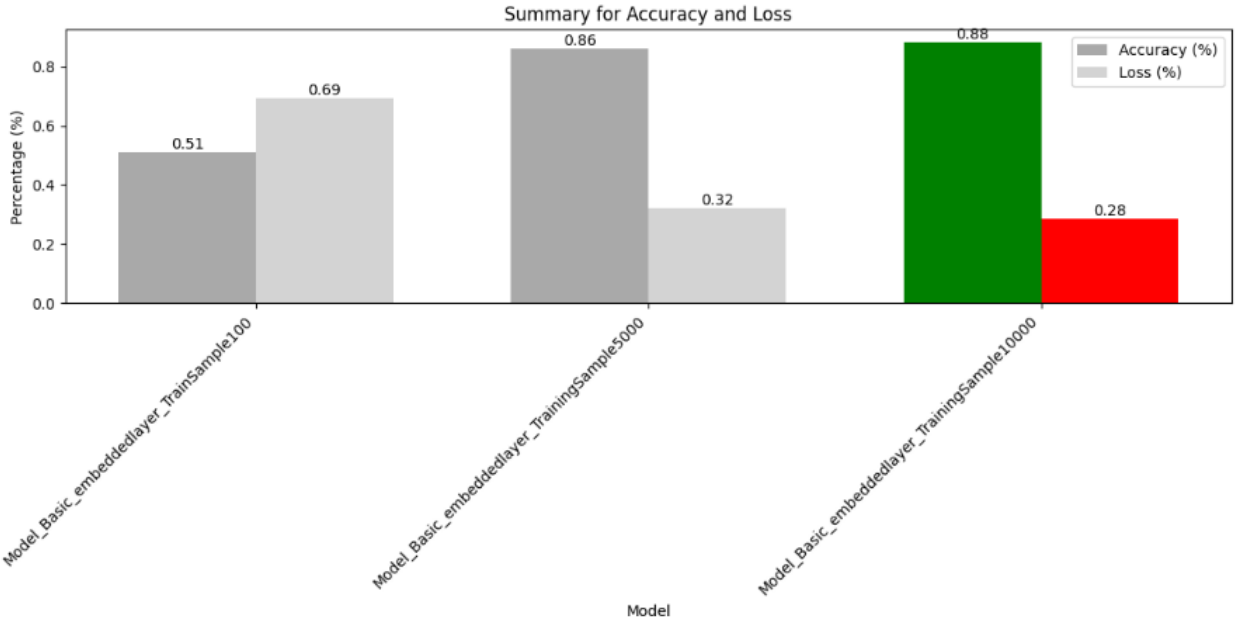
Assignment4: Text and Sequence Data using IMDB dataset

Name: Kandarp Barot

Date: 27 July 2024

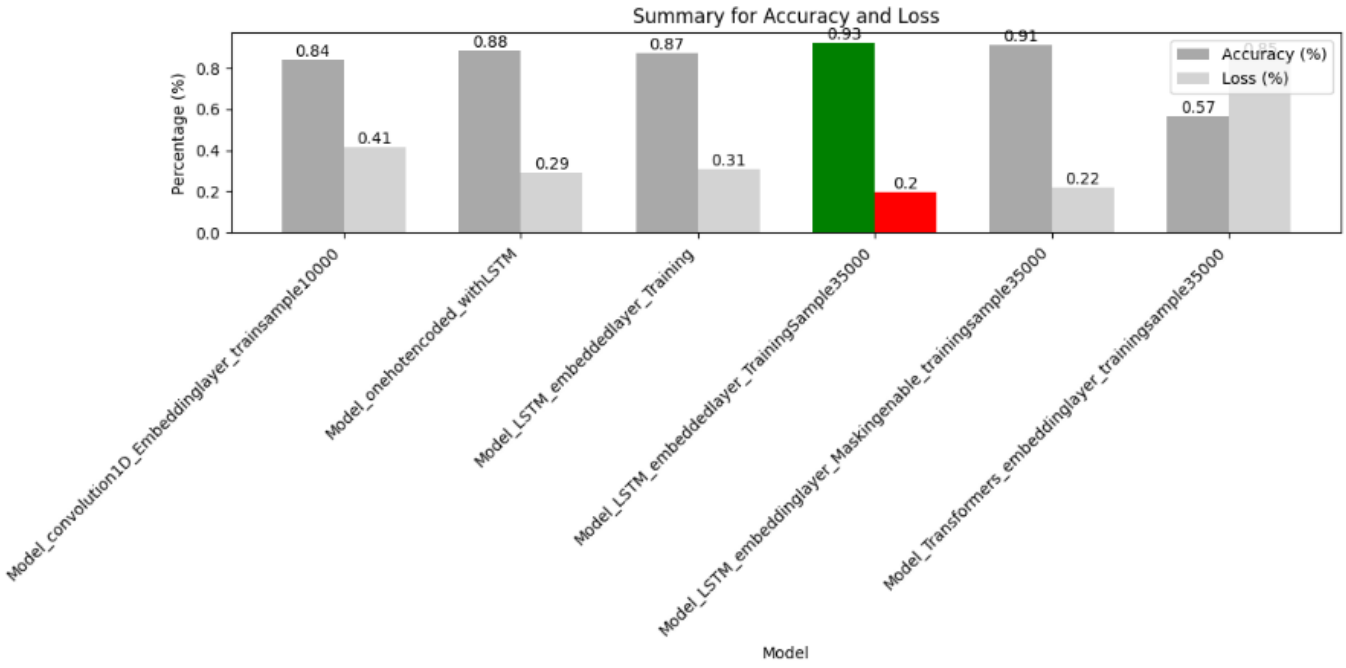
In this assignment, we will construct an embedding layer and utilize pre-trained word embedding models to train on the IMDB reviews dataset. Our goal is to identify distinctive features within the reviews. The ultimate objective is to enable the model to make binary predictions, determining whether a given review is negative or positive. Additionally, we will modify the model by varying the training sample size to determine the optimal point at which the embedding layers yield better performance.

We have created 3 different models with **100, 5,000 and 10,000 train samples, 10,000 validations, considered only top 10, 000 words and cutoff revies after 150 words.** Below is the comparison graph for Accuracy and Loss for those 3 models.



As we experimented with three basic sequence models using an embedding layer and varied the training sample sizes**. A notable trend emerged: increasing the training sample size correlated with higher model accuracy and reduced loss**. Specifically, the model trained on 100 samples exhibited 51% accuracy and 69% loss. The model trained on 5,000 samples showed significant improvement with 86% accuracy and 32% loss, while the model trained on 10,000 samples demonstrated a slight improvement with 88% accuracy and 28% loss. This trend suggests that as the training sample size increases, models become more adept at generalizing from the data, resulting in improved predictive performance and lower loss metrics. Larger datasets contribute to better learning of patterns and relationships within the data, leading to more accurate predictions.

We Modified the models also by changing the model architectures and other hyperparameters. Below is the graph showing accuracy and loss of model with the models' architectures and sample sizes.



In the first model shown in the graph, we combined a 1D convolutional layer with an embedding layer. **However, the model's accuracy dropped from 88% to 84% after adding the convolutional layer. This decrease can be attributed to the sequential nature of**

language. While CNNs are excellent at capturing local patterns and are ideal for tasks where input order is irrelevant, such as image recognition, sentiment analysis requires understanding contextual and relational nuances among words, which involves capturing long-range dependencies. Therefore, architectures like Recurrent Neural Networks (RNNs) are more suitable for sentiment analysis due to their ability to handle sequential data effectively.

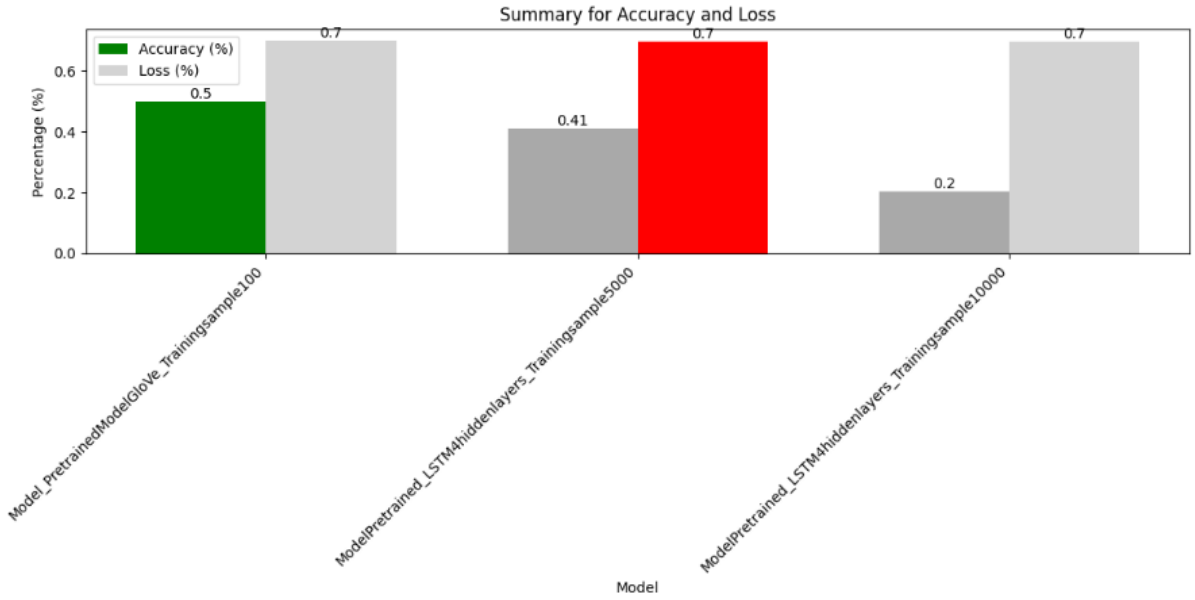
The second model was trained using a one-hot encoder and achieved an accuracy of 88% with a training sample size of 10,000. The next step involves integrating an LSTM model with an embedding layer to evaluate how this combination, known for capturing sequential dependencies, influences overall model performance.

We trained two LSTM models (the 3rd and 4th models in the graph) with embedding layers using different training sample sizes. **The model with a larger training sample size (35,000) exhibited superior accuracy (92%) and lower loss (20%) compared to the other model. This improvement is due to the more extensive dataset, which allowed for better learning and capturing of nuanced patterns, resulting in improved generalization and predictive performance**.

The 5th model in the graph was trained with masking enabled for the embedding layer, but this did not improve accuracy. In fact, its performance was lower than the 4th model (LSTM with 35,000 training samples).

Having explored convolutional networks, one-hot encoding, and LSTM models with embedding layers, another impactful architecture to consider is transformers. Known for their effectiveness in handling text and sequence data, transformers excel at capturing intricate, long-range dependencies within sequences. The 6th model in the graph uses a transformer decoder, but its accuracy is still slightly lower than that of the LSTM model.

We trained three different models using **pre-trained "GloVe"** embeddings with varying amounts of training data and observed an interesting trend. As we increased the training data, the model's Accuracy reduced However, when we simplified the model by reducing the number of LSTM hidden layers to prevent overfitting and used more training data, the Accuracy decreased further.



In the last three models we noticed an interesting trend: increasing the training data from 100 to 5,000 accuracy decreased from 50% to 41%. However, when I simplified the model by reducing the number of LSTM hidden layers to avoid overfitting and used even more training data (10,000), the accuracy dropped to 20% instead of increasing. Additionally, this model had the highest loss compared to all other models above.

Conclusion

Our exploration of various sentiment analysis models revealed significant trends and trade-offs. Increasing the training sample sizes consistently enhanced model performance, highlighting the importance of data volume. Convolutional layers proved less effective, suggesting that **sequential architectures like LSTM are more suitable for sentiment analysis.**

LSTM models with embedding layers consistently outperformed other architectures, achieving the highest accuracy (93%) among all models. Interestingly, introducing transformer architectures did not surpass the performance of LSTM models in this context.

“GloVe” embeddings showed the lowest results. highlighting the challenge of balancing model complexity.

Best Model: an LSTM with embedding layers and 35,000 training samples, achieved the highest accuracy (93%) and the lowest loss (20%) among all models, demonstrating its robust performance in sentiment analysis.

Worst Model: Pretrained model, which incorporated "GloVe" embeddings with reduced LSTM layers and 10,000 training samples, exhibited the lowest accuracy (20% and the highest loss (70%) among all models, underscoring the importance of thoughtful model simplification to avoid overfitting.

These findings underscore the nuanced interplay of architecture, training data, and embeddings in sentiment analysis. Future model development should consider these insights for optimal performance.

Recommendations,

After closely examining various ways to improve the models, here are some recommendations to enhance their performance:

1. **Prioritize LSTM Models with Embedding Layers**: Given the consistently superior performance of LSTM models with embedding layers, it is recommended to prioritize this architecture for sentiment analysis tasks. LSTMs excel at capturing sequential dependencies, which aligns well with the nuanced nature of sentiment in language.
2. **Expand Training Datasets**: The positive correlation between training data size and model performance underscores the importance of acquiring larger datasets. Efforts should be directed towards expanding training datasets to further enhance model generalization and accuracy.
3. **Optimize GloVe Embeddings**: While GloVe embeddings have shown promise, their effectiveness depends on careful fine-tuning. Future efforts should focus on optimizing GloVe embeddings in conjunction with model architecture to maximize performance.