Assignment 2

Name: Pola Gnana Shekar
Roll No: 21CS10052

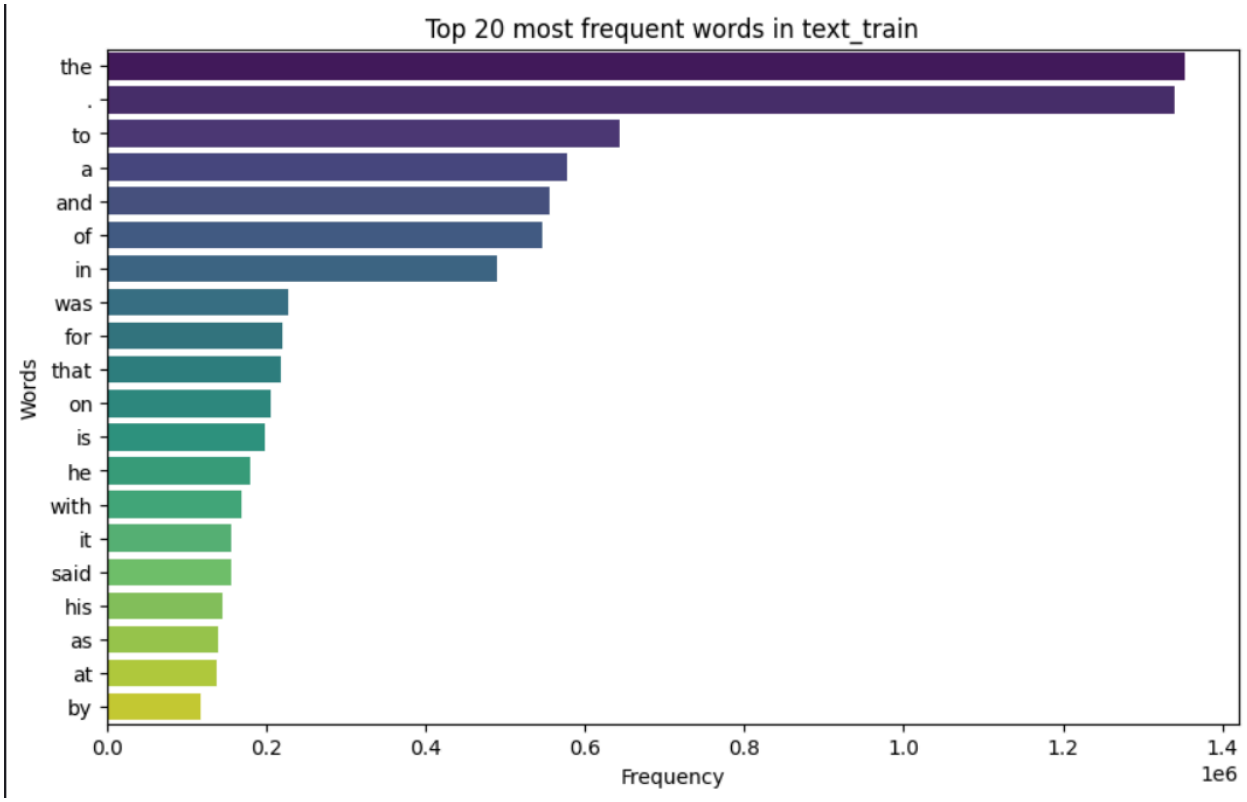
Data exploration findings:

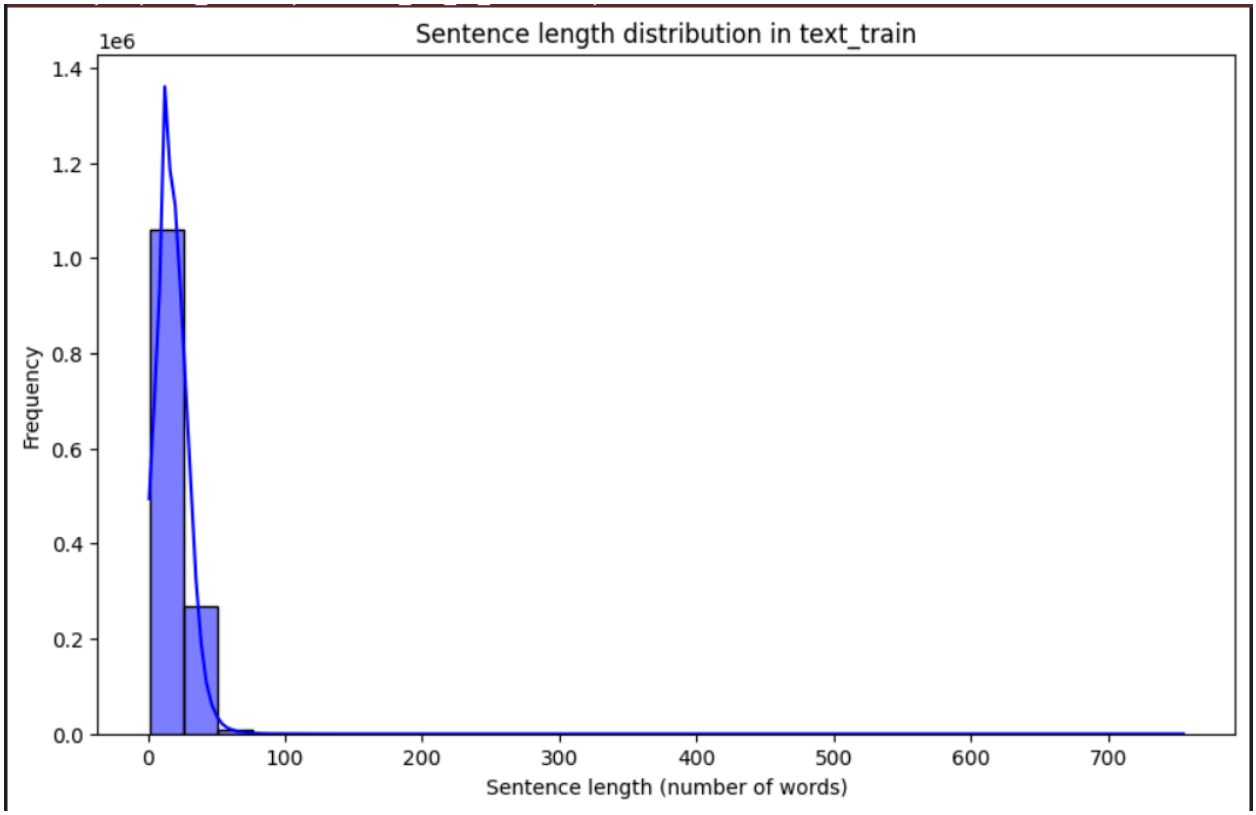
The exploratory data analysis conducted on a subset of the CNN/Daily Mail dataset aimed to understand its structure and linguistic features, utilizing 34,454 samples from the training split and the complete test split of 11,490 samples. Data cleaning was performed to standardize the text by lowercasing and removing special characters, using a regular expression. Visualizations were created to illustrate the distribution of word frequencies and sentence lengths, revealing insights into the dataset's characteristics.

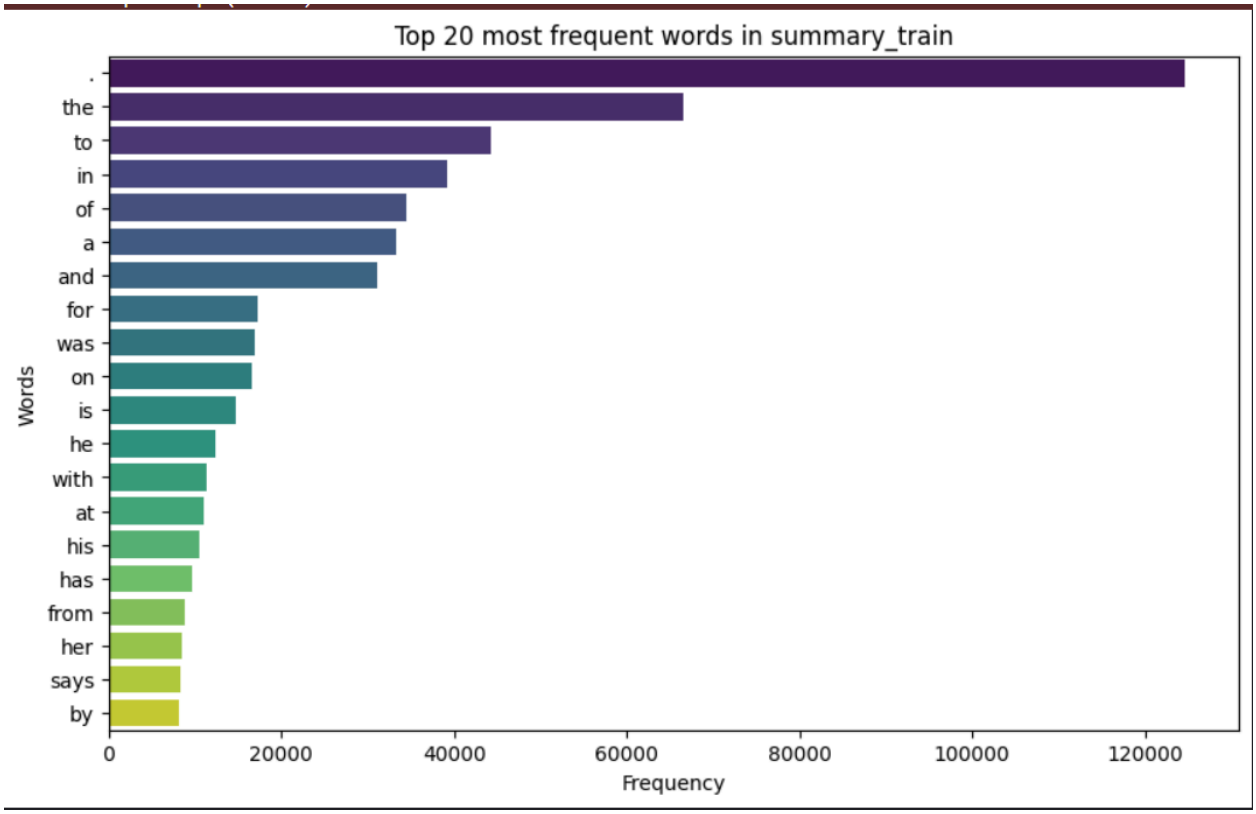
Key Findings:

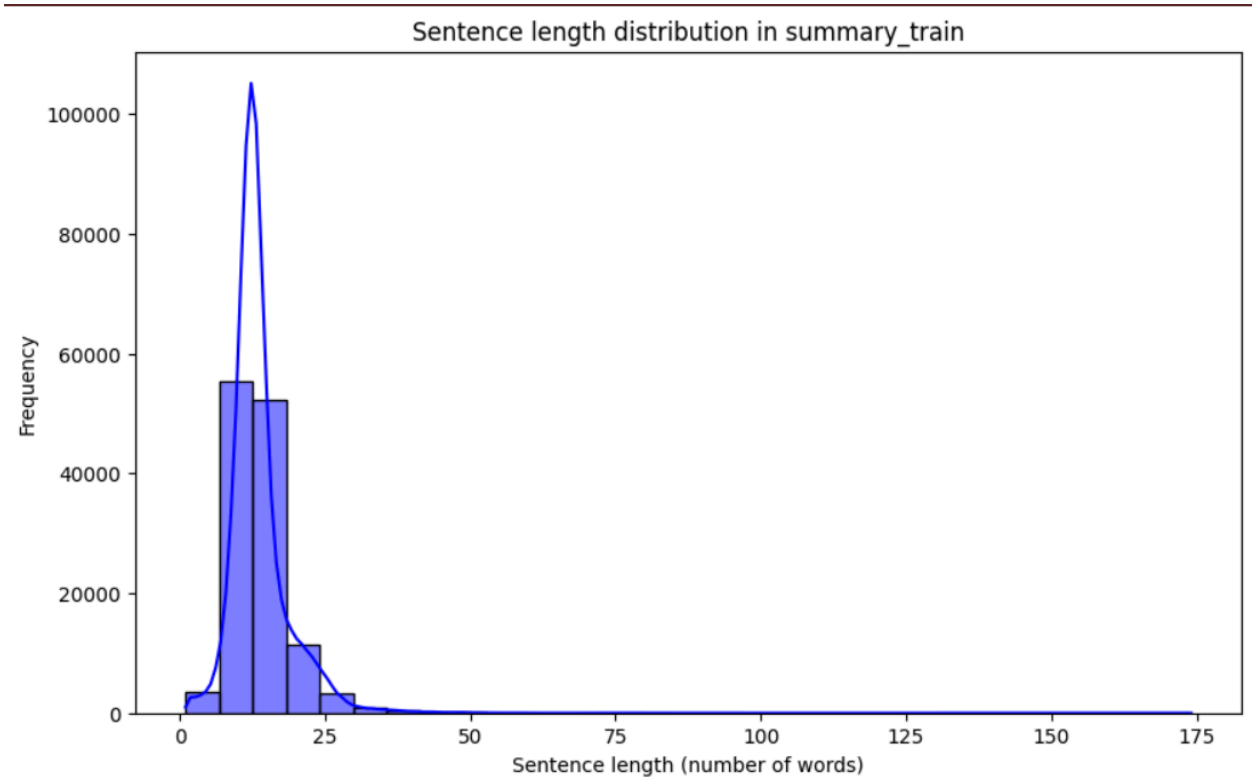
- **Training Dataset Size:** 34,454 samples (12% of CNN/Dailymail dataset)
  10% - for training, 2% acts as a validation set.
- **Test Dataset Size:** 11,490 samples
- **Data Cleaning:** Standardization involved lowercasing and removal of special characters.
- **Frequent Words:** Top 20 most frequent words identified and visualized.
- **Sentence Length Analysis:** Sentence lengths were assessed and plotted for distribution.
- **Visualizations Included:**
    - Word frequency distributions for both articles and summaries.
    - Sentence length distributions for both articles and summaries.

Below are the plots obtained:


Top 20 most frequent words in text_train
the
.
to
a
and
of
in
was
for
that
on
is
he
with
it
said
his
as
at
by
Words
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1e6
Frequency


Sentence length distribution in text_train
1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Frequency
0
100
200
300
400
500
600
700
Sentence length (number of words)


Top 20 most frequent words in summary_train
Words
.
the
to
in
of
a
and
for
was
on
is
he
with
at
his
has
from
her
says
by
0
20000
40000
60000
80000
100000
120000
Frequency


Sentence length distribution in summary_train
Frequency
100000
80000
60000
40000
20000
0
0
25
50
75
100
125
150
175
Sentence length (number of words)

Model Description:

This model is a sequence-to-sequence architecture designed for tasks such as text summarization or machine translation, utilizing Long Short-Term Memory (LSTM) networks and an attention mechanism. The model leverages pre-trained GloVe (Global Vectors for Word Representation) embeddings to initialize the embedding layers, enabling it to understand semantic relationships between words effectively.

Model Architecture Overview:

1. **Inputs**:
    - **Encoder Input**: Accepts sequences of variable length (up to 800 tokens) representing the input text.
    - **Decoder Input**: Accepts sequences of variable length (up to 55 tokens) representing the target text.
2. **Embedding Layers**:
    - **Encoder Embedding**: Maps the encoder input tokens to a dense vector space of 100 dimensions using GloVe embeddings (1,600,000 parameters).
    - **Decoder Embedding**: Maps the decoder input tokens to a similar dense vector space using GloVe embeddings (1,600,000 parameters).
3. **LSTM Layers**:
    - **Encoder LSTM**: Processes the embedded input sequences, maintaining a hidden state across the sequence. It outputs the sequences and the final hidden and cell states (365,568 parameters).
    - **Decoder LSTM**: Processes the embedded target sequences, using the final hidden and cell states from the encoder as initial states. It outputs sequences and maintains hidden states (365,568 parameters).
4. **Attention Mechanism**:
    - The attention layer computes a context vector by weighting the encoder outputs based on the decoder's current state. This allows the decoder to focus on relevant parts of the input sequence dynamically (0 parameters).
5. **Concatenation Layer**:
    - Combines the context vector and the decoder outputs to enhance the information fed into the output layer (0 parameters).
6. **Output Layer**:
    - A Dense layer that outputs a probability distribution across the vocabulary for each token in the output sequence, using a softmax activation function. This layer has a significant number of parameters (8,208,000) due to the size of the vocabulary.

Summary of Model Parameters:

- **Total Parameters**: 12,139,136 (approximately 46.31 MB)
- **Trainable Parameters**: 8,939,136 (approximately 34.10 MB)
- **Non-trainable Parameters**: 3,200,000 (approximately 12.21 MB)

This architecture allows for effective learning of complex sequential dependencies and relationships in text, making it suitable for generating coherent and contextually relevant summaries or translations. The inclusion of GloVe embeddings enhances its capacity to capture word semantics right from the beginning of training.

Model training:

The model was trained for **100 epochs** with a **batch size of 64** to optimize its performance on the text summarization task. The training process involved fitting the model to the training data while evaluating its performance on a validation set at each epoch. The model aims to learn the relationships between input text and their corresponding summaries, using a sequence-to-sequence approach with attention.

Training Parameters:

- **Epochs**: 100
- **Batch Size**: 64

Training Procedure:

- The model was trained using the following configurations:
    - **Input Data**: The input consisted of `text_train_final` for the encoder and `summary_train_final[:, :-1]` for the decoder, utilizing all tokens of the summary except the last token.
    - **Output Data**: The output comprised `summary_train_final[:, 1:]`, which includes all tokens of the summary except the first token to facilitate teacher forcing.
- The training history recorded metrics for both training and validation datasets, allowing for performance monitoring.

Training Performance:

- **1st Epoch**:
    - **Accuracy**: 26.70%
    - **Loss**: 4.7857
    - **Validation Accuracy**: 23.91%
    - **Validation Loss**: 5.9806
    - **Time**: 88 seconds
- **100th Epoch**:
    - **Accuracy**: 43.56%
    - **Loss**: 3.1194
    - **Validation Accuracy**: 21.77%
    - **Validation Loss**: 7.9357
    - **Time**: 142 seconds

The training metrics indicate that the model improved its accuracy on the training data over the epochs, reaching 43.56% accuracy by the 100th epoch. However, the validation accuracy decreased, suggesting potential overfitting, as indicated by the higher validation loss compared to the training loss. The training and validation metrics provide insight into the model's learning dynamics and highlight areas for further improvement, such as regularization or data augmentation strategies to enhance generalization.

Evaluation results:

The model's performance was evaluated on two datasets: the CNN/Daily Mail dataset and the Wikipedia dataset. Due to time constraints, a limited number of samples were assessed from each dataset.

CNN/Daily Mail Dataset:

- **Samples Evaluated**: 114 (1% of test split)
- **Average ROUGE-2 Score**: 0.0
- **Average ROUGE-L Score**: 0.0123

Wikipedia Dataset:

- **Samples Evaluated**: 1000
- **Average ROUGE-2 Score**: 1.63e-05
- **Average ROUGE-L Score**: 0.0041

Observations and Inferences:

The evaluation metrics, specifically the ROUGE scores, indicate that the model's performance on both datasets was quite low. The ROUGE-2 scores for both datasets suggest that there was minimal overlap between the generated summaries and the reference summaries, indicating that the model struggled to capture the relevant n-grams. Similarly, the ROUGE-L scores, which evaluate the longest common subsequence between the generated and reference summaries, also reflect poor performance.

These results highlight the need for further refinement of the model, which may involve optimizing the training process, experimenting with different architectures, or utilizing additional data for training to enhance summarization quality.

Limitations:

The implementation and evaluation of the model faced several limitations primarily due to resource constraints, which significantly impacted the training and prediction processes:

- **Limited Batch Size**: Due to restrictions on RAM size and GPU memory, we had to use a smaller batch size during training. This decision led to increased training time and necessitated the use of a reduced training dataset, further impacting the model's ability to learn effectively.
- **Resource Constraints**: The limited resources not only affected the training phase but also posed challenges during the prediction phase. The RAM frequently reached capacity, resulting in slower processing times and potential memory-related issues. This limitation hindered the ability to process larger datasets efficiently.
- **Downloading Datasets**: Downloading the Wikipedia dataset proved to be cumbersome, as the resource constraints meant that we could not store or handle the entire dataset simultaneously. This limitation forced us to work with a smaller sample size, which may not have been representative of the full dataset.
- **File Management for Continuity**: To mitigate the impact of these constraints, I implemented a strategy to save each output and data locally. By doing so, I could reload previously processed files and models, allowing me to avoid starting from scratch when continuing my work. This approach facilitated resuming from where I left off, particularly after downloading files and models, ensuring continuity in the workflow.

As a result of these factors, the model experienced suboptimal training conditions, leading to poor performance and low evaluation metrics. Future work would benefit from addressing these resource constraints, allowing for larger batch sizes and more comprehensive datasets to enhance training efficacy and predictive accuracy.