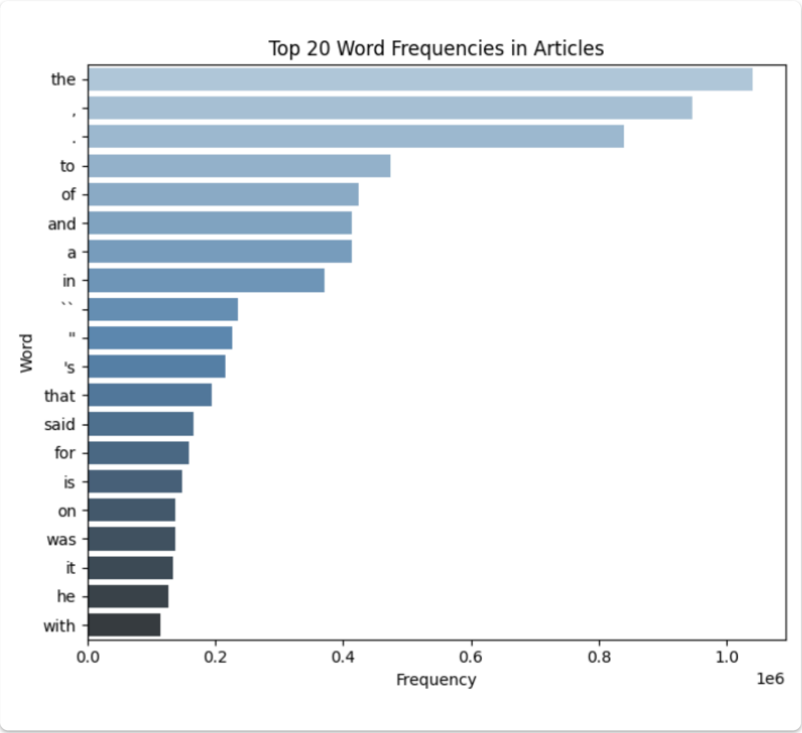
NLP Assignment 2 report

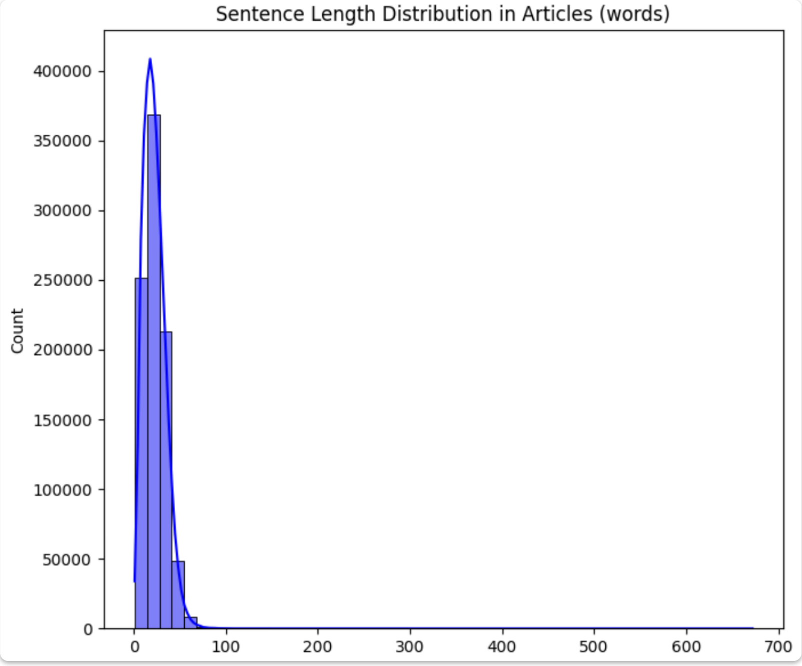
Name : Navaneeth Shaji
Roll : 21CS30032

Data Exploration Findings

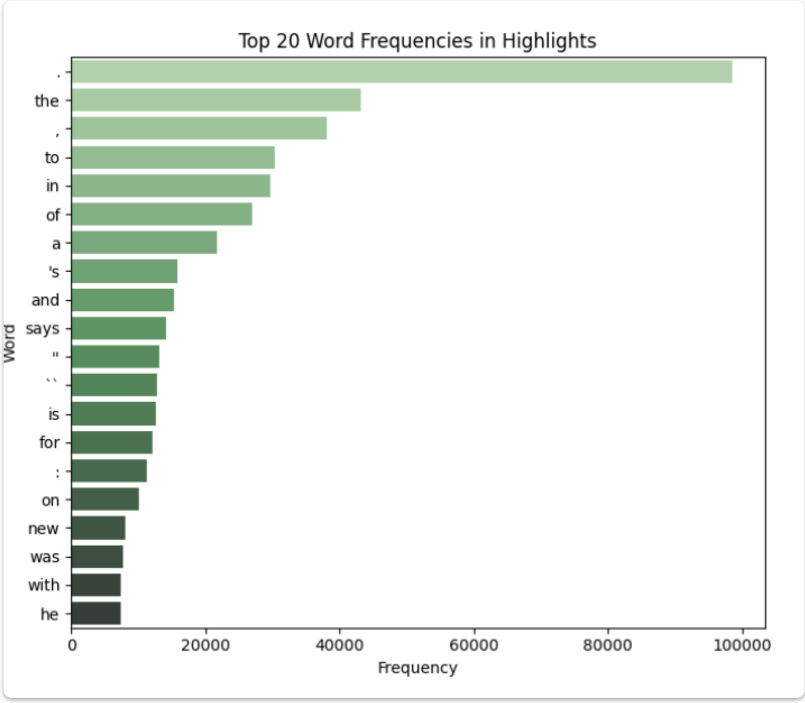
The data exploration was done on the words (with the frequency) and sentence lengths
This was done without any preprocessing done to the train and test set

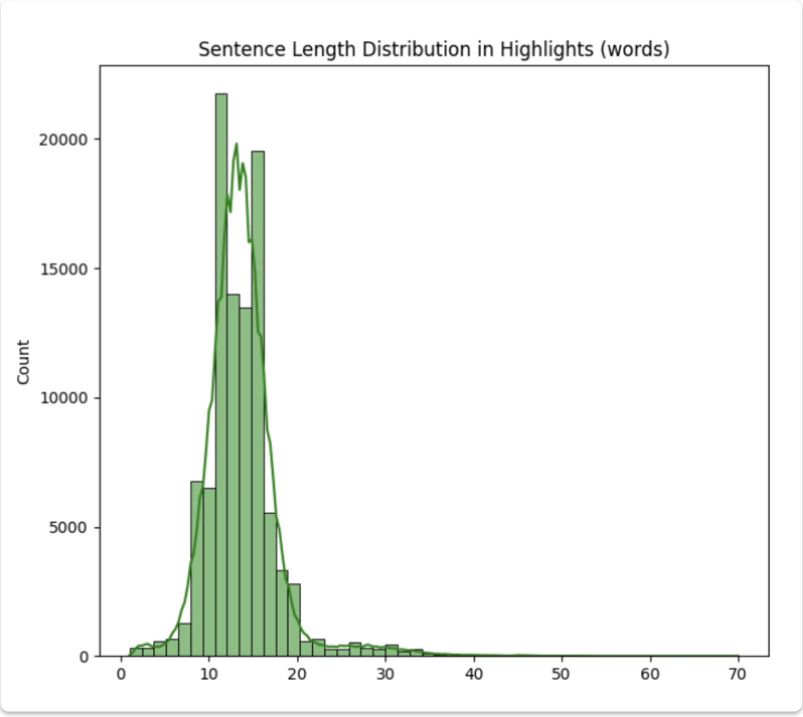
On the train set , the observations are as follows :





On the test set , the observations are as follows :





Data preprocessing

I have done the following Preprocessing to the dataset before using it

1. Converted all text to lowercase in the article and the summary
2. removed all punctuations and symbols
3. tokenised the sentences using nltk
4. reduced the size of the ariticles to atmost 768 words

Model Training and Evaluation Result

Dataset Size
Training : 28000 samples (the first 28000 samples of the dataset)
Test : 11490 samples

The model structure is like :
Seq2Seq(
(encoder): Encoder(
(embedding): Embedding(166578, 256)
(rnn): LSTM(256, 512, num_layers=2, dropout=0.2)
(dropout): Dropout(p=0.2, inplace=False)
)
(decoder): Decoder(
(attention): Attention()
(embedding): Embedding(166578, 256)
(rnn): LSTM(256, 512, num_layers=2, dropout=0.2)
(fc_out): Linear(in_features=1024, out_features=166578, bias=True)
(dropout): Dropout(p=0.2, inplace=False)
)
)

Training

This was how the training phase panned out :
10%|█ | 1/10 [29:52<4:28:49, 1792.17s/it]

```
Train Loss:   7.570 | Train PPL: 1938.630
```

20%|██ | 2/10 [59:44<3:58:58, 1792.32s/it]

```
Train Loss:   6.583 | Train PPL: 722.691
```

30%|███ | 3/10 [1:29:39<3:29:15, 1793.71s/it]

```
Train Loss:   6.010 | Train PPL: 407.472
```

40%|████ | 4/10 [1:59:32<2:59:20, 1793.37s/it]

```
Train Loss:   5.525 | Train PPL: 250.952
```

50%|█████ | 5/10 [2:29:28<2:29:31, 1794.22s/it]

```
Train Loss:   5.094 | Train PPL: 163.031
```

60%|██████ | 6/10 [2:59:25<1:59:39, 1794.99s/it]

```
Train Loss:   4.711 | Train PPL: 111.164
```

70%|███████ | 7/10 [3:29:21<1:29:46, 1795.48s/it]

```
Train Loss:   4.383 | Train PPL:  80.086
```

80%|████████ | 8/10 [3:59:20<59:52, 1796.45s/it]

```
Train Loss:   4.103 | Train PPL:  60.528
```

90%|█████████ | 9/10 [4:29:19<29:57, 1797.51s/it]

```
Train Loss:   3.864 | Train PPL:  47.661
```

100%|██████████| 10/10 [4:59:24<00:00, 1796.40s/it]

```
Train Loss:   3.650 | Train PPL:  38.493
```

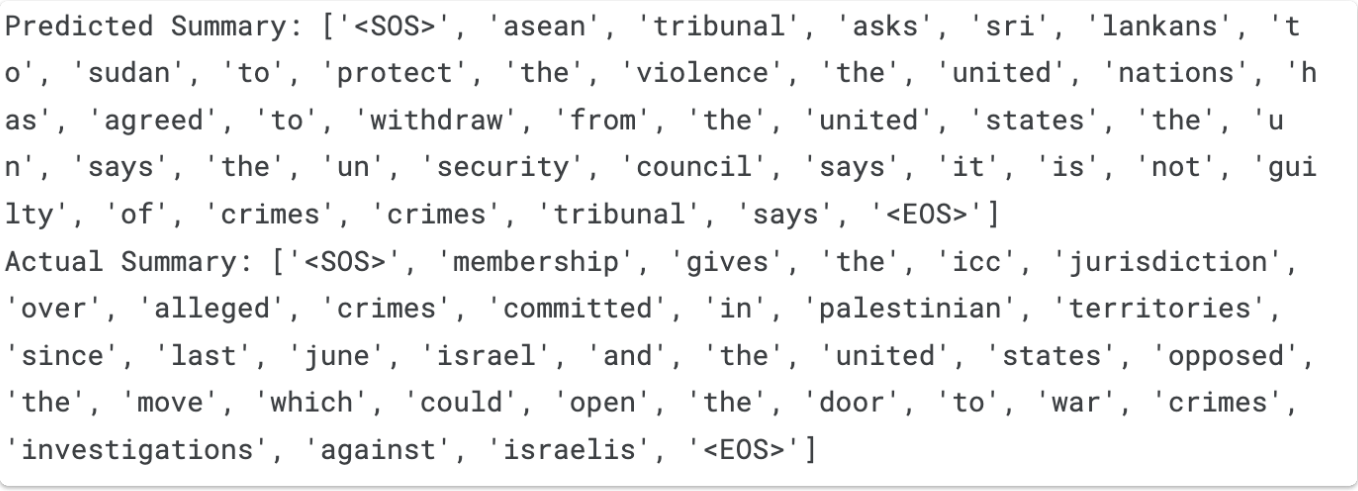
Final Train loss = 3.650

The high train loss can be attributed to the the fact that the each of the training sample is pretty big . Some of the ariticles have well over 1000 words. For the encoder, it becomes really difficult to encode all the content of the articles when they are really big. So the performance of the encoder is not that great which also leads to poor predictions and thus the high loss value.

Scores on the test set :
ROUGE-2 Score: 0.01427
ROUGE-L Score: 0.14807

The model was then tested on the Wikipedia summary dataset
the scores are as shown :
ROUGE-2 Score: 0.02094
ROUGE-L Score: 0.12217

Heres a Sample summarization :
Predicted Summary :



```
Predicted Summary: ['<SOS>', 'asean', 'tribunal', 'asks', 'sri', 'lankans', 'to', 'sudan', 'to', 'protect', 'the', 'violence', 'the', 'united', 'nations', 'has', 'agreed', 'to', 'withdraw', 'from', 'the', 'united', 'states', 'the', 'un', 'says', 'the', 'un', 'security', 'council', 'says', 'it', 'is', 'not', 'guilty', 'of', 'crimes', 'crimes', 'tribunal', 'says', '<EOS>']
Actual Summary: ['<SOS>', 'membership', 'gives', 'the', 'icc', 'jurisdiction', 'over', 'alleged', 'crimes', 'committed', 'in', 'palestinian', 'territories', 'since', 'last', 'june', 'israel', 'and', 'the', 'united', 'states', 'opposed', 'the', 'move', 'which', 'could', 'open', 'the', 'door', 'to', 'war', 'crimes', 'investigations', 'against', 'israelis', '<EOS>']
```

Observations

The performance of the model is that not great. This is due to the following reasons :

1. the primary reason can be attributed to the limitation in the size of the model being used. The 16gb gpu cap in Kaggle limits us from increasing the size of the model beyond a certain limit.
2. The smaller model also makes it difficult for its encoder to capture details of the long articles in the dataset. this results in poor results from the encoder layer which also gives poor results
3. Also there are some smaller points like the way rouge scores are measured. These scores are measured on the basis of how many words are similar between the prediction and ground truth. Due to poor performance shown above rouge scores are pretty low , although the model does predict meaningful sentences. On the other hand we can get much higher rouge scores if we have predictions with just stopwords in them.