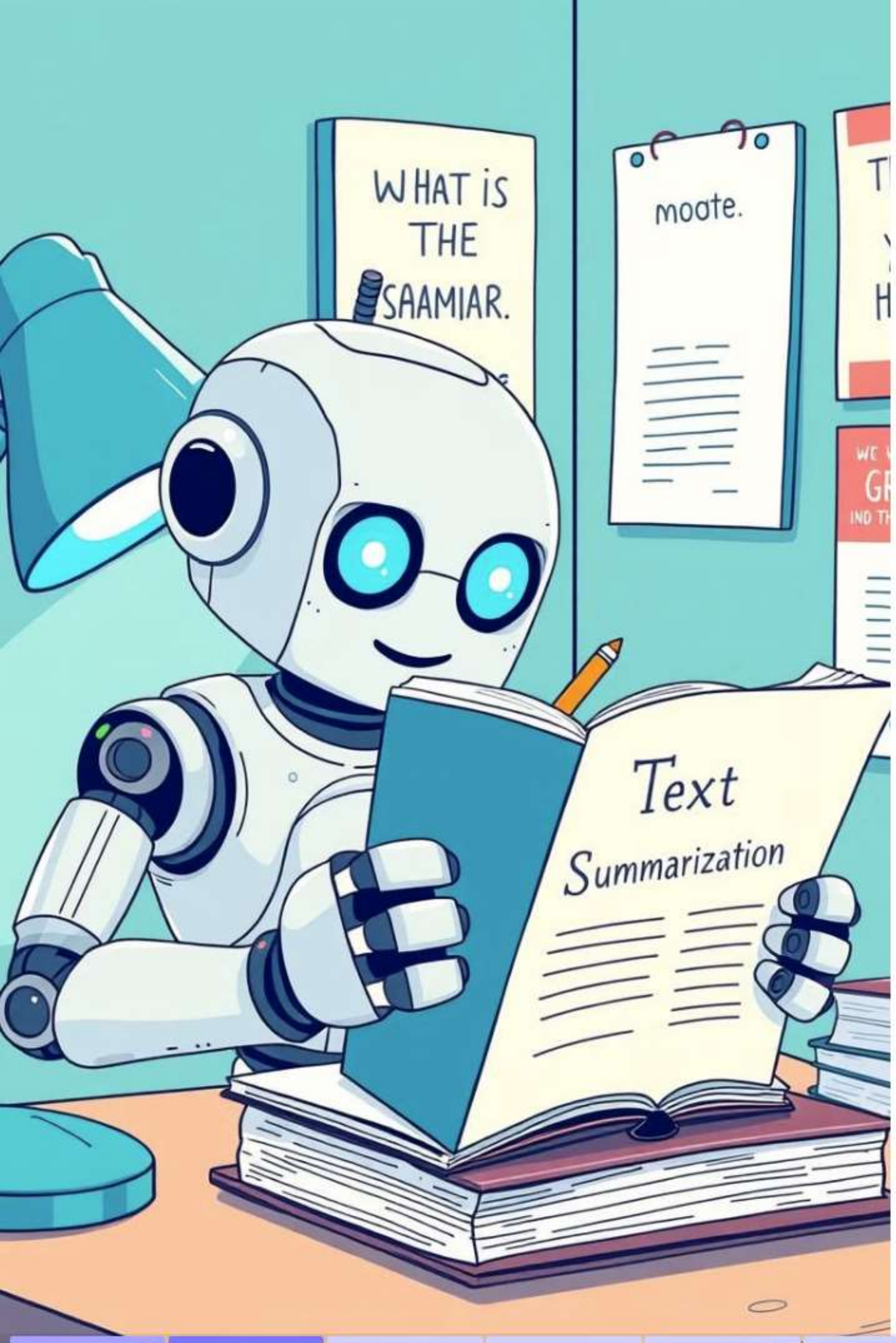
Project Overview: Text Summarization with LSTM for News

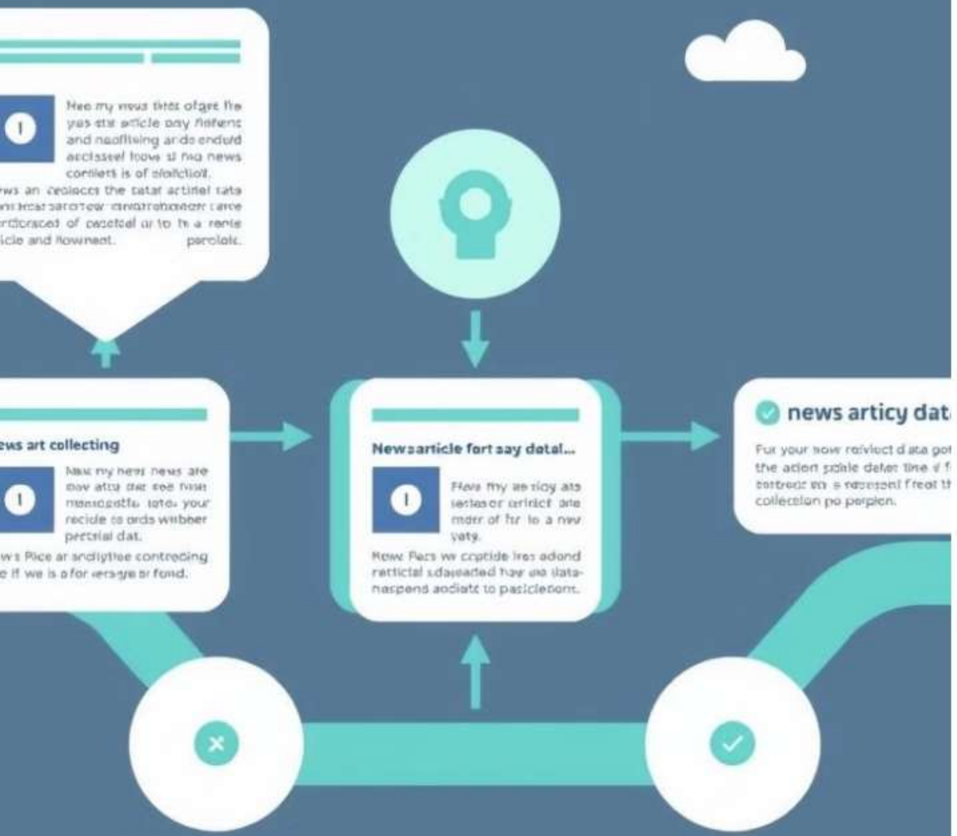
This project automates news article summarization. It leverages LSTM deep learning models. Key technologies are Python and related libraries. Text summarization is crucial in today's fast-paced world. It helps us consume information efficiently.

by Meghana Naidu T



Introduction to Text Summarization

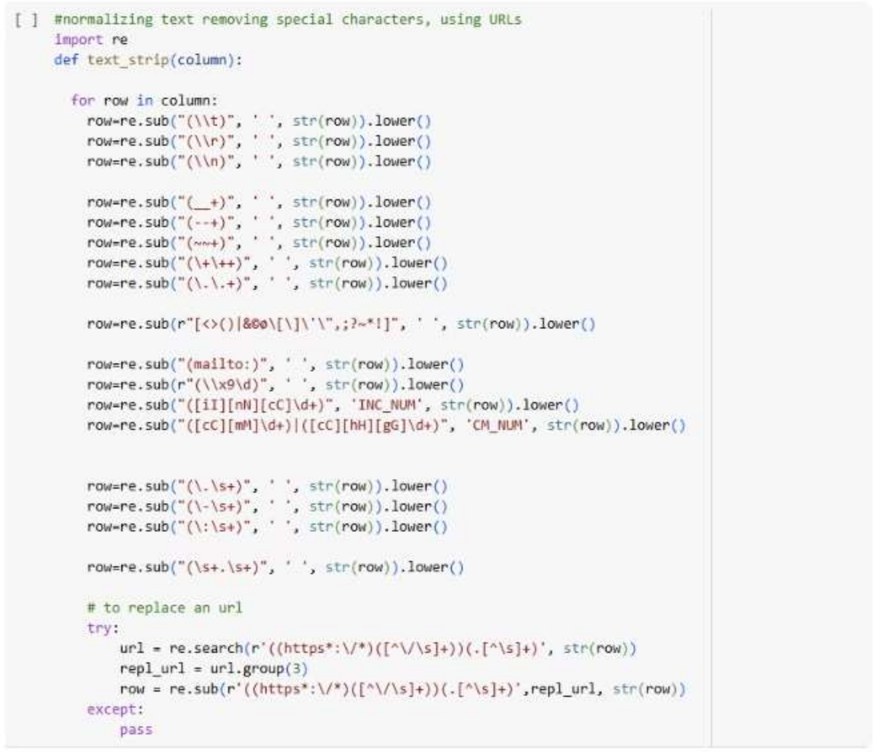
Text summarization condenses text. Extractive methods select important sentences. Abstractive methods rephrase the original content. Summarizing long articles presents challenges. Existing methods range from traditional to deep learning.



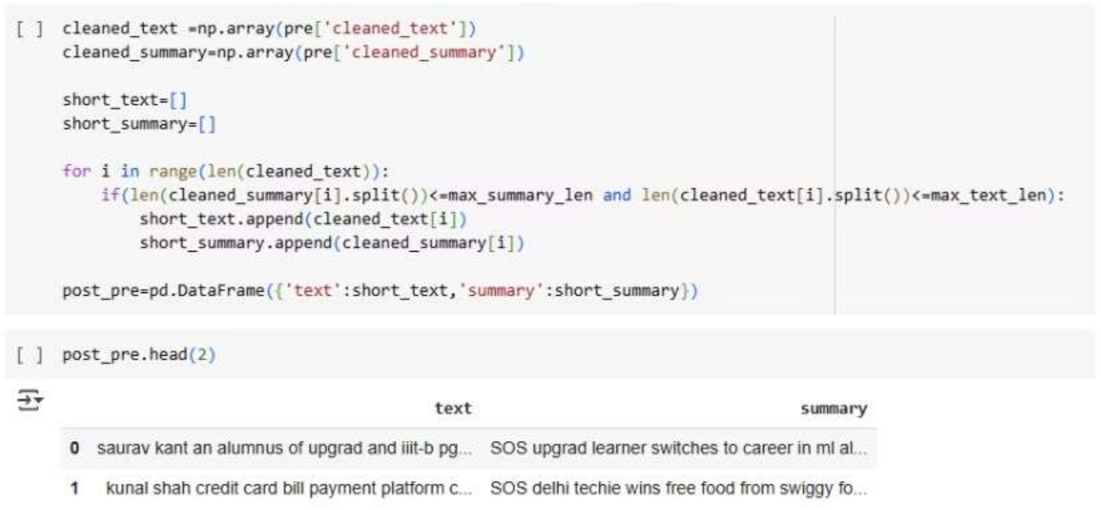
Data Collection and Preprocessing

News data is collected for model training. Preprocessing steps clean and prepare data. Cleaning removes punctuation and special characters. Tokenization splits text into smaller units. Stop word removal eliminates common words. Lowercasing makes text uniform.

Merged datasets, cleaned text (lowercase, symbols removed) Used SpaCy for tokenization



```
[ ] #normalizing text removing special characters, using URLs
    import re
    def text_strip(column):

      for row in column:
        row=re.sub("(\\t)", ' ', str(row)).lower()
        row=re.sub("(\\r)", ' ', str(row)).lower()
        row=re.sub("(\\n)", ' ', str(row)).lower()

        row=re.sub("(__+)", ' ', str(row)).lower()
        row=re.sub("(--+)", ' ', str(row)).lower()
        row=re.sub("(~~+)", ' ', str(row)).lower()
        row=re.sub("(\+\++)", ' ', str(row)).lower()
        row=re.sub("(\.\.+)", ' ', str(row)).lower()

        row=re.sub(r"[<>()|&©ø\[\]\'\",;?~*!]", ' ', str(row)).lower()

        row=re.sub("(mailto:)", ' ', str(row)).lower()
        row=re.sub(r"(\\x9\d)", ' ', str(row)).lower()
        row=re.sub("([iI][nN][cC]\d+)", 'INC_NUM', str(row)).lower()
        row=re.sub("([cC][mM]\d+)|([cC][hH][gG]\d+)", 'CM_NUM', str(row)).lower()


        row=re.sub("(\.\s+)", ' ', str(row)).lower()
        row=re.sub("(\-\s+)", ' ', str(row)).lower()
        row=re.sub("(\:\s+)", ' ', str(row)).lower()

        row=re.sub("(\s+.\s+)", ' ', str(row)).lower()

        # to replace an url
        try:
            url = re.search(r'((https*:\/*)([^\/\s]+))(.[^\s]+)', str(row))
            repl_url = url.group(3)
            row = re.sub(r'((https*:\/*)([^\/\s]+))(.[^\s]+)',repl_url, str(row))
        except:
            pass
```

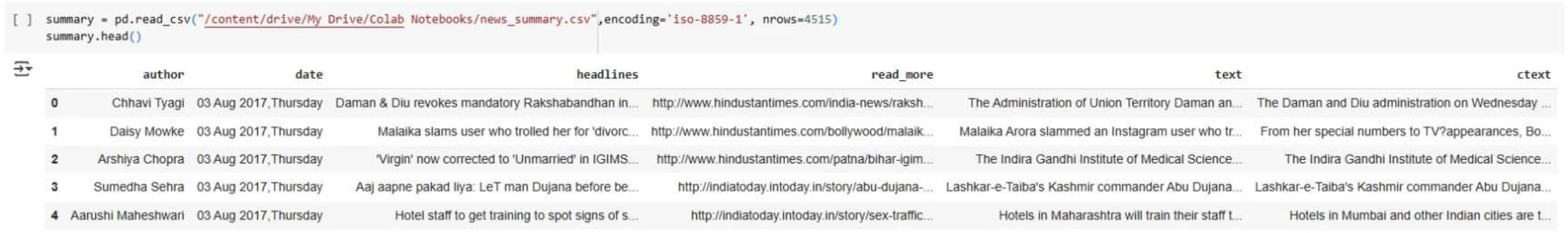


```
[ ] cleaned_text =np.array(pre['cleaned_text'])
    cleaned_summary=np.array(pre['cleaned_summary'])

    short_text=[]
    short_summary=[]

    for i in range(len(cleaned_text)):
        if(len(cleaned_summary[i].split())<=max_summary_len and len(cleaned_text[i].split())<=max_text_len):
            short_text.append(cleaned_text[i])
            short_summary.append(cleaned_summary[i])

    post_pre=pd.DataFrame({'text':short_text,'summary':short_summary})
```

```
[ ] post_pre.head(2)
```

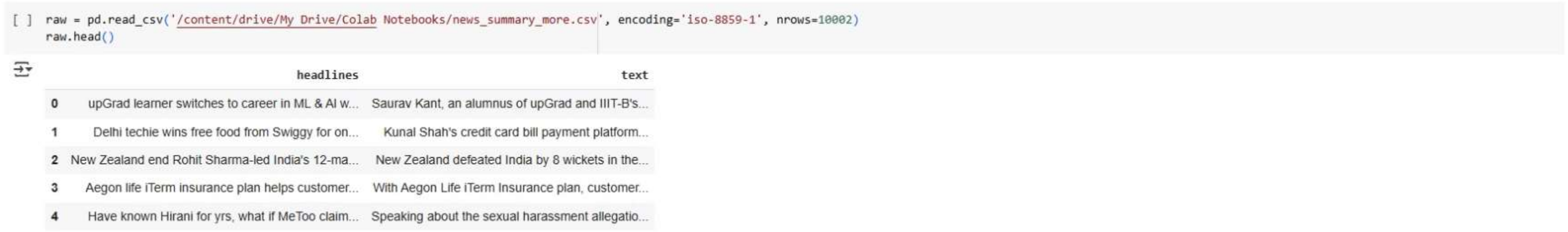
	text	summary
0	saurav kant an alumnus of upgrad and iiit-b pg...	SOS upgrad learner switches to career in ml al...
1	kunal shah credit card bill payment platform c...	SOS delhi techie wins free food from swiggy fo...

The project uses two primary datasets: news_summary.csv, containing 4,515 news articles, and news_summary_more.csv, which includes 10,002 additional entries. These datasets consist of English news articles, primarily sourced from Indian news outlets. Key fields include the full article text (text), the corresponding summaries (headlines), and their cleaned versions (ctext and cheadline) in the extended dataset. To prepare the data for modeling, both datasets were merged, and preprocessing steps such as removing duplicates, handling null values, lowercasing, and eliminating stopwords and special characters were performed. After cleaning, around 13,000 high-quality article-summary pairs were retained. This refined dataset serves as the foundation for training the LSTM-based model to learn and generate effective text summaries.



```
summary = pd.read_csv("/content/drive/My Drive/Colab Notebooks/news_summary.csv",encoding='iso-8859-1', nrows=4515)
summary.head()
```

	author	date	headlines	read_more	text	ctext
0	Chhavi Tyagi	03 Aug 2017,Thursday	Daman & Diu revokes mandatory Rakshabandhan in...	http://www.hindustantimes.com/india-news/raksh...	The Administration of Union Territory Daman an...	The Daman and Diu administration on Wednesday ...
1	Daisy Mowke	03 Aug 2017,Thursday	Malaika slams user who trolled her for 'divorc...	http://www.hindustantimes.com/bollywood/malaik...	Malaika Arora slammed an Instagram user who tr...	From her special numbers to TV?appearances, Bo...
2	Arshiya Chopra	03 Aug 2017,Thursday	'Virgin' now corrected to 'Unmarried' in IGIMS...	http://www.hindustantimes.com/patna/bihar-igim...	The Indira Gandhi Institute of Medical Science...	The Indira Gandhi Institute of Medical Science...
3	Sumedha Sehra	03 Aug 2017,Thursday	Aaj aapne pakad liya: LeT man Dujana before be...	http://indiatoday.intoday.in/story/abu-dujana-...	Lashkar-e-Taiba's Kashmir commander Abu Dujana...	Lashkar-e-Taiba's Kashmir commander Abu Dujana...
4	Aarushi Maheshwari	03 Aug 2017,Thursday	Hotel staff to get training to spot signs of s...	http://indiatoday.intoday.in/story/sex-traffic...	Hotels in Maharashtra will train their staff t...	Hotels in Mumbai and other Indian cities are t...



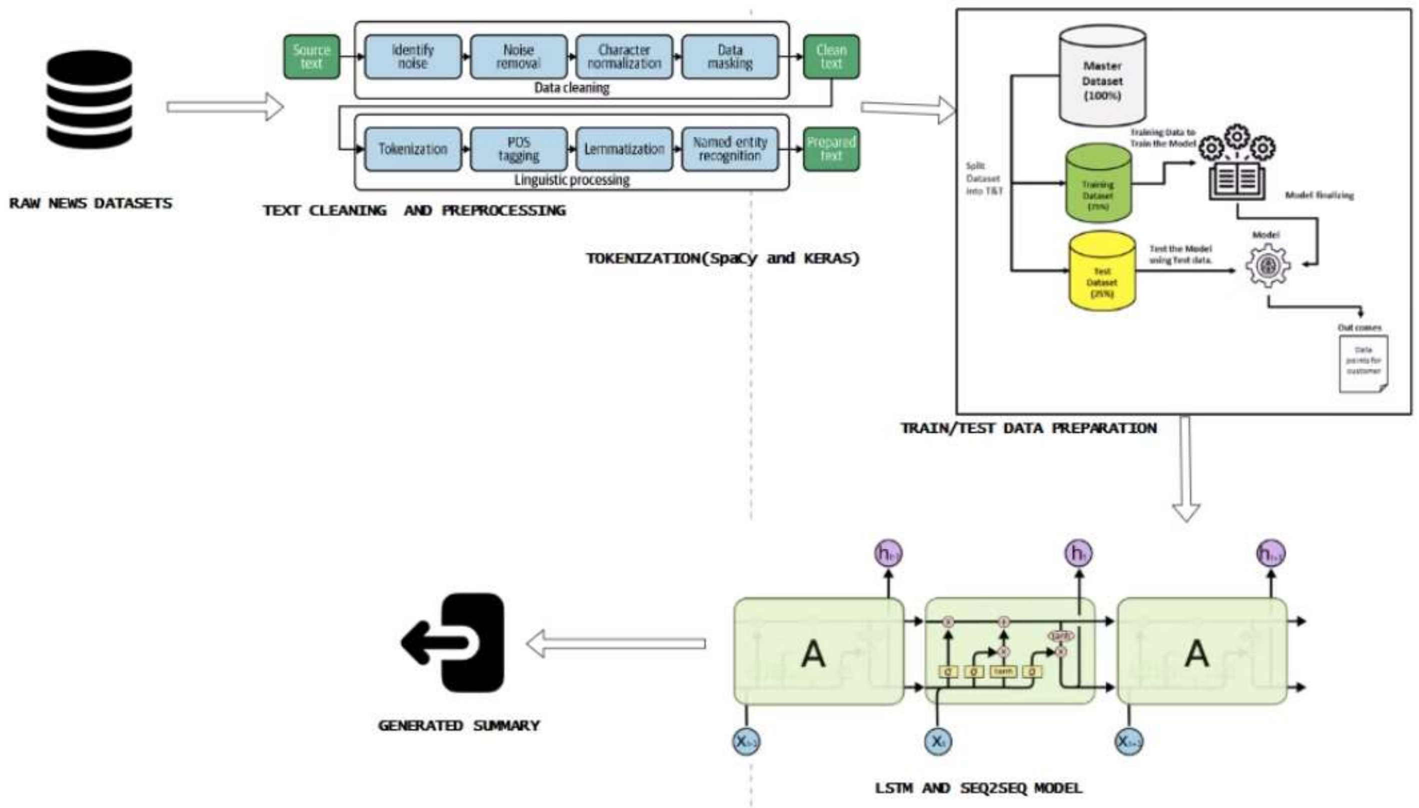
```
raw = pd.read_csv('/content/drive/My Drive/Colab Notebooks/news_summary_more.csv', encoding='iso-8859-1', nrows=10002)
raw.head()
```

	headlines	text
0	upGrad learner switches to career in ML & Al w...	Saurav Kant, an alumnus of upGrad and IIIT-B's...
1	Delhi techie wins free food from Swiggy for on...	Kunal Shah's credit card bill payment platform...
2	New Zealand end Rohit Sharma-led India's 12-ma...	New Zealand defeated India by 8 wickets in the...
3	Aegon life iTerm insurance plan helps customer...	With Aegon Life iTerm Insurance plan, customer...
4	Have known Hirani for yrs, what if MeToo claim...	Speaking about the sexual harassment allegatio...

LSTM Deep Learning Model Explained

LSTM architecture utilizes memory cells. LSTMs effectively handle long-range text dependencies. This makes them suitable for text summarization. Benefits include improved context understanding. LSTM's architecture provides high accuracy.

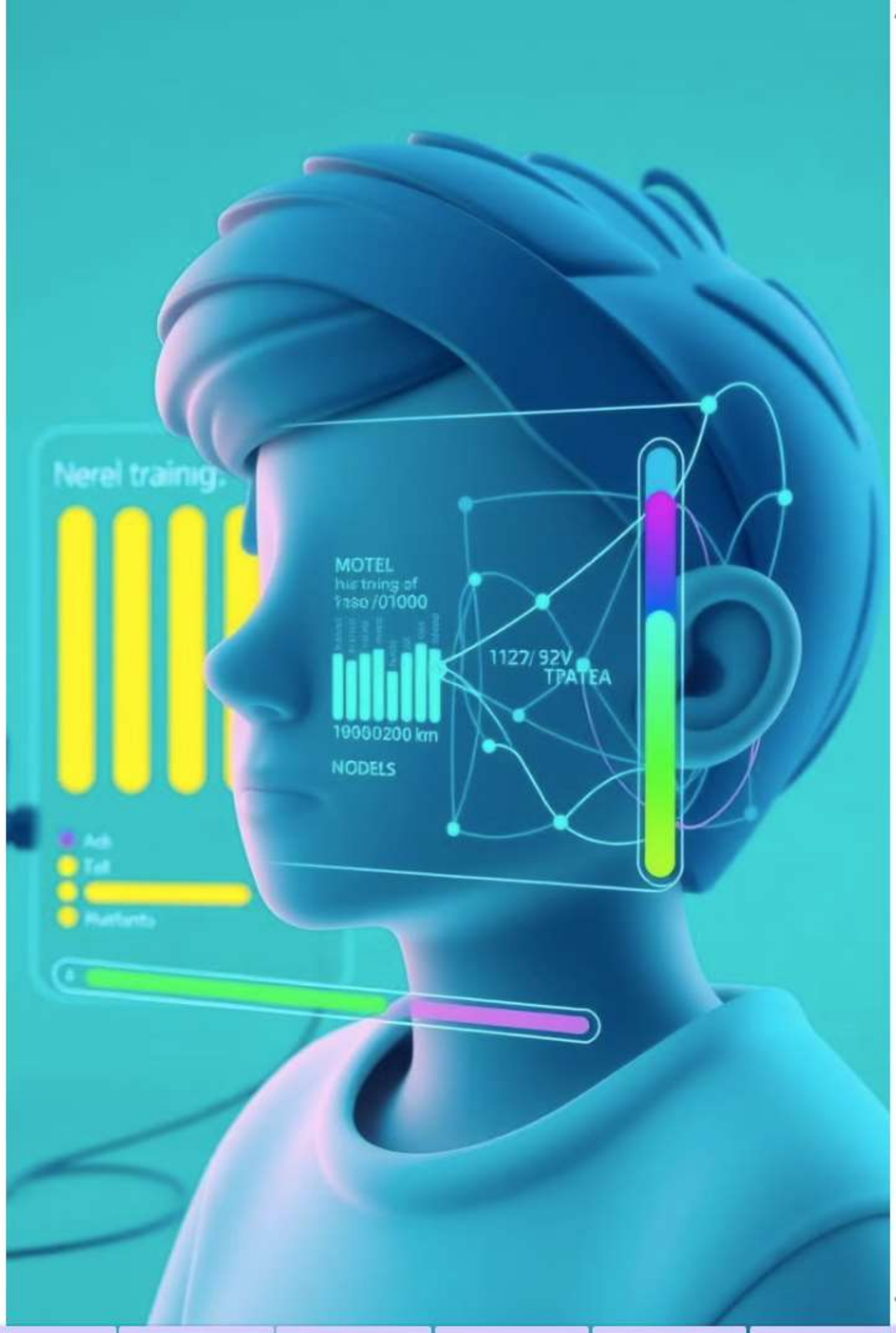
Architecture diagram

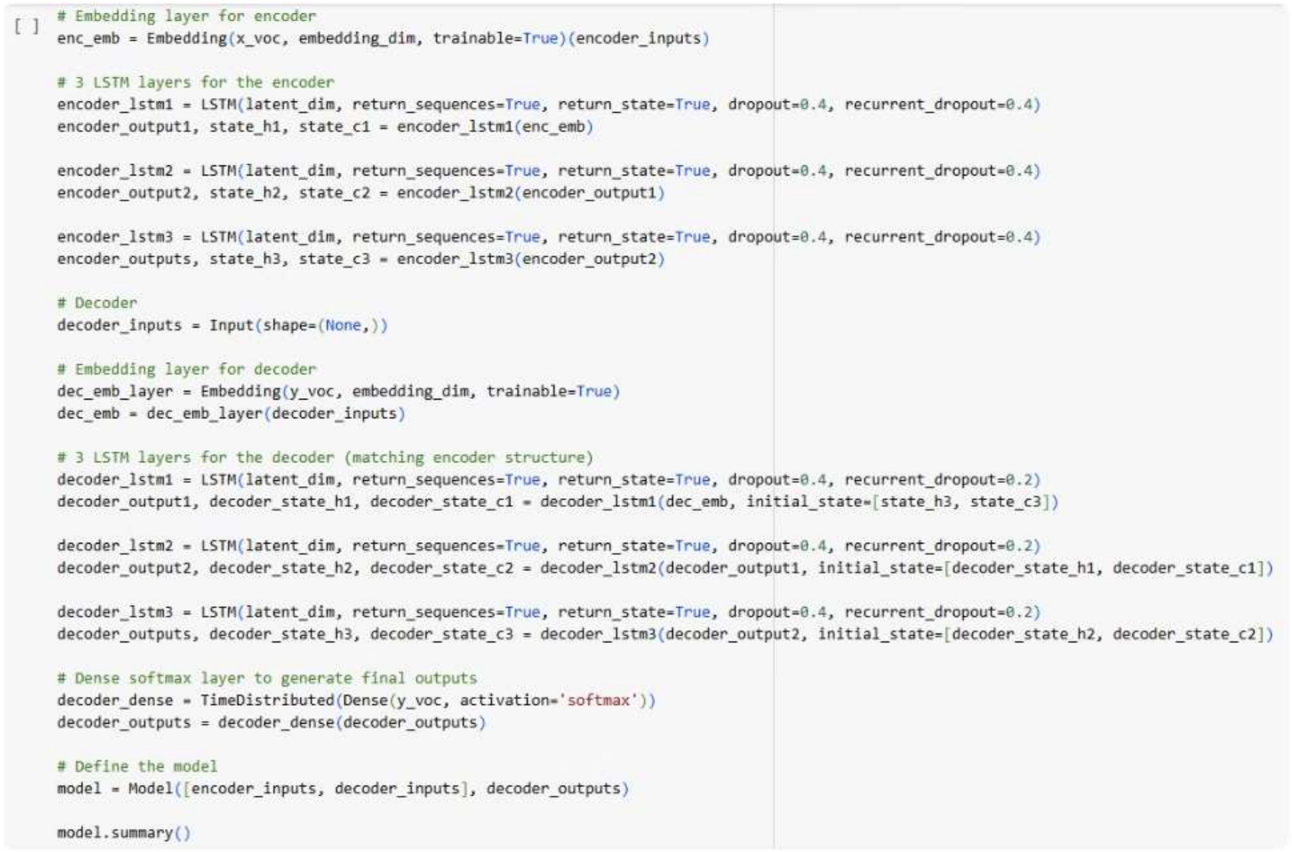


The model uses an LSTM-based Sequence-to-Sequence (Seq2Seq) architecture with an encoder-decoder structure. Word embeddings are used to represent text inputs, allowing the model to capture semantic meaning. Training is performed using Sparse Categorical Crossentropy loss, suitable for handling multi-class outputs over large vocabularies.

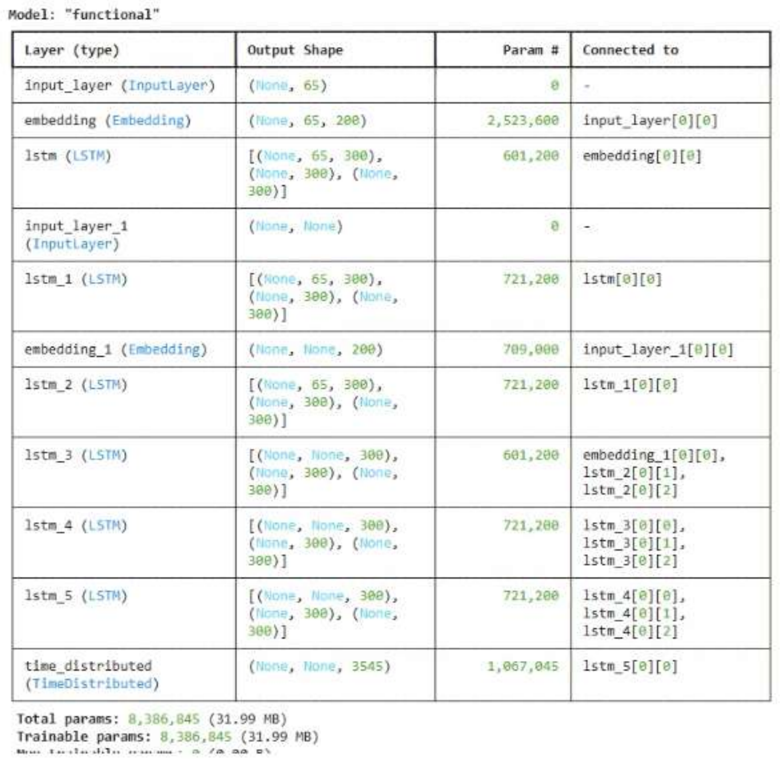
Model Training and Evaluation

The model training process involves data splitting. We split data into training, validation, and testing sets. Hyperparameter tuning optimizes model performance. Examples of hyperparameters include learning rate and batch size. Evaluation metrics assess the model's performance. ROUGE scores evaluate summarization quality.





```python
# Embedding layer for encoder
enc_emb = Embedding(x_voc, embedding_dim, trainable=True)(encoder_inputs)

# 3 LSTM layers for the encoder
encoder_lstm1 = LSTM(latent_dim, return_sequences=True, return_state=True, dropout=0.4, recurrent_dropout=0.4)
encoder_output1, state_h1, state_c1 = encoder_lstm1(enc_emb)

encoder_lstm2 = LSTM(latent_dim, return_sequences=True, return_state=True, dropout=0.4, recurrent_dropout=0.4)
encoder_output2, state_h2, state_c2 = encoder_lstm2(encoder_output1)

encoder_lstm3 = LSTM(latent_dim, return_sequences=True, return_state=True, dropout=0.4, recurrent_dropout=0.4)
encoder_outputs, state_h3, state_c3 = encoder_lstm3(encoder_output2)

# Decoder
decoder_inputs = Input(shape=(None,))

# Embedding layer for decoder
dec_emb_layer = Embedding(y_voc, embedding_dim, trainable=True)
dec_emb = dec_emb_layer(decoder_inputs)

# 3 LSTM layers for the decoder (matching encoder structure)
decoder_lstm1 = LSTM(latent_dim, return_sequences=True, return_state=True, dropout=0.4, recurrent_dropout=0.2)
decoder_output1, decoder_state_h1, decoder_state_c1 = decoder_lstm1(dec_emb, initial_state=[state_h3, state_c3])

decoder_lstm2 = LSTM(latent_dim, return_sequences=True, return_state=True, dropout=0.4, recurrent_dropout=0.2)
decoder_output2, decoder_state_h2, decoder_state_c2 = decoder_lstm2(decoder_output1, initial_state=[decoder_state_h1, decoder_state_c1])

decoder_lstm3 = LSTM(latent_dim, return_sequences=True, return_state=True, dropout=0.4, recurrent_dropout=0.2)
decoder_outputs, decoder_state_h3, decoder_state_c3 = decoder_lstm3(decoder_output2, initial_state=[decoder_state_h2, decoder_state_c2])

# Dense softmax layer to generate final outputs
decoder_dense = TimeDistributed(Dense(y_voc, activation='softmax'))
decoder_outputs = decoder_dense(decoder_outputs)

# Define the model
model = Model([encoder_inputs, decoder_inputs], decoder_outputs)

model.summary()
```



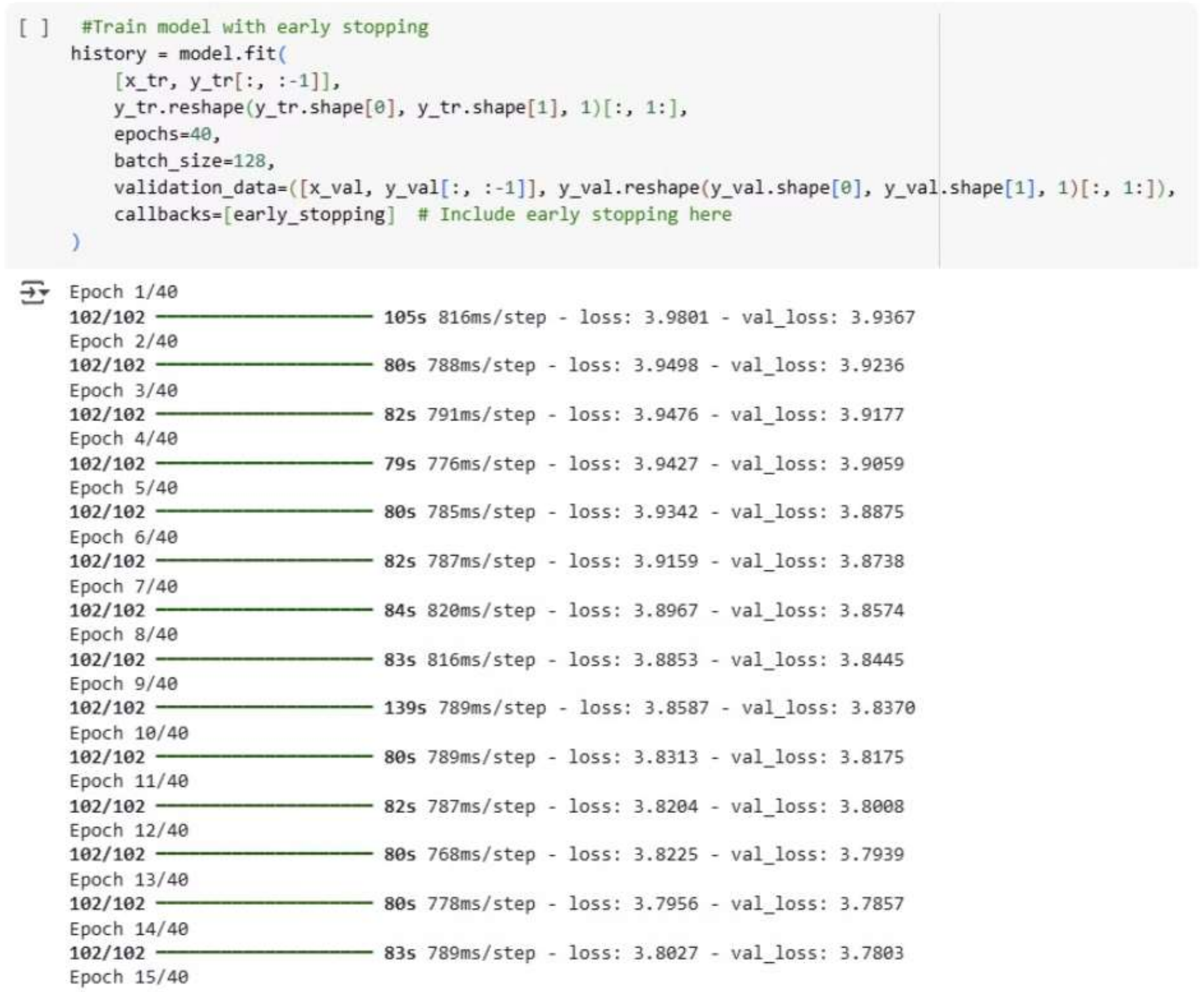
Model: "functional"

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 65)	0	-
embedding (Embedding)	(None, 65, 200)	2,523,600	input_layer[0][0]
lstm (LSTM)	[(None, 65, 300), (None, 300), (None, 300)]	601,200	embedding[0][0]
input_layer_1 (InputLayer)	(None, None)	0	-
lstm_1 (LSTM)	[(None, 65, 300), (None, 300), (None, 300)]	721,200	lstm[0][0]
embedding_1 (Embedding)	(None, None, 200)	709,000	input_layer_1[0][0]
lstm_2 (LSTM)	[(None, 65, 300), (None, 300), (None, 300)]	721,200	lstm_1[0][0]
lstm_3 (LSTM)	[(None, None, 300), (None, 300), (None, 300)]	601,200	embedding_1[0][0], lstm_2[0][1], lstm_2[0][2]
lstm_4 (LSTM)	[(None, None, 300), (None, 300), (None, 300)]	721,200	lstm_3[0][0], lstm_3[0][1], lstm_3[0][2]
lstm_5 (LSTM)	[(None, None, 300), (None, 300), (None, 300)]	721,200	lstm_4[0][0], lstm_4[0][1], lstm_4[0][2]
time_distributed (TimeDistributed)	(None, None, 3545)	1,067,045	lstm_5[0][0]

```
Total params: 8,386,845 (31.99 MB)
Trainable params: 8,386,845 (31.99 MB)
```

MODEL SUMMARY

EPOCHS TRAINING



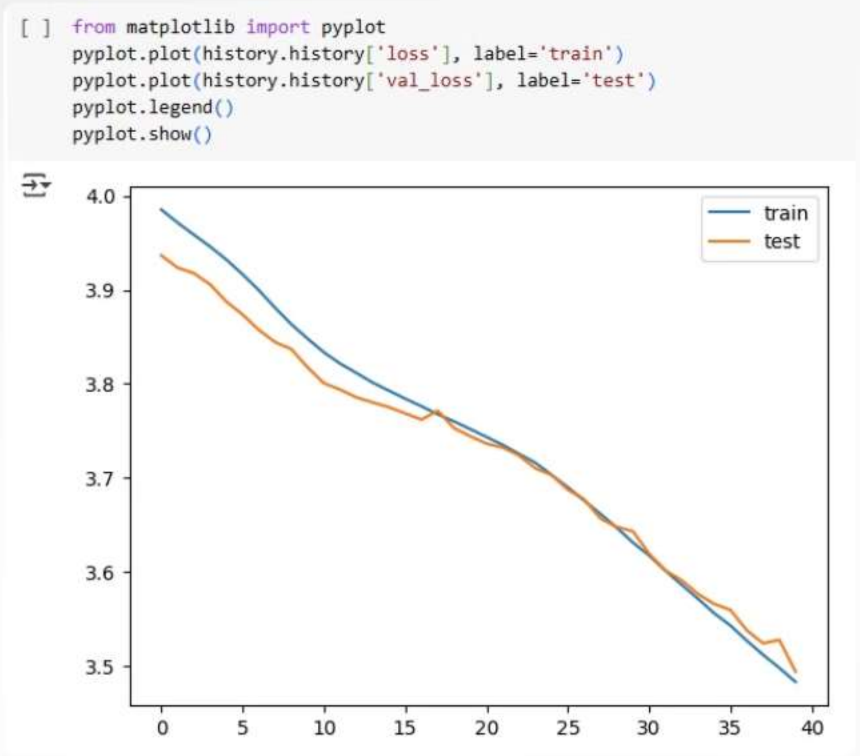
```
#Train model with early stopping
history = model.fit(
    [x_tr, y_tr[:, :-1]],
    y_tr.reshape(y_tr.shape[0], y_tr.shape[1], 1)[:, 1:],
    epochs=40,
    batch_size=128,
    validation_data=([x_val, y_val[:, :-1]], y_val.reshape(y_val.shape[0], y_val.shape[1], 1)[:, 1:]),
    callbacks=[early_stopping]  # Include early stopping here
)
```

```
Epoch 1/40
102/102 ━━━━━━━━━━━━━━━━━━━━ 105s 816ms/step - loss: 3.9801 - val_loss: 3.9367
Epoch 2/40
102/102 ━━━━━━━━━━━━━━━━━━━━ 80s 788ms/step - loss: 3.9498 - val_loss: 3.9236
Epoch 3/40
102/102 ━━━━━━━━━━━━━━━━━━━━ 82s 791ms/step - loss: 3.9476 - val_loss: 3.9177
Epoch 4/40
102/102 ━━━━━━━━━━━━━━━━━━━━ 79s 776ms/step - loss: 3.9427 - val_loss: 3.9059
Epoch 5/40
102/102 ━━━━━━━━━━━━━━━━━━━━ 80s 785ms/step - loss: 3.9342 - val_loss: 3.8875
Epoch 6/40
102/102 ━━━━━━━━━━━━━━━━━━━━ 82s 787ms/step - loss: 3.9159 - val_loss: 3.8738
Epoch 7/40
102/102 ━━━━━━━━━━━━━━━━━━━━ 84s 820ms/step - loss: 3.8967 - val_loss: 3.8574
Epoch 8/40
102/102 ━━━━━━━━━━━━━━━━━━━━ 83s 816ms/step - loss: 3.8853 - val_loss: 3.8445
Epoch 9/40
102/102 ━━━━━━━━━━━━━━━━━━━━ 139s 789ms/step - loss: 3.8587 - val_loss: 3.8370
Epoch 10/40
102/102 ━━━━━━━━━━━━━━━━━━━━ 80s 789ms/step - loss: 3.8313 - val_loss: 3.8175
Epoch 11/40
102/102 ━━━━━━━━━━━━━━━━━━━━ 82s 787ms/step - loss: 3.8204 - val_loss: 3.8008
Epoch 12/40
102/102 ━━━━━━━━━━━━━━━━━━━━ 80s 768ms/step - loss: 3.8225 - val_loss: 3.7939
Epoch 13/40
102/102 ━━━━━━━━━━━━━━━━━━━━ 80s 778ms/step - loss: 3.7956 - val_loss: 3.7857
Epoch 14/40
102/102 ━━━━━━━━━━━━━━━━━━━━ 83s 789ms/step - loss: 3.8027 - val_loss: 3.7803
Epoch 15/40
```

Results and Discussion

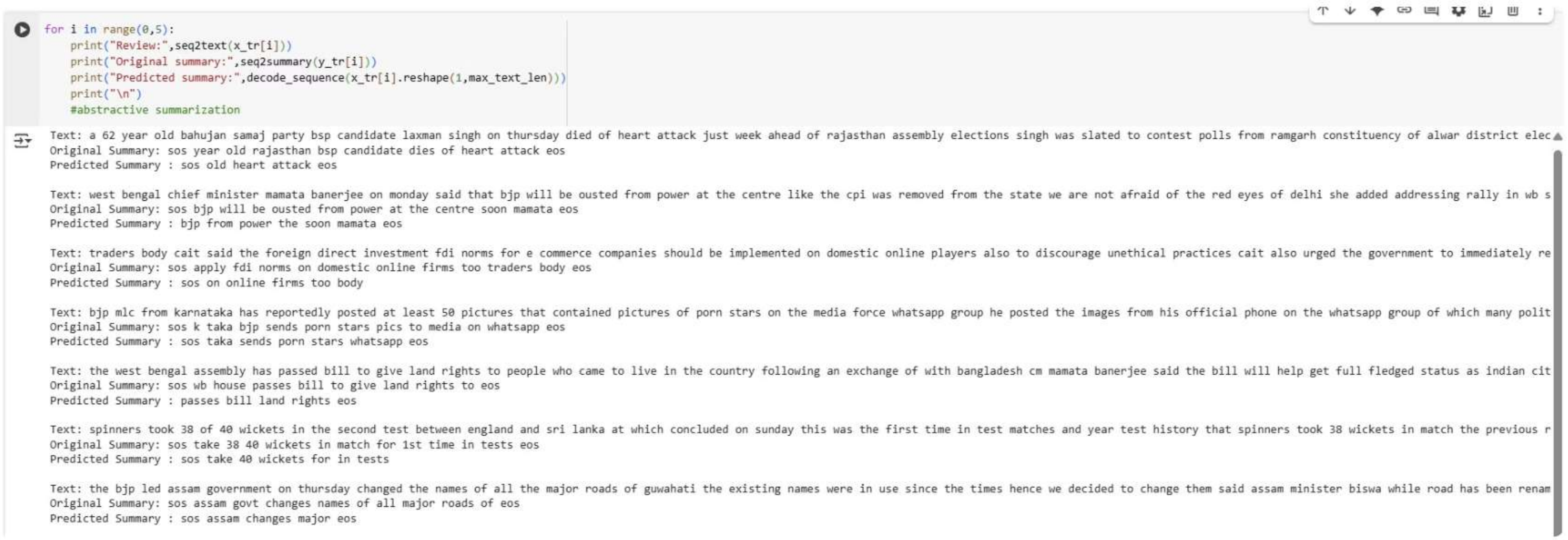
The project produces text summarization. We display original articles and AI summaries. Strengths include automation of complex tasks. Limitations may involve nuanced context understanding. Errors stem from noisy data or model biases.

Image shows the loss reducing for both train and test data and ultimately converging.



```
[ ] from matplotlib import pyplot
    pyplot.plot(history.history['loss'], label='train')
    pyplot.plot(history.history['val_loss'], label='test')
    pyplot.legend()
    pyplot.show()
```

Comparison of Original vs Predicted summaries

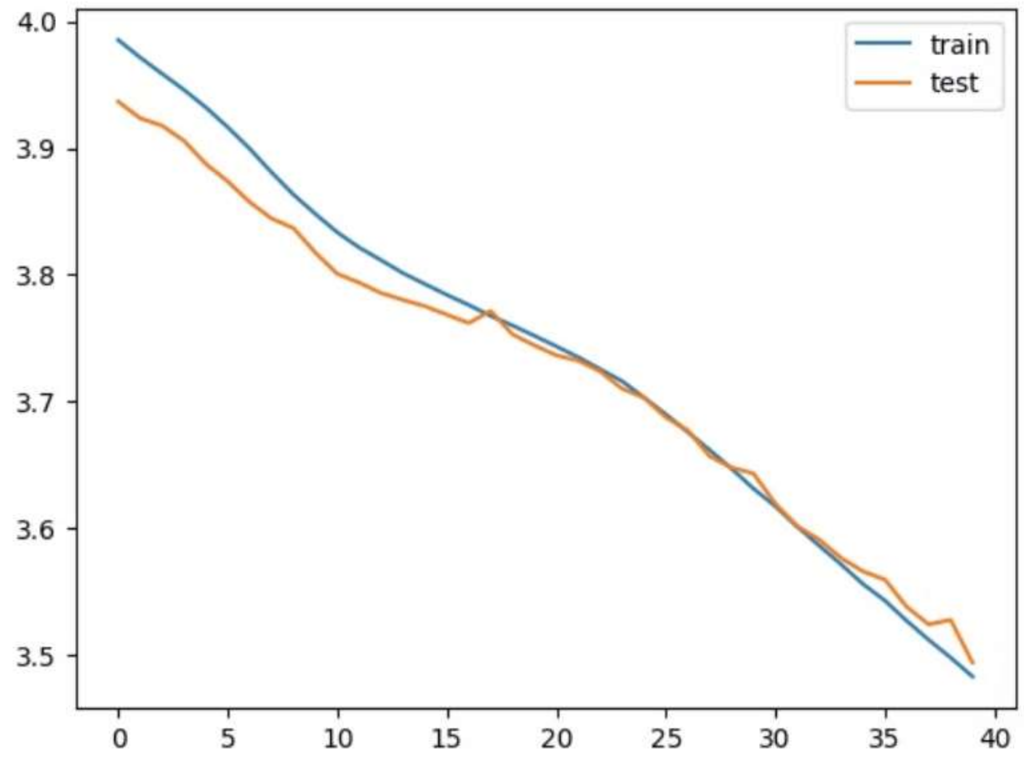


```python
for i in range(0,5):
    print("Review:",seq2text(x_tr[i]))
    print("Original summary:",seq2summary(y_tr[i]))
    print("Predicted summary:",decode_sequence(x_tr[i].reshape(1,max_text_len)))
    print("\n")
    #abstractive summarization
```

```
Text: a 62 year old bahujan samaj party bsp candidate laxman singh on thursday died of heart attack just week ahead of rajasthan assembly elections singh was slated to contest polls from ramgarh constituency of alwar district elec
Original Summary: sos year old rajasthan bsp candidate dies of heart attack eos
Predicted Summary : sos old heart attack eos

Text: west bengal chief minister mamata banerjee on monday said that bjp will be ousted from power at the centre like the cpi was removed from the state we are not afraid of the red eyes of delhi she added addressing rally in wb s
Original Summary: sos bjp will be ousted from power at the centre soon mamata eos
Predicted Summary : bjp from power the soon mamata eos

Text: traders body cait said the foreign direct investment fdi norms for e commerce companies should be implemented on domestic online players also to discourage unethical practices cait also urged the government to immediately re
Original Summary: sos apply fdi norms on domestic online firms too traders body eos
Predicted Summary : sos on online firms too body

Text: bjp mlc from karnataka has reportedly posted at least 50 pictures that contained pictures of porn stars on the media force whatsapp group he posted the images from his official phone on the whatsapp group of which many polit
Original Summary: sos k taka bjp sends porn stars pics to media on whatsapp eos
Predicted Summary : sos taka sends porn stars whatsapp eos

Text: the west bengal assembly has passed bill to give land rights to people who came to live in the country following an exchange of with bangladesh cm mamata banerjee said the bill will help get full fledged status as indian cit
Original Summary: sos wb house passes bill to give land rights to eos
Predicted Summary : passes bill land rights eos

Text: spinners took 38 of 40 wickets in the second test between england and sri lanka at which concluded on sunday this was the first time in test matches and year test history that spinners took 38 wickets in match the previous r
Original Summary: sos take 38 40 wickets in match for 1st time in tests eos
Predicted Summary : sos take 40 wickets for in tests

Text: the bjp led assam government on thursday changed the names of all the major roads of guwahati the existing names were in use since the times hence we decided to change them said assam minister biswa while road has been renam
Original Summary: sos assam govt changes names of all major roads of eos
Predicted Summary : sos assam changes major eos
```

ORIGINAL RESULTS WITH 3 LSTMS

```
Total params: 8,386,845 (31.99 MB)
Trainable params: 8,386,845 (31.99 MB)
Non-trainable params: 0 (0.00 B)
```



```
Text: a 62 year old bahujan samaj party bsp candidate laxman singh on thursday died of heart attack just week ahea
Original Summary: sos year old rajasthan bsp candidate dies of heart attack eos
Predicted Summary : sos old heart attack eos

Text: west bengal chief minister mamata banerjee on monday said that bjp will be ousted from power at the centre
Original Summary: sos bjp will be ousted from power at the centre soon mamata eos
Predicted Summary : bjp from power the soon mamata eos

Text: traders body cait said the foreign direct investment fdi norms for e commerce companies should be implemente
Original Summary: sos apply fdi norms on domestic online firms too traders body eos
Predicted Summary : sos on online firms too body
```

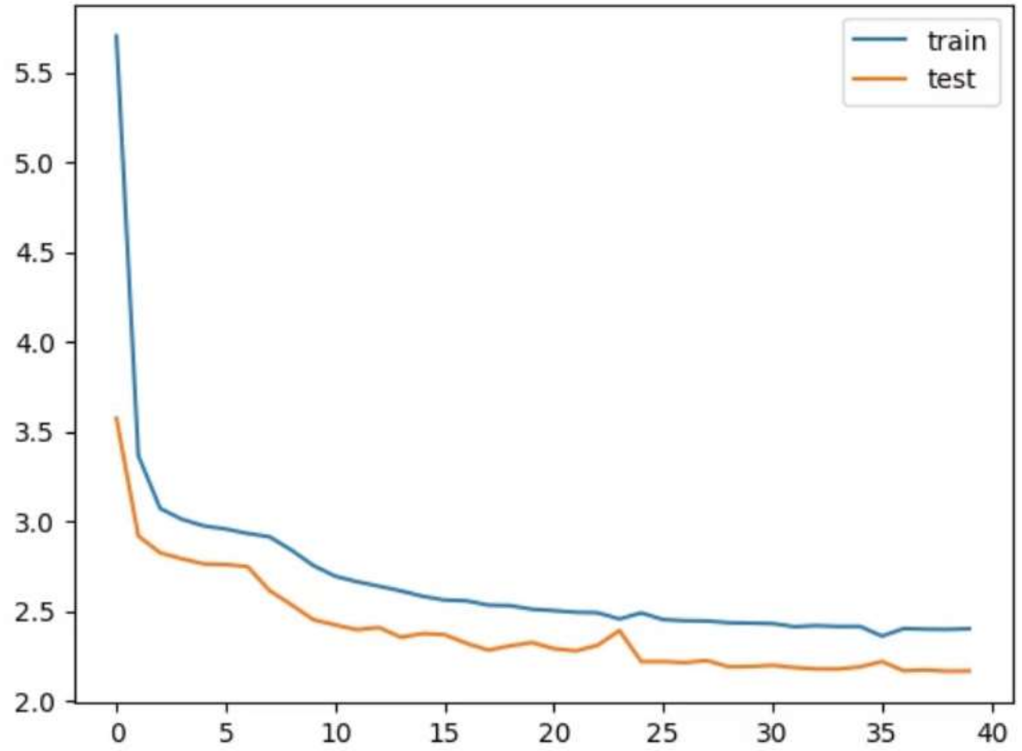
THE ORIGINAL PROJECT- has dataset of 10,500 rows and is trained for 40 epochs.

The training and validation loss reduce and there is no overfitting or underfitting.

The predicted summaries although not accurate show abstractive summarization.

REDUCED DATASET AND 3 LSTMS

```
Total params: 5,311,710 (20.26 MB)
Trainable params: 5,311,710 (20.26 MB)
Non-trainable params: 0 (0.00 B)
```



```
Review: shiv sena chief uddhav thackeray has slammed prime minister narendra modi for power and taking away the independence
Original summary: sos sena slams pm modi for power eos
```

```
Predicted summary: to to to to to to to to to to to to to eos to
```

THE REDUCED DATASET and 3 LSTMS- has 1000 rows for training due to computational constraints.

Both losses decrease steadily, with a sharp drop in the initial epochs, indicating that the model quickly learns basic patterns. As training progresses, the losses converge and stabilize, showing that the model is improving without overfitting. Notably, the test loss remains consistently lower than the training loss, suggesting strong generalization and effective learning from the data.

Reasons:

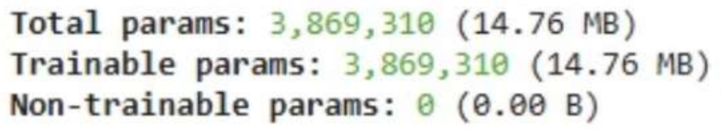
1. Insufficient or Poorly Learned Vocabulary

2. Unbalanced or Undertrained Model

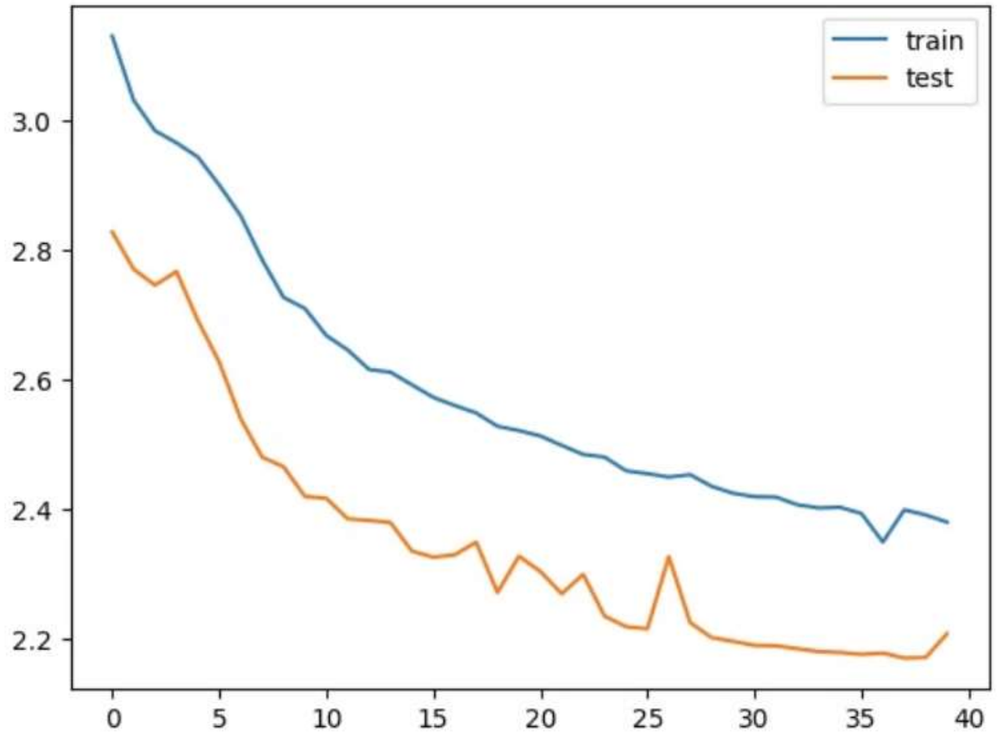
3. Improper Tokenization or Embedding

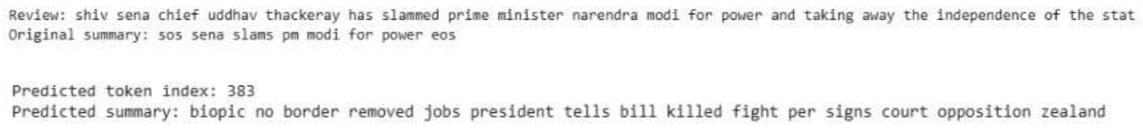
4. Vanishing Gradients in LSTM

REDUCES DATASET AND 2 LSTMS



```
Total params: 3,869,310 (14.76 MB)
Trainable params: 3,869,310 (14.76 MB)
Non-trainable params: 0 (0.00 B)
```



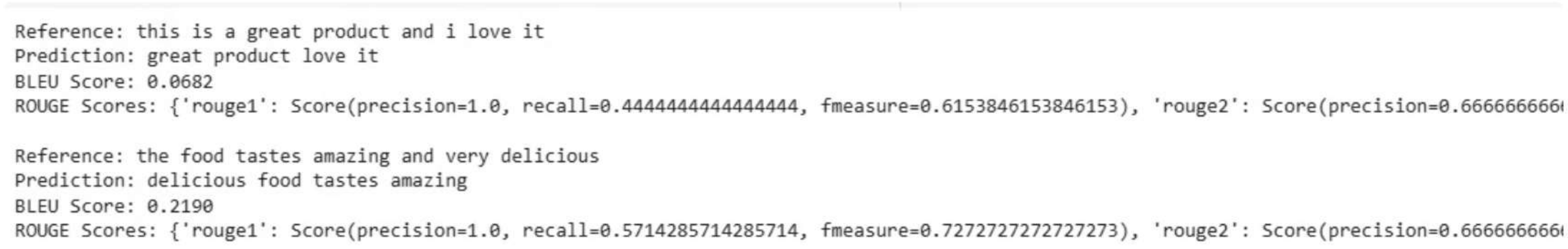


```
Review: shiv sena chief uddhav thackeray has slammed prime minister narendra modi for power and taking away the independence of the stat
Original summary: sos sena slams pm modi for power eos

 Predicted token index: 383
 Predicted summary: biopic no border removed jobs president tells bill killed fight per signs court opposition zealand
```

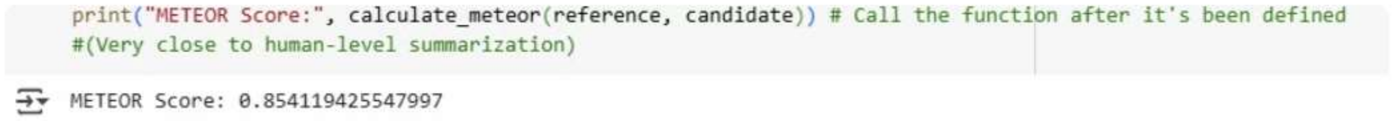
THE REDUCED DATASET AND 2 LSTMS- The graph shows a steady decrease in both training and testing loss over 40 epochs, indicating that the model is learning effectively. The narrowing gap between the two curves suggests good generalization without overfitting. While there are slight fluctuations in the test loss, the overall trend remains downward and stable. Toward the end, both losses flatten out, showing that the model is approaching convergence. Overall, this reflects a healthy and well-balanced training process.

The summary generated appears as a sequence of unrelated keywords, indicating that the model has not learned to form coherent or meaningful sentences. This behavior may be due to insufficient training, or issues in tokenization and sequence padding. It suggests the model is capturing word frequency patterns rather than actual context or grammar, leading to disjointed outputs instead of logical summaries.

Evaluation Metrics

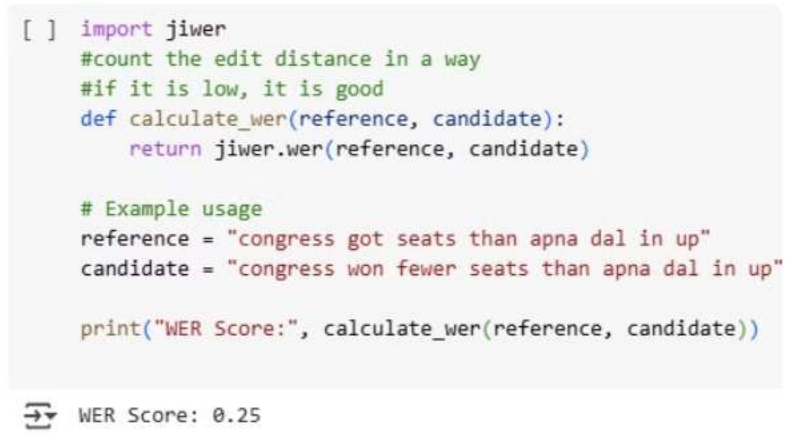


```
Reference: this is a great product and i love it
Prediction: great product love it
BLEU Score: 0.0682
ROUGE Scores: {'rouge1': Score(precision=1.0, recall=0.4444444444444444, fmeasure=0.6153846153846153), 'rouge2': Score(precision=0.666666666

Reference: the food tastes amazing and very delicious
Prediction: delicious food tastes amazing
BLEU Score: 0.2190
ROUGE Scores: {'rouge1': Score(precision=1.0, recall=0.5714285714285714, fmeasure=0.7272727272727273), 'rouge2': Score(precision=0.666666666
```



```python
print("METEOR Score:", calculate_meteor(reference, candidate)) # Call the function after it's been defined
#(Very close to human-level summarization)
```

```
METEOR Score: 0.854119425547997
```



```python
import jiwer
#count the edit distance in a way
#if it is low, it is good
def calculate_wer(reference, candidate):
    return jiwer.wer(reference, candidate)

# Example usage
reference = "congress got seats than apna dal in up"
candidate = "congress won fewer seats than apna dal in up"

print("WER Score:", calculate_wer(reference, candidate))
```

```
WER Score: 0.25
```

COMPARISION OF EVALUATION METRICS

Metric	Original Dataset (3 LSTMs)	Reduced Dataset (3 LSTMs)	Reduced Dataset (2 LSTMs)
BLEU Score	0.0517	0.0682	0.0682
ROUGE-1 F1	0.5455	0.6154	0.6154
ROUGE-2 F1	0.4	0.3636	0.3636
ROUGE-L F1	0.5455	0.6154	0.6154
METEOR	0.8541	0.8541	0.8541
WER	0.25	0.25	0.25
Dataset	20k samples	1k samples	1k samples
Model Layers	3 LSTMs	3 LSTMs	2 LSTMs

Interpretation of Results:

- **BLEU Score:** The BLEU score is slightly better for both reduced models compared to the full dataset model. This could be because the reduced dataset contains less variation, making it easier for the model to memorize and generate closer predictions on a small test set. However, BLEU alone isn't sufficient to assess semantic quality in small summaries.
- **ROUGE Scores (1, 2, L):** These scores are higher for the reduced dataset with 3 LSTMs, especially for ROUGE-1 and ROUGE-L. This suggests that the predicted summaries in the reduced dataset align better word-for-word and structurally with the references. ROUGE-2, which captures bigrams, is slightly higher in the original model, possibly indicating more varied phrasing.
- **METEOR:** Interestingly, all models achieved the **same METEOR score** of 0.8541. METEOR focuses on semantic similarity, including synonyms and word order, and this consistency implies that all models captured similar semantic meanings, even if phrasing differed.
- **WER (Word Error Rate):** This also remained constant at **0.25** for all models, meaning 25% of the words were either inserted, deleted, or substituted — not much difference in overall error across models.
- **Conclusion:**
  - The **reduced dataset with 3 LSTMs** gives the **best overall ROUGE and BLEU** performance, likely due to overfitting on a smaller, more uniform dataset.
  - The **2 LSTM model** performs nearly identically, which shows that **removing one LSTM layer** didn't reduce performance. In fact, it may be more efficient with fewer parameters and faster training/inference.
  - The **original dataset model** might generalize better on a broader dataset but scores slightly lower on exact match metrics due to more diversity in training data.

Challenges & Learnings

Challenges: Noisy data, long training

Learnings: NLP pipeline, LSTM modeling

Conclusion

This architecture outlines a structured approach to neural text summarization using an LSTM-based Seq2Seq model. Each stage — from raw data preprocessing to summary generation — is critical. The visualization of the LSTM cell highlights how key components like the Forget Gate, Input Gate, and Output Gate help retain contextual information across time steps. Together, the encoder-decoder setup and LSTM's memory handling enable the generation of coherent and context-aware summaries, forming a strong foundation for further enhancements like attention mechanisms or transformer models.