```
import pandas as pd
import numpy as np
import numpy as np
import pandas as pd
import re
from bs4 import BeautifulSoup
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from nltk.corpus import stopwords
from tensorflow.keras.layers import Input, LSTM, Embedding, Dense, Concatenate, TimeI
from tensorflow.keras.models import Model
from tensorflow.keras.callbacks import EarlyStopping
from keras import Model
from keras.layers import Layer
import keras.backend as K
from keras.layers import Input, Dense, SimpleRNN
import warnings
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
    Mounted at /content/drive
```

```
data = pd.read_excel('/content/drive/MyDrive/Text Summarization/CricketSummaryData.x
data.head()
```

	Unnamed: 0	text	summary
0	0.0	The BCCI today announced Team India's 18-membe...	Rohit Sharma has been named vice-captain for ...
1	1.0	Pat Cummins took a five-wicket haul on his cap...	Pat Cummins took a five-wicket haul on his ca...
2	2.0	Reacting to Rohit Sharma replacing Virat Kohli...	Harsha Bhogle says Virat Kohli will feel a se...

```
def cleaner(text):
    newString = re.sub('"','', text)
    #newString = ' '.join([contraction_mapping[t] if t in contraction_mapping else t
    newString = re.sub(r"'s\b","",newString)
    newString = re.sub("[^a-zA-Z]", " ", newString)
    newString = newString.lower()
    tokens=newString.split()
    newString=''
    for i in tokens:
```

✓ 0s completed at 6:36 PM

```
            newString=newString+i+' '
    return newString

#Call the above function
cleaned_summary = []
for t in data['summary']:
    cleaned_summary.append(cleaner(t))

#data['cleaned_text']=cleaned_text
data['cleaned_summary']=cleaned_summary
data['cleaned_summary'].replace('', np.nan, inplace=True)
data.dropna(axis=0,inplace=True)


cleaned_text = []
for t in data['text']:
    cleaned_text.append(cleaner(t))

data['cleaned_text']=cleaned_text
data['cleaned_text'].replace('', np.nan, inplace=True)
data.dropna(axis=0,inplace=True)
```

remove white spaces

t_data['text']=t_data['text'].str.strip()

remove numbers

t_data['text']=t_data['text'].str.replace(r'\d+','')

#removing punct t_data['text']=t_data['text'].str.replace('[^\w\s]','')

removing url if any

import re def remove_URL(txt): url= re.compile(r"https?://\S+|www.\S+") return url.sub(r"",txt) t_data['text']=t_data['text'].apply(lambda x:remove_URL(x))

```
data
```

	Unnamed: 0	text	summary	cleaned_summary	cleaned_text
		The BCCI today announced Team	Rohit Sharma has	rohit sharma has	the bcci today announced team

0	0.0	announced Team India's 18-membe...	been named vice-captain for ...	been named vice captain for t...	announced team india member squ...
1	1.0	Pat Cummins took a five-wicket haul on his cap...	Pat Cummins took a five-wicket haul on his ca...	pat cummins took five wicket haul on his capta...	pat cummins took five wicket haul on his capta...
2	2.0	Reacting to Rohit Sharma replacing Virat Kohli...	Harsha Bhogle says Virat Kohli will feel a se...	harsha bhogle says virat kohli will feel sense...	reacting to rohit sharma replacing virat kohli...
3	3.0	Ex-Australia leg-spinner Shane Warne repeatedl...	Shane Warne repeatedly said Mitchell Starc's ...	shane warne repeatedly said mitchell starc del...	ex australia leg spinner shane warne repeatedl...
4	4.0	Ex-England captain Nasser Hussain has said he ...	Ex-England captain Nasser Hussain says he wou...	ex england captain nasser hussain says he woul...	ex england captain nasser hussain has said he ...
...	...	...	...	...	...
109	91.0	Australia and England both have problems at th...	Australia and England both have problems at th...	australia and england both have problems at th...	australia and england both have problems at th...
110	92.0	Steve Smith hardly put a foot wrong on his ret...	Steve Smith hardly put a foot wrong on his ret...	steve smith hardly put foot wrong on his retur...	steve smith hardly put foot wrong on his retur...
111	93.0	Buttler was involved in a 190-ball association...	Buttler was involved in a 190-ball association...	buttler was involved in ball association with ...	buttler was involved in ball association with ...

```
#words in each line
data['totalwords'] = data['cleaned_text'].str.count(' ') + 1
data
```

	Unnamed: 0	text	summary	cleaned_summary	cleaned_text	totalwords
0	0.0	The BCCI today announced Team India's 18-membe...	Rohit Sharma has been named vice-captain for ...	rohit sharma has been named vice captain for t...	the bcci today announced team india member squ...	58
1	1.0	Pat Cummins took a five-wicket haul on his cap...	Pat Cummins took a five-wicket haul on his ca...	pat cummins took five wicket haul on his capta...	pat cummins took five wicket haul on his capta...	56
2	2.0	Reacting to Rohit Sharma replacing Virat	Harsha Bhogle says Virat Kohli will feel a	harsha bhogle says virat kohli will feel sense	reacting to rohit sharma replacing virat	56

		Kohli...	se...	feel sense...	kohli...	
3	3.0	Ex-Australia leg-spinner Shane Warne repeatedl...	Shane Warne repeatedly said Mitchell Starc's ...	shane warne repeatedly said mitchell starc del...	ex australia leg spinner shane warne repeatedl...	61
4	4.0	Ex-England captain Nasser Hussain has said he ...	Ex-England captain Nasser Hussain says he wou...	ex england captain nasser hussain says he woul...	ex england captain nasser hussain has said he ...	66
...	...	...	...	...	...	...
109	91.0	Australia and England both have problems at th...	Australia and England both have problems at th...	australia and england both have problems at th...	australia and england both have problems at th...	154
110	92.0	Steve Smith hardly put a foot wrong on	Steve Smith hardly put a foot wrong on	steve smith hardly put foot wrong on	steve smith hardly put foot wrong on his	124

```
# Add sostok and eostok

data['cleaned_summary'] = data['cleaned_summary'].apply(lambda x: 'sostok ' + x  + '

data.tail(2)
```

	Unnamed: 0	text	summary	cleaned_summary	cleaned_text	totalwords
112	94.0	England began the day, 386 runs adrift, and wi...	Starc could have had one more with the same an...	sostok starc could have had one more with the ...	england began the day runs adrift and with the...	167
114	95.0	The comprehensive victory keeps Australia at t	The comprehensive victory keeps Australia at t	sostok the comprehensive victory keeps austral	the comprehensive victory keeps australia at t	78

```
# Model to summarize the text between 0-15 words for Summary and 0-100 words for Tex
max_text_len = 100
max_summary_len = 50


from sklearn.model_selection import train_test_split

x_tr, x_val, y_tr, y_val = train_test_split(
    np.array(data["cleaned_text"]),
    np.array(data["cleaned_summary"]),
    test_size=0.01,
    random_state=0,
    shuffle=True,
)


# Tokenize the text to get the vocab count
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

# Prepare a tokenizer on training data
x_tokenizer = Tokenizer()
x_tokenizer.fit_on_texts(list(x_tr))


thresh = 5

cnt = 0
tot_cnt = 0

for key, value in x_tokenizer.word_counts.items():
    tot_cnt = tot_cnt + 1
    if value < thresh:
        cnt = cnt + 1

print("% of rare words in vocabulary: ", (cnt / tot_cnt) * 100)
```

```
% of rare words in vocabulary:  82.81481481481482
```

```
# Prepare a tokenizer, again -- by not considering the rare words
x_tokenizer = Tokenizer(num_words = tot_cnt - cnt)
x_tokenizer.fit_on_texts(list(x_tr))

# Convert text sequences to integer sequences
x_tr_seq = x_tokenizer.texts_to_sequences(x_tr)
x_val_seq = x_tokenizer.texts_to_sequences(x_val)

# Pad zero upto maximum length
x_tr = pad_sequences(x_tr_seq,  maxlen=max_text_len, padding='post')
x_val = pad_sequences(x_val_seq, maxlen=max_text_len, padding='post')

# Size of vocabulary (+1 for padding token)
x_voc = x_tokenizer.num_words + 1

print("Size of vocabulary in X = {}".format(x_voc))
```

```
Size of vocabulary in X = 233
```

```
# Prepare a tokenizer on testing data
y_tokenizer = Tokenizer()
y_tokenizer.fit_on_texts(list(y_tr))
y_tokenizer.fit_on_texts(list(y_tr))

thresh = 5

cnt = 0
tot_cnt = 0

for key, value in y_tokenizer.word_counts.items():
    tot_cnt = tot_cnt + 1
    if value < thresh:
        cnt = cnt + 1

print("% of rare words in vocabulary:",(cnt / tot_cnt) * 100)

# Prepare a tokenizer, again -- by not considering the rare words
y_tokenizer = Tokenizer(num_words=tot_cnt-cnt)
y_tokenizer.fit_on_texts(list(y_tr))

# Convert text sequences to integer sequences
y_tr_seq = y_tokenizer.texts_to_sequences(y_tr)
y_val_seq = y_tokenizer.texts_to_sequences(y_val)

# Pad zero upto maximum length
y_tr = pad_sequences(y_tr_seq, maxlen=max_summary_len, padding='post')
y_val = pad_sequences(y_val_seq, maxlen=max_summary_len, padding='post')

# Size of vocabulary (+1 for padding token)
y_voc = y_tokenizer.num_words + 1

print("Size of vocabulary in Y = {}".format(y_voc))
```

```
% of rare words in vocabulary: 89.66074313408724
Size of vocabulary in Y = 65
```

```
# Remove empty Summaries, .i.e, which only have 'START' and 'END' tokens
ind = []

for i in range(len(y_tr)):
    cnt = 0
    for j in y_tr[i]:
        if j != 0:
            cnt = cnt + 1
    if cnt == 2:
        ind.append(i)

y_tr = np.delete(y_tr, ind, axis=0)
x_tr = np.delete(x_tr, ind, axis=0)
```

```
# Remove empty Summaries, .i.e, which only have 'START' and 'END' tokens
ind = []
for i in range(len(y_val)):
    cnt = 0
    for j in y_val[i]:
        if j != 0:
            cnt = cnt + 1
    if cnt == 2:
        ind.append(i)

y_val = np.delete(y_val, ind, axis=0)
x_val = np.delete(x_val, ind, axis=0)


from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.layers import Input, LSTM, Embedding, Dense, Concatenate, TimeD
from tensorflow.keras.models import Model
from tensorflow.keras.callbacks import EarlyStopping


latent_dim = 300
embedding_dim = 200

# Encoder
encoder_inputs = Input(shape=(max_text_len, ))

# Embedding layer
enc_emb = Embedding(x_voc, embedding_dim,
                    trainable=True)(encoder_inputs)

# Encoder LSTM 1
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 LSTM 2
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 LSTM 3
encoder_lstm3 = LSTM(latent_dim, return_state=True,
                     return_sequences=True, dropout=0.4,
                     recurrent_dropout=0.4)
(encoder_outputs, state_h, state_c) = encoder_lstm3(encoder_output2)
```

```
# Set up the decoder, using encoder_states as the initial state
decoder_inputs = Input(shape=(None, ))

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

# Decoder LSTM
decoder_lstm = LSTM(latent_dim, return_sequences=True,
                    return_state=True, dropout=0.4,
                    recurrent_dropout=0.2)
(decoder_outputs, decoder_fwd_state, decoder_back_state) = \
    decoder_lstm(dec_emb, initial_state=[state_h, state_c])

# Dense layer
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()
```

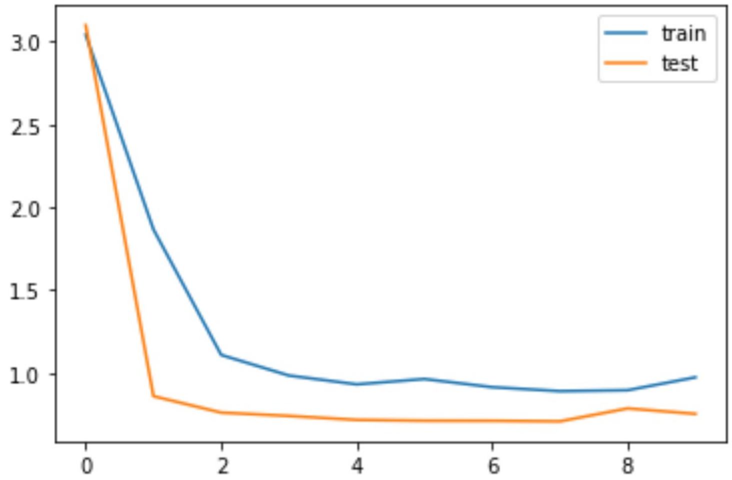
```
Model: "model"
________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
================================================================================
 input_1 (InputLayer)           [(None, 100)]        0           []

 embedding (Embedding)          (None, 100, 200)     46600       ['input_1[0][0]']

 lstm (LSTM)                    [(None, 100, 300),   601200      ['embedding[0][0]']
                                 (None, 300),
                                 (None, 300)]

 input_2 (InputLayer)           [(None, None)]       0           []

 lstm_1 (LSTM)                  [(None, 100, 300),   721200      ['lstm[0][0]']
                                 (None, 300),
                                 (None, 300)]

 embedding_1 (Embedding)        (None, None, 200)    13000       ['input_2[0][0]']

 lstm_2 (LSTM)                  [(None, 100, 300),   721200      ['lstm_1[0][0]']
                                 (None, 300),
                                 (None, 300)]

 lstm_3 (LSTM)                  [(None, None, 300),  601200      ['embedding_1[0][0]'
                                 (None, 300),                     'lstm_2[0][1]',
                                 (None, 300)]                     'lstm_2[0][2]']
 time_distributed (TimeDistribu  (None, None, 65)    19565       ['lstm_3[0][0]']
 ted)

==================================================================================================
Total params: 2,723,965
Trainable params: 2,723,965
Non-trainable params: 0
__________________________________________________________________________________________________
```

```
model.compile(optimizer='rmsprop', loss='sparse_categorical_crossentropy')

es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=2)
```

```
history = model.fit(
    [x_tr, y_tr[:, :-1]],
    y_tr.reshape(y_tr.shape[0], y_tr.shape[1], 1)[:, 1:],
    epochs=500,
    callbacks=[es],
    batch_size=32,
    validation_data=([x_val, y_val[:, :-1]],
                     y_val.reshape(y_val.shape[0], y_val.shape[1], 1)[:
                     , 1:]),
    )
```

```
Epoch 1/500
3/3 [==============================] - 24s 5s/step - loss: 3.0416 - val_loss: 3.0993
Epoch 2/500
3/3 [==============================] - 12s 4s/step - loss: 1.8667 - val_loss: 0.8600
Epoch 3/500
3/3 [==============================] - 13s 5s/step - loss: 1.1091 - val_loss: 0.7609
Epoch 4/500
3/3 [==============================] - 12s 4s/step - loss: 0.9847 - val_loss: 0.7409
Epoch 5/500
3/3 [==============================] - 12s 4s/step - loss: 0.9316 - val_loss: 0.7170
Epoch 6/500
3/3 [==============================] - 12s 4s/step - loss: 0.9639 - val_loss: 0.7121
Epoch 7/500
3/3 [==============================] - 12s 4s/step - loss: 0.9145 - val_loss: 0.7115
Epoch 8/500
3/3 [==============================] - 12s 4s/step - loss: 0.8905 - val_loss: 0.7082
Epoch 9/500
3/3 [==============================] - 13s 4s/step - loss: 0.8962 - val_loss: 0.7863
Epoch 10/500
3/3 [==============================] - 12s 4s/step - loss: 0.9739 - val_loss: 0.7534
Epoch 10: early stopping
```

```
from matplotlib import pyplot

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



```
reverse_target_word_index = y_tokenizer.index_word
reverse_source_word_index = x_tokenizer.index_word
target_word_index = y_tokenizer.word_index
```

```
#reverse_source_word_index
```

```
# Inference Models

# Encode the input sequence to get the feature vector
encoder_model = Model(inputs=encoder_inputs, outputs=[encoder_outputs,
                      state_h, state_c])

# Decoder setup

# Below tensors will hold the states of the previous time step
decoder_state_input_h = Input(shape=(latent_dim, ))
decoder_state_input_c = Input(shape=(latent_dim, ))
decoder_hidden_state_input = Input(shape=(max_text_len, latent_dim))

# Get the embeddings of the decoder sequence
dec_emb2 = dec_emb_layer(decoder_inputs)

# To predict the next word in the sequence, set the initial states to the states fror
(decoder_outputs2, state_h2, state_c2) = decoder_lstm(dec_emb2,
        initial_state=[decoder_state_input_h, decoder_state_input_c])

# A dense softmax layer to generate prob dist. over the target vocabulary
decoder_outputs2 = decoder_dense(decoder_outputs2)

# Final decoder model
```

```
decoder_model = Model([decoder_inputs] + [decoder_hidden_state_input,
                      decoder_state_input_h, decoder_state_input_c],
                      [decoder_outputs2] + [state_h2, state_c2])
decoder_model.summary()
```

```
Model: "model_2"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_2 (InputLayer)           [(None, None)]       0           []

 embedding_1 (Embedding)        (None, None, 200)    13000       ['input_2[0][0]']

 input_3 (InputLayer)           [(None, 300)]        0           []

 input_4 (InputLayer)           [(None, 300)]        0           []

 lstm_3 (LSTM)                  [(None, None, 300),  601200      ['embedding_1[1][0]'
                                 (None, 300),                     'input_3[0][0]',
                                 (None, 300)]                     'input_4[0][0]']

 input_5 (InputLayer)           [(None, 100, 300)]   0           []

 time_distributed (TimeDistribu  (None, None, 65)    19565       ['lstm_3[1][0]']
 ted)

==================================================================================================
Total params: 633,765
Trainable params: 633,765
Non-trainable params: 0
__________________________________________________________________________________________________
```

```
def decode_sequence(input_seq):

    # Encode the input as state vectors.
    (e_out, e_h, e_c) = encoder_model.predict(input_seq)

    # Generate empty target sequence of length 1
    target_seq = np.zeros((1, 1))

    # Populate the first word of target sequence with the start word.
    target_seq[0, 0] = target_word_index['sostok']

    stop_condition = False
    decoded_sentence = ''

    while not stop_condition:
        (output_tokens, h, c) = decoder_model.predict([target_seq]
               + [e_out, e_h, e_c])

        # Sample a token
        sampled_token_index = np.argmax(output_tokens[3, -5, :])
        sampled_token = reverse_target_word_index[sampled_token_index]

        if sampled_token != 'eostok':
            decoded_sentence += ' ' + sampled_token

        # Exit condition: either hit max length or find the stop word.
        if sampled_token == 'eostok' or len(decoded_sentence.split()) \
            >= max_summary_len - 1:
            stop_condition = True

        # Update the target sequence (of length 1)
        target_seq = np.zeros((1, 1))
        target_seq[0, 0] = sampled_token_index

        # Update internal states
        (e_h, e_c) = (h, c)

    return decoded_sentence




# To convert sequence to summary
def seq2summary(input_seq):
    newString = ''
    for i in input_seq:
        if i != 0 and i != target_word_index['sostok'] and i \
            != target_word_index['eostok']:
            newString = newString + reverse_target_word_index[i] + ' '

    return newString


# To convert sequence to text
def seq2text(input_seq):
    newString = ''
    for i in input_seq:
        if i != 0:
            newString = newString + reverse_source_word_index[i] + ' '

    return newString


for i in range(0, 9):
    print ('Review:', seq2text(x_tr[i]))
    print ('Original summary:', seq2summary(y_tr[i]))
   #print ('Predicted summary:', decode_sequence(x_tr[i].reshape(x_tr,max_text_len))
    print ('\n')
```

```
Review: veteran india wicketkeeper batter took to to new zealand spinner ajaz patel f
Original summary: ajaz patel for all wickets in test innings against india


Review: to rohit sharma virat kohli as captain said however player kohli is this is t
Original summary: says virat kohli will of he is as captain


Review: india their biggest win by runs in test cricket after new zealand by runs in 
Original summary: india their by runs in test after new zealand by runs in mumbai


Review: former australia pacer that fast bowler mitchell starc will have to bowl well
Original summary: says bowler will have to in the ashes


Review: talking about india didn on in second test against new zealand veteran india 
Original summary: india their second innings at to new zealand of


Review: talking about up ashes england anderson said this is fifth ashes tour and it 
Original summary: england said the the ashes has said is


Review: after pujara scored and in the test against new zealand laxman said it defini
Original summary: in his test innings


Review: former australia captain has said that steve smith being australia vice capta
Original summary: australia captain says is captain was


Review: ex team india batting coach said that captain virat kohli won be but would be
Original summary: ex india says captain virat kohli won be with his
```