PROJECT 2

OVERVIEW

Authored by Sahrudayi Caroline Parampogu (SXP230055) for NLP 6320 by Dr. Mazidi

This notebook describes my submission for project 2 the JESTMASTER - personalized joke assistant that specializes in delivering hilarious one-liners and joke punchlines tailored to the user's preference.

Report Link: https://docs.google.com/document/d/1-GmWSbolW0UrwkRw7LjB5Vt4vTrVgMyPX93tyZCDoeU/edit?usp=sharing

In [6]:

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

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

In [7]:

```
%cd /content/drive/MyDrive/NLP_PROJECT2//
```

```
/content/drive/MyDrive/NLP_PROJECT2
```

In [8]:

```
ls
```

```
astronomy.stackexchange.com_title_best_upvoted_answer.jsonl  jokes.gsheet
Conversation.csv                                             Models/
Conversation.gsheet                                          music.stackexchange.com.jsonl
Conversations_proof.gdoc                                     seq2seq_model1.keras
football.csv                                                 user_models/
jokes.csv
```

In [9]:

```
import numpy as np # linear algebra
import pandas as pd
```

In [10]:

```
df = pd.read_csv('jokes.csv')
```

```
df.head(10)
```

Out[10]:

	ID	Question	Answer
0	1	Did you hear about the Native American man tha...	He nearly drown in his own tea pee.
1	2	What's the best anti diarrheal prescription?	Mycheexarphlexin
2	3	What do you call a person who is outside a doo...	Matt
3	4	Which Star Trek character is a member of the m...	Jean-Luc Pickacard
4	5	What's the difference between a bullet and a h...	A bullet doesn't miss Harambe
5	6	Why was the Ethiopian baby crying?	He was having a mid-life crisis
6	7	What's the difference between a corn husker wi...	One shucks between fits...
7	8	Who is 2016's biggest sellout?	Kevin Durant or Bernie Sanders?
8	9	Why is little Annie's shoe floating in the sea?	Because the shark burped.
9	10	What's the difference between a married man an...	A bachelor will go to the fridge, sees nothing...

In [11]:

```
df.count()
```

Out[11]:

```
ID          38269
Question    38269
Answer      38252
dtype: int64
```

In [12]:

```
df2=pd.read_csv('Conversation.csv')
df2.head(10)
```

Out[12]:

	Unnamed: 0	question	answer
0	0	hi, how are you doing?	i'm fine. how about yourself?
1	1	i'm fine. how about yourself?	i'm pretty good. thanks for asking.
2	2	i'm pretty good. thanks for asking.	no problem. so how have you been?
3	3	no problem. so how have you been?	i've been great. what about you?
4	4	i've been great. what about you?	i've been good. i'm in school right now.

	Unnamed: 0	question	answer
5	5	i've been good. i'm in school right now.	what school do you go to?
6	6	what school do you go to?	i go to pcc.
7	7	i go to pcc.	do you like it there?
8	8	do you like it there?	it's okay. it's a really big campus.
9	9	it's okay. it's a really big campus.	good luck with school.

In [13]:

```
df2.count()
```

Out[13]:

```
Unnamed: 0    3832
question      3762
answer        3762
dtype: int64
```

DATASET CLEANING AND PREPROCESSING

In [14]:

```
df_cleaned=df.dropna()
df2_cleaned=df2.dropna()
```

In [15]:

```
df_cleaned.count()
```

Out[15]:

```
ID          38252
Question    38252
Answer      38252
dtype: int64
```

In [16]:

```
df2_cleaned.count()
```

Out[16]:

```
Unnamed: 0    3762
question      3762
answer        3762
dtype: int64
```

In [17]:

```
ques=[]
ans=[]
for index, row in df_cleaned.iterrows():
    ques.append(row['Question'])
    ans.append(row['Answer'])
```

In [18]:

```
for index2, row2 in df2_cleaned.iterrows():
    ques.append(row2['question'])
    ans.append(row2['answer'])
```

In [61]:

```
# ques[10:]
```

In [20]:

```
ques[:10]
```

Out[20]:

```
['Did you hear about the Native American man that drank 200 cups of tea?',
 "What's the best anti diarrheal prescription?",
 'What do you call a person who is outside a door and has no arms nor legs?',
 'Which Star Trek character is a member of the magic circle?',
 "What's the difference between a bullet and a human?",
 'Why was the Ethiopian baby crying?',
 "What's the difference between a corn husker with epilepsy and a hooker with dysentery?",
 "Who is 2016's biggest sellout?",
 "Why is little Annie's shoe floating in the sea?",
 "What's the difference between a married man and a bachelor?"]
```

In [21]:

```
len(ques)
```

Out[21]:

```
42014
```

In [ ]:

```
# type(ques[1000])
```

In [22]:

```
"""
    This section of code removes duplicate questions and corresponding answers from the dataset.
"""
```

```python
import re

unique_ques = []
unique_ans = []
ques_check = {}
for i in range(len(ques)):
    if len(ques[i]) < 30 and ques[i] not in ques_check:
      ques_check[ques[i]] = True
      unique_ques.append(ques[i])
      unique_ans.append(ans[i])



def clean_sent(sent):
    """
    Clean the input sentence by lowercasing it and expanding contractions.

    Args:
    sent (str): The input sentence to be cleaned.

    Returns:
    str: The cleaned sentence.
    """
    sent = sent.lower()
    sent= re.sub(r"[^\w\s]", "", sent)
    sent = re.sub(r"won't", "will not", sent)
    sent = re.sub(r"can't", "can not", sent)
    sent = re.sub(r"i'm", "i am", sent)
    sent = re.sub(r"he's", "he is", sent)
    sent = re.sub(r"she's", "she is", sent)
    sent = re.sub(r"\'ll", " will", sent)
    sent = re.sub(r"\'ve", " have", sent)
    sent = re.sub(r"\'re", " are", sent)
    sent = re.sub(r"\'d", " would", sent)
    sent = re.sub(r"that's", "that is", sent)
    sent = re.sub(r"what's", "what is", sent)
    sent = re.sub(r"where's", "where is", sent)
    return sent

cleaned_ques = []
cleaned_ans = []

for line in unique_ques:
    cleaned_ques.append(clean_sent(line))

for line in unique_ans:
    cleaned_ans.append(clean_sent(line))



del(ans, ques, line)
```

In [23]:

```
for i in range(len(cleaned_ans)):
    cleaned_ans[i] = ' '.join(cleaned_ans[i].split()[:28])




del(unique_ans, unique_ques)
```

In [27]:

```
"""
counts the occurrences of each word in both the cleaned questions and answers.
"""
words_count = {}

for line in cleaned_ques:
    for word in line.split():
        if word not in words_count:
            words_count[word] = 1
        else:
            words_count[word] += 1
for line in cleaned_ans:
    for word in line.split():
        if word not in words_count:
            words_count[word] = 1
        else:
            words_count[word] += 1
```

In [28]:

```
"""
 creates a vocabulary dictionary based on the word frequencies obtained previously.
"""
freq_thresh = 0

vocab_dict = {}
word_num = 0
for word, count in words_count.items():
    if count >= freq_thresh:
        vocab_dict[word] = word_num
        word_num += 1


del(words_count, word, count, freq_thresh)
del(word_num)
```

In [29]:

```
"""
    Preprocesses the answers by adding '<SOS>' (Start of Sentence) and '<EOS>' (End of Sentence) tokens
    at the beginning and end of each answer respectively. It also extends the vocabulary dictionary to include special tokens.
"""
for i in range(len(cleaned_ans)):
    cleaned_ans[i] = '<SOS> ' + cleaned_ans[i] + ' <EOS>'



tokens = ['<PAD>', '<EOS>', '<OUT>', '<SOS>']
vocab_len = len(vocab_dict)
for token in tokens:
    vocab_dict[token] = vocab_len
    vocab_len += 1

# vocab_dict
```

In [30]:

```
del(token, tokens)
del(vocab_len)

inv_vocab = {w:v for v, w in vocab_dict.items()}



## delete
del(i)
```

In [ ]:

```
# inv_vocab
```

BUILDING THE SEQ2SEQ MODEL

In [31]:

```
import tensorflow.keras.preprocessing.sequence
import tensorflow.keras.utils
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical
```

In [32]:

```
"""
    This function encodes sequences of words into sequences of integers using a given vocabulary dictionary.
"""
```

```python
def encode_sequences(cleaned_lst, vocab):
    encoded_seqs = []
    for sent in cleaned_lst:
        a=[]
        for word in sent.split():
            if word not in vocab:
                a.append(vocab['<OUT>'])
            else:
                a.append(vocab[word])
        encoded_seqs.append(a)
    return encoded_seqs

enc_input=encode_sequences(cleaned_ques,vocab_dict)
dec_input=encode_sequences(cleaned_ans,vocab_dict)
```

In [33]:

```python
"""
    Pad the encoded input sequences to ensure uniform length and remove the first token from the encoded output sequences.

    For the encoded input sequences:
    - The pad_sequences function pads each sequence with zeros to a maximum length of 30 tokens.
    - The padding parameter is set to 'post', which pads sequences at the end.
    - The truncating parameter is set to 'post', which truncates sequences from the end if they exceed the maximum length.

    For the decoded input sequences (final output):
    - The first token of each sequence is removed to exclude the '<SOS>' token.
    - Then, the sequences are padded to ensure uniform length, similar to the encoded input sequences.
"""


enc_input = pad_sequences(enc_input, 30, padding='post', truncating='post')
dec_input = pad_sequences(dec_input, 30, padding='post', truncating='post')





dec_final_output = []
for i in dec_input:
    dec_final_output.append(i[1:])

dec_final_output = pad_sequences(dec_final_output, 30, padding='post', truncating='post')


del(i)
```

In [34]:

```python
"""
```

```
    Converts the final output (decoded sequences) into one-hot encoded format using the to_categorical function from Keras.

    Additionally, it prints the shapes of the final output, encoded input, and decoded input sequences.
"""
dec_final_output = to_categorical(dec_final_output, len(vocab_dict))



print(dec_final_output.shape)
print(enc_input.shape)
print(dec_input.shape)
```

```
(4986, 30, 6895)
(4986, 30)
(4986, 30)
```

In [35]:

```
len(vocab_dict)
```

Out[35]:

```
6895
```

In [42]:

```
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, Embedding, LSTM, Input
```

In [60]:

```
"""
    This code defines a sequence-to-sequence model architecture for training.

    - `MAX_SEQUENCE_LENGTH`: Maximum length of input and output sequences.
    - `EMBEDDING_DIM`: Dimensionality of the word embeddings.
    - `LSTM_UNITS`: Number of units in the LSTM layer.
    - `VOCAB_SIZE`: Size of the vocabulary dictionary.

    The architecture consists of an encoder-decoder framework with an embedding layer, LSTM layers, and a dense layer.

    - `encoder_inp`: Input layer for the encoder sequences.
    - `decoder_inp`: Input layer for the decoder sequences.
    - `embed`: Embedding layer with trainable weights.
    - `encoder_embed`: Embedded representation of encoder input sequences.
    - `encoder_lstm`: LSTM layer for the encoder.
    - `encoder_op1`: Output sequence from the encoder LSTM.
    - `h1`, `c1`: Hidden and cell states of the encoder LSTM.
    - `encoder_states1`: States of the encoder LSTM.
    - `decoder_embed`: Embedded representation of decoder input sequences.
    - `decoder_lstm`: LSTM layer for the decoder.
```

```python
    - `decoder_op1`: Output sequence from the decoder LSTM.
    - `dense`: Dense layer for output prediction.
    - `dense_op`: Output sequence after applying the softmax activation function.
    - `model`: Sequence-to-sequence model.

    The model is compiled with categorical cross-entropy loss, accuracy metric, and the Adam optimizer.
"""
MAX_SEQUENCE_LENGTH = 30
EMBEDDING_DIM = 50
LSTM_UNITS = 300
VOCAB_SIZE = len(vocab_dict)

encoder_inp = Input(shape=(MAX_SEQUENCE_LENGTH,))
decoder_inp = Input(shape=(MAX_SEQUENCE_LENGTH,))

embed = Embedding(VOCAB_SIZE+1, output_dim=EMBEDDING_DIM, input_length=MAX_SEQUENCE_LENGTH,trainable=True)

encoder_embed = embed(encoder_inp)
encoder_lstm = LSTM(LSTM_UNITS, return_sequences=True, return_state=True)
encoder_op1,h1,c1=encoder_lstm(encoder_embed)
encoder_states1 = [h1, c1]

decoder_embed = embed(decoder_inp)
decoder_lstm = LSTM(LSTM_UNITS, return_sequences=True, return_state=True)
decoder_op1, _, _=decoder_lstm(decoder_embed, initial_state=encoder_states1)

dense = Dense(VOCAB_SIZE, activation='softmax')
dense_op = dense(decoder_op1)

model = Model([encoder_inp, decoder_inp], dense_op)

model.compile(loss='categorical_crossentropy', metrics=['accuracy'], optimizer='adam')
```

In [44]:

```python
# Train the model

model.fit([enc_input, dec_input],dec_final_output,epochs= 500)
```

```
Epoch 1/500
156/156 [==============================] - 12s 58ms/step - loss: 2.6970 - accuracy: 0.7122
Epoch 2/500
156/156 [==============================] - 4s 26ms/step - loss: 1.8298 - accuracy: 0.7486
Epoch 3/500
156/156 [==============================] - 3s 22ms/step - loss: 1.6898 - accuracy: 0.7601
Epoch 4/500
156/156 [==============================] - 4s 22ms/step - loss: 1.6524 - accuracy: 0.7607
Epoch 5/500
156/156 [==============================] - 4s 23ms/step - loss: 1.6173 - accuracy: 0.7627
Epoch 6/500
156/156 [==============================] - 3s 20ms/step - loss: 1.5814 - accuracy: 0.7637
```

```
Epoch 7/500
156/156 [==============================] - 3s 19ms/step - loss: 1.5478 - accuracy: 0.7653
Epoch 8/500
156/156 [==============================] - 3s 20ms/step - loss: 1.5175 - accuracy: 0.7668
Epoch 9/500
156/156 [==============================] - 3s 20ms/step - loss: 1.4889 - accuracy: 0.7678
Epoch 10/500
156/156 [==============================] - 3s 21ms/step - loss: 1.4604 - accuracy: 0.7693
Epoch 11/500
156/156 [==============================] - 3s 20ms/step - loss: 1.4325 - accuracy: 0.7709
Epoch 12/500
156/156 [==============================] - 3s 20ms/step - loss: 1.4043 - accuracy: 0.7723
Epoch 13/500
156/156 [==============================] - 3s 18ms/step - loss: 1.3752 - accuracy: 0.7739
Epoch 14/500
156/156 [==============================] - 3s 20ms/step - loss: 1.3466 - accuracy: 0.7758
Epoch 15/500
156/156 [==============================] - 3s 18ms/step - loss: 1.3173 - accuracy: 0.7771
Epoch 16/500
156/156 [==============================] - 3s 19ms/step - loss: 1.2884 - accuracy: 0.7793
Epoch 17/500
156/156 [==============================] - 3s 19ms/step - loss: 1.2593 - accuracy: 0.7809
Epoch 18/500
156/156 [==============================] - 3s 21ms/step - loss: 1.2303 - accuracy: 0.7829
Epoch 19/500
156/156 [==============================] - 3s 20ms/step - loss: 1.2005 - accuracy: 0.7847
Epoch 20/500
156/156 [==============================] - 3s 19ms/step - loss: 1.1720 - accuracy: 0.7869
Epoch 21/500
156/156 [==============================] - 3s 19ms/step - loss: 1.1437 - accuracy: 0.7892
Epoch 22/500
156/156 [==============================] - 3s 21ms/step - loss: 1.1165 - accuracy: 0.7915
Epoch 23/500
156/156 [==============================] - 3s 18ms/step - loss: 1.0888 - accuracy: 0.7942
Epoch 24/500
156/156 [==============================] - 3s 20ms/step - loss: 1.0618 - accuracy: 0.7981
Epoch 25/500
156/156 [==============================] - 3s 18ms/step - loss: 1.0341 - accuracy: 0.8017
Epoch 26/500
156/156 [==============================] - 3s 19ms/step - loss: 1.0066 - accuracy: 0.8061
Epoch 27/500
156/156 [==============================] - 3s 19ms/step - loss: 0.9798 - accuracy: 0.8101
Epoch 28/500
156/156 [==============================] - 3s 20ms/step - loss: 0.9523 - accuracy: 0.8140
Epoch 29/500
156/156 [==============================] - 3s 18ms/step - loss: 0.9254 - accuracy: 0.8184
Epoch 30/500
156/156 [==============================] - 3s 18ms/step - loss: 0.8994 - accuracy: 0.8220
Epoch 31/500
156/156 [==============================] - 3s 18ms/step - loss: 0.8752 - accuracy: 0.8264
Epoch 32/500
```

```
156/156 [==============================] - 3s 18ms/step - loss: 0.8512 - accuracy: 0.8298
Epoch 33/500
156/156 [==============================] - 3s 20ms/step - loss: 0.8290 - accuracy: 0.8334
Epoch 34/500
156/156 [==============================] - 3s 19ms/step - loss: 0.8080 - accuracy: 0.8366
Epoch 35/500
156/156 [==============================] - 3s 19ms/step - loss: 0.7866 - accuracy: 0.8402
Epoch 36/500
156/156 [==============================] - 3s 18ms/step - loss: 0.7662 - accuracy: 0.8434
Epoch 37/500
156/156 [==============================] - 3s 18ms/step - loss: 0.7466 - accuracy: 0.8467
Epoch 38/500
156/156 [==============================] - 3s 18ms/step - loss: 0.7279 - accuracy: 0.8493
Epoch 39/500
156/156 [==============================] - 3s 19ms/step - loss: 0.7104 - accuracy: 0.8521
Epoch 40/500
156/156 [==============================] - 3s 18ms/step - loss: 0.6929 - accuracy: 0.8555
Epoch 41/500
156/156 [==============================] - 3s 18ms/step - loss: 0.6789 - accuracy: 0.8578
Epoch 42/500
156/156 [==============================] - 3s 18ms/step - loss: 0.6621 - accuracy: 0.8608
Epoch 43/500
156/156 [==============================] - 3s 18ms/step - loss: 0.6452 - accuracy: 0.8640
Epoch 44/500
156/156 [==============================] - 3s 19ms/step - loss: 0.6301 - accuracy: 0.8666
Epoch 45/500
156/156 [==============================] - 3s 18ms/step - loss: 0.6161 - accuracy: 0.8694
Epoch 46/500
156/156 [==============================] - 3s 18ms/step - loss: 0.6027 - accuracy: 0.8717
Epoch 47/500
156/156 [==============================] - 3s 18ms/step - loss: 0.5879 - accuracy: 0.8749
Epoch 48/500
156/156 [==============================] - 3s 19ms/step - loss: 0.5770 - accuracy: 0.8772
Epoch 49/500
156/156 [==============================] - 3s 19ms/step - loss: 0.5644 - accuracy: 0.8800
Epoch 50/500
156/156 [==============================] - 3s 20ms/step - loss: 0.5496 - accuracy: 0.8826
Epoch 51/500
156/156 [==============================] - 3s 18ms/step - loss: 0.5384 - accuracy: 0.8847
Epoch 52/500
156/156 [==============================] - 3s 18ms/step - loss: 0.5273 - accuracy: 0.8873
Epoch 53/500
156/156 [==============================] - 3s 18ms/step - loss: 0.5153 - accuracy: 0.8895
Epoch 54/500
156/156 [==============================] - 3s 18ms/step - loss: 0.5038 - accuracy: 0.8916
Epoch 55/500
156/156 [==============================] - 3s 18ms/step - loss: 0.4931 - accuracy: 0.8941
Epoch 56/500
156/156 [==============================] - 3s 19ms/step - loss: 0.4813 - accuracy: 0.8969
Epoch 57/500
156/156 [==============================] - 3s 20ms/step - loss: 0.4714 - accuracy: 0.8986
```

```
Epoch 58/500
156/156 [==============================] - 3s 19ms/step - loss: 0.4620 - accuracy: 0.9006
Epoch 59/500
156/156 [==============================] - 3s 19ms/step - loss: 0.4515 - accuracy: 0.9029
Epoch 60/500
156/156 [==============================] - 3s 20ms/step - loss: 0.4428 - accuracy: 0.9045
Epoch 61/500
156/156 [==============================] - 3s 18ms/step - loss: 0.4322 - accuracy: 0.9068
Epoch 62/500
156/156 [==============================] - 3s 19ms/step - loss: 0.4210 - accuracy: 0.9093
Epoch 63/500
156/156 [==============================] - 3s 19ms/step - loss: 0.4111 - accuracy: 0.9114
Epoch 64/500
156/156 [==============================] - 3s 19ms/step - loss: 0.4016 - accuracy: 0.9134
Epoch 65/500
156/156 [==============================] - 3s 19ms/step - loss: 0.3930 - accuracy: 0.9151
Epoch 66/500
156/156 [==============================] - 3s 18ms/step - loss: 0.3845 - accuracy: 0.9168
Epoch 67/500
156/156 [==============================] - 3s 18ms/step - loss: 0.3758 - accuracy: 0.9188
Epoch 68/500
156/156 [==============================] - 3s 19ms/step - loss: 0.3690 - accuracy: 0.9196
Epoch 69/500
156/156 [==============================] - 3s 19ms/step - loss: 0.3591 - accuracy: 0.9218
Epoch 70/500
156/156 [==============================] - 3s 19ms/step - loss: 0.3503 - accuracy: 0.9235
Epoch 71/500
156/156 [==============================] - 3s 18ms/step - loss: 0.3429 - accuracy: 0.9251
Epoch 72/500
156/156 [==============================] - 3s 18ms/step - loss: 0.3329 - accuracy: 0.9277
Epoch 73/500
156/156 [==============================] - 3s 19ms/step - loss: 0.3232 - accuracy: 0.9293
Epoch 74/500
156/156 [==============================] - 3s 19ms/step - loss: 0.3158 - accuracy: 0.9310
Epoch 75/500
156/156 [==============================] - 3s 18ms/step - loss: 0.3090 - accuracy: 0.9327
Epoch 76/500
156/156 [==============================] - 3s 18ms/step - loss: 0.2996 - accuracy: 0.9345
Epoch 77/500
156/156 [==============================] - 3s 18ms/step - loss: 0.2909 - accuracy: 0.9364
Epoch 78/500
156/156 [==============================] - 3s 19ms/step - loss: 0.2847 - accuracy: 0.9376
Epoch 79/500
156/156 [==============================] - 3s 18ms/step - loss: 0.2768 - accuracy: 0.9399
Epoch 80/500
156/156 [==============================] - 3s 18ms/step - loss: 0.2707 - accuracy: 0.9404
Epoch 81/500
156/156 [==============================] - 3s 19ms/step - loss: 0.2621 - accuracy: 0.9423
Epoch 82/500
156/156 [==============================] - 3s 20ms/step - loss: 0.2561 - accuracy: 0.9433
Epoch 83/500
```

```
156/156 [==============================] - 3s 18ms/step - loss: 0.2473 - accuracy: 0.9457
Epoch 84/500
156/156 [==============================] - 3s 18ms/step - loss: 0.2402 - accuracy: 0.9471
Epoch 85/500
156/156 [==============================] - 3s 18ms/step - loss: 0.2333 - accuracy: 0.9484
Epoch 86/500
156/156 [==============================] - 3s 19ms/step - loss: 0.2268 - accuracy: 0.9502
Epoch 87/500
156/156 [==============================] - 3s 18ms/step - loss: 0.2208 - accuracy: 0.9514
Epoch 88/500
156/156 [==============================] - 3s 18ms/step - loss: 0.2137 - accuracy: 0.9535
Epoch 89/500
156/156 [==============================] - 3s 18ms/step - loss: 0.2077 - accuracy: 0.9543
Epoch 90/500
156/156 [==============================] - 3s 20ms/step - loss: 0.2014 - accuracy: 0.9559
Epoch 91/500
156/156 [==============================] - 3s 18ms/step - loss: 0.1942 - accuracy: 0.9576
Epoch 92/500
156/156 [==============================] - 3s 18ms/step - loss: 0.1898 - accuracy: 0.9587
Epoch 93/500
156/156 [==============================] - 3s 18ms/step - loss: 0.1847 - accuracy: 0.9597
Epoch 94/500
156/156 [==============================] - 3s 19ms/step - loss: 0.1789 - accuracy: 0.9608
Epoch 95/500
156/156 [==============================] - 3s 18ms/step - loss: 0.1730 - accuracy: 0.9626
Epoch 96/500
156/156 [==============================] - 3s 18ms/step - loss: 0.1664 - accuracy: 0.9643
Epoch 97/500
156/156 [==============================] - 3s 18ms/step - loss: 0.1601 - accuracy: 0.9659
Epoch 98/500
156/156 [==============================] - 3s 18ms/step - loss: 0.1559 - accuracy: 0.9670
Epoch 99/500
156/156 [==============================] - 3s 19ms/step - loss: 0.1506 - accuracy: 0.9682
Epoch 100/500
156/156 [==============================] - 3s 19ms/step - loss: 0.1452 - accuracy: 0.9695
Epoch 101/500
156/156 [==============================] - 3s 18ms/step - loss: 0.1416 - accuracy: 0.9700
Epoch 102/500
156/156 [==============================] - 3s 18ms/step - loss: 0.1353 - accuracy: 0.9714
Epoch 103/500
156/156 [==============================] - 3s 19ms/step - loss: 0.1316 - accuracy: 0.9726
Epoch 104/500
156/156 [==============================] - 3s 19ms/step - loss: 0.1262 - accuracy: 0.9736
Epoch 105/500
156/156 [==============================] - 3s 18ms/step - loss: 0.1202 - accuracy: 0.9755
Epoch 106/500
156/156 [==============================] - 3s 18ms/step - loss: 0.1158 - accuracy: 0.9766
Epoch 107/500
156/156 [==============================] - 3s 18ms/step - loss: 0.1131 - accuracy: 0.9769
Epoch 108/500
156/156 [==============================] - 3s 18ms/step - loss: 0.1083 - accuracy: 0.9784
Epoch 109/500
```

```
Epoch 109/500
156/156 [==============================] - 3s 18ms/step - loss: 0.1051 - accuracy: 0.9787
Epoch 110/500
156/156 [==============================] - 3s 18ms/step - loss: 0.1040 - accuracy: 0.9791
Epoch 111/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0998 - accuracy: 0.9801
Epoch 112/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0956 - accuracy: 0.9810
Epoch 113/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0910 - accuracy: 0.9823
Epoch 114/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0881 - accuracy: 0.9831
Epoch 115/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0839 - accuracy: 0.9840
Epoch 116/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0817 - accuracy: 0.9844
Epoch 117/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0781 - accuracy: 0.9852
Epoch 118/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0741 - accuracy: 0.9862
Epoch 119/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0704 - accuracy: 0.9874
Epoch 120/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0676 - accuracy: 0.9881
Epoch 121/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0645 - accuracy: 0.9888
Epoch 122/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0633 - accuracy: 0.9888
Epoch 123/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0632 - accuracy: 0.9887
Epoch 124/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0612 - accuracy: 0.9894
Epoch 125/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0582 - accuracy: 0.9900
Epoch 126/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0554 - accuracy: 0.9908
Epoch 127/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0514 - accuracy: 0.9919
Epoch 128/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0488 - accuracy: 0.9923
Epoch 129/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0462 - accuracy: 0.9928
Epoch 130/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0439 - accuracy: 0.9933
Epoch 131/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0440 - accuracy: 0.9931
Epoch 132/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0426 - accuracy: 0.9935
Epoch 133/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0407 - accuracy: 0.9940
Epoch 134/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0440 - accuracy: 0.9935
```

```
156/156 [==============================] - 3s 19ms/step - loss: 0.0440 - accuracy: 0.9935
Epoch 135/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0473 - accuracy: 0.9923
Epoch 136/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0430 - accuracy: 0.9929
Epoch 137/500
156/156 [==============================] - 3s 20ms/step - loss: 0.0381 - accuracy: 0.9943
Epoch 138/500
156/156 [==============================] - 3s 20ms/step - loss: 0.0338 - accuracy: 0.9952
Epoch 139/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0292 - accuracy: 0.9962
Epoch 140/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0266 - accuracy: 0.9968
Epoch 141/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0254 - accuracy: 0.9969
Epoch 142/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0253 - accuracy: 0.9967
Epoch 143/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0246 - accuracy: 0.9969
Epoch 144/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0236 - accuracy: 0.9971
Epoch 145/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0226 - accuracy: 0.9971
Epoch 146/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0220 - accuracy: 0.9973
Epoch 147/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0232 - accuracy: 0.9969
Epoch 148/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0277 - accuracy: 0.9956
Epoch 149/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0298 - accuracy: 0.9950
Epoch 150/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0444 - accuracy: 0.9910
Epoch 151/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0424 - accuracy: 0.9921
Epoch 152/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0332 - accuracy: 0.9945
Epoch 153/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0231 - accuracy: 0.9970
Epoch 154/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0169 - accuracy: 0.9980
Epoch 155/500
156/156 [==============================] - 3s 20ms/step - loss: 0.0143 - accuracy: 0.9983
Epoch 156/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0126 - accuracy: 0.9984
Epoch 157/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0119 - accuracy: 0.9985
Epoch 158/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0112 - accuracy: 0.9985
Epoch 159/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0108 - accuracy: 0.9986
Epoch 160/500
```

```
Epoch 160/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0112 - accuracy: 0.9985
Epoch 161/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0108 - accuracy: 0.9985
Epoch 162/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0102 - accuracy: 0.9986
Epoch 163/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0102 - accuracy: 0.9986
Epoch 164/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0101 - accuracy: 0.9985
Epoch 165/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0101 - accuracy: 0.9986
Epoch 166/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0111 - accuracy: 0.9983
Epoch 167/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0351 - accuracy: 0.9918
Epoch 168/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0667 - accuracy: 0.9828
Epoch 169/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0397 - accuracy: 0.9911
Epoch 170/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0198 - accuracy: 0.9968
Epoch 171/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0118 - accuracy: 0.9983
Epoch 172/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0090 - accuracy: 0.9986
Epoch 173/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0079 - accuracy: 0.9987
Epoch 174/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0073 - accuracy: 0.9986
Epoch 175/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0069 - accuracy: 0.9988
Epoch 176/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0067 - accuracy: 0.9987
Epoch 177/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0065 - accuracy: 0.9987
Epoch 178/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0063 - accuracy: 0.9988
Epoch 179/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0063 - accuracy: 0.9988
Epoch 180/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0062 - accuracy: 0.9987
Epoch 181/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0062 - accuracy: 0.9988
Epoch 182/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0062 - accuracy: 0.9988
Epoch 183/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0074 - accuracy: 0.9986
Epoch 184/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0154 - accuracy: 0.9968
Epoch 185/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0533 - accuracy: 0.9858
```

```
Epoch 186/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0503 - accuracy: 0.9872
Epoch 187/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0220 - accuracy: 0.9955
Epoch 188/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0111 - accuracy: 0.9981
Epoch 189/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0072 - accuracy: 0.9987
Epoch 190/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0064 - accuracy: 0.9987
Epoch 191/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0059 - accuracy: 0.9988
Epoch 192/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0055 - accuracy: 0.9988
Epoch 193/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0053 - accuracy: 0.9988
Epoch 194/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0051 - accuracy: 0.9988
Epoch 195/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0050 - accuracy: 0.9988
Epoch 196/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0050 - accuracy: 0.9987
Epoch 197/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0050 - accuracy: 0.9988
Epoch 198/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0049 - accuracy: 0.9987
Epoch 199/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0050 - accuracy: 0.9987
Epoch 200/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0051 - accuracy: 0.9987
Epoch 201/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0056 - accuracy: 0.9987
Epoch 202/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0099 - accuracy: 0.9978
Epoch 203/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0505 - accuracy: 0.9861
Epoch 204/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0421 - accuracy: 0.9892
Epoch 205/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0192 - accuracy: 0.9957
Epoch 206/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0090 - accuracy: 0.9983
Epoch 207/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0062 - accuracy: 0.9987
Epoch 208/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0051 - accuracy: 0.9988
Epoch 209/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0047 - accuracy: 0.9988
Epoch 210/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0046 - accuracy: 0.9988
Epoch 211/500
```

```
Epoch 211/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0044 - accuracy: 0.9988
Epoch 212/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0043 - accuracy: 0.9989
Epoch 213/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0043 - accuracy: 0.9988
Epoch 214/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0043 - accuracy: 0.9988
Epoch 215/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0043 - accuracy: 0.9987
Epoch 216/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0043 - accuracy: 0.9987
Epoch 217/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0043 - accuracy: 0.9988
Epoch 218/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0043 - accuracy: 0.9988
Epoch 219/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0043 - accuracy: 0.9987
Epoch 220/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0046 - accuracy: 0.9987
Epoch 221/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0051 - accuracy: 0.9987
Epoch 222/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0172 - accuracy: 0.9956
Epoch 223/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0537 - accuracy: 0.9849
Epoch 224/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0334 - accuracy: 0.9915
Epoch 225/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0140 - accuracy: 0.9968
Epoch 226/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0068 - accuracy: 0.9985
Epoch 227/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0051 - accuracy: 0.9987
Epoch 228/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0043 - accuracy: 0.9988
Epoch 229/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0041 - accuracy: 0.9988
Epoch 230/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0039 - accuracy: 0.9988
Epoch 231/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0039 - accuracy: 0.9987
Epoch 232/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0039 - accuracy: 0.9987
Epoch 233/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0038 - accuracy: 0.9987
Epoch 234/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0036 - accuracy: 0.9988
Epoch 235/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0037 - accuracy: 0.9988
Epoch 236/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0037 - accuracy: 0.9987
```

```
156/156 [==============================] - 3s 19ms/step - loss: 0.0037 - accuracy: 0.9987
Epoch 237/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0037 - accuracy: 0.9988
Epoch 238/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0037 - accuracy: 0.9987
Epoch 239/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0037 - accuracy: 0.9988
Epoch 240/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0037 - accuracy: 0.9988
Epoch 241/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0039 - accuracy: 0.9987
Epoch 242/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0037 - accuracy: 0.9987
Epoch 243/500
156/156 [==============================] - 3s 20ms/step - loss: 0.0037 - accuracy: 0.9989
Epoch 244/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0038 - accuracy: 0.9987
Epoch 245/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0041 - accuracy: 0.9987
Epoch 246/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0401 - accuracy: 0.9890
Epoch 247/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0622 - accuracy: 0.9824
Epoch 248/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0232 - accuracy: 0.9940
Epoch 249/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0088 - accuracy: 0.9979
Epoch 250/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0050 - accuracy: 0.9986
Epoch 251/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0042 - accuracy: 0.9987
Epoch 252/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0038 - accuracy: 0.9988
Epoch 253/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0037 - accuracy: 0.9987
Epoch 254/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0035 - accuracy: 0.9988
Epoch 255/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0034 - accuracy: 0.9988
Epoch 256/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0034 - accuracy: 0.9988
Epoch 257/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0033 - accuracy: 0.9988
Epoch 258/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0034 - accuracy: 0.9988
Epoch 259/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0034 - accuracy: 0.9988
Epoch 260/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0033 - accuracy: 0.9988
Epoch 261/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0035 - accuracy: 0.9987
Epoch 262/500
```

```
156/156 [==============================] - 3s 18ms/step - loss: 0.0034 - accuracy: 0.9988
Epoch 263/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0033 - accuracy: 0.9988
Epoch 264/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0033 - accuracy: 0.9988
Epoch 265/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0033 - accuracy: 0.9988
Epoch 266/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0033 - accuracy: 0.9987
Epoch 267/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0035 - accuracy: 0.9988
Epoch 268/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0043 - accuracy: 0.9986
Epoch 269/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0223 - accuracy: 0.9937
Epoch 270/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0512 - accuracy: 0.9852
Epoch 271/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0262 - accuracy: 0.9931
Epoch 272/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0104 - accuracy: 0.9976
Epoch 273/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0053 - accuracy: 0.9987
Epoch 274/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0041 - accuracy: 0.9987
Epoch 275/500
156/156 [==============================] - 3s 20ms/step - loss: 0.0036 - accuracy: 0.9988
Epoch 276/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0033 - accuracy: 0.9988
Epoch 277/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0033 - accuracy: 0.9988
Epoch 278/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0033 - accuracy: 0.9988
Epoch 279/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0032 - accuracy: 0.9988
Epoch 280/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0031 - accuracy: 0.9988
Epoch 281/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0032 - accuracy: 0.9988
Epoch 282/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0031 - accuracy: 0.9988
Epoch 283/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0031 - accuracy: 0.9987
Epoch 284/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0032 - accuracy: 0.9987
Epoch 285/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0031 - accuracy: 0.9987
Epoch 286/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0032 - accuracy: 0.9987
Epoch 287/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0031 - accuracy: 0.9988
```

```
Epoch 288/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0031 - accuracy: 0.9988
Epoch 289/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0031 - accuracy: 0.9988
Epoch 290/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0035 - accuracy: 0.9987
Epoch 291/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0076 - accuracy: 0.9978
Epoch 292/500
156/156 [==============================] - 3s 20ms/step - loss: 0.0456 - accuracy: 0.9870
Epoch 293/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0314 - accuracy: 0.9909
Epoch 294/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0124 - accuracy: 0.9970
Epoch 295/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0053 - accuracy: 0.9986
Epoch 296/500
156/156 [==============================] - 3s 20ms/step - loss: 0.0040 - accuracy: 0.9987
Epoch 297/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0034 - accuracy: 0.9988
Epoch 298/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0032 - accuracy: 0.9988
Epoch 299/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0031 - accuracy: 0.9988
Epoch 300/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0030 - accuracy: 0.9988
Epoch 301/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0030 - accuracy: 0.9988
Epoch 302/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0030 - accuracy: 0.9988
Epoch 303/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0030 - accuracy: 0.9988
Epoch 304/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0030 - accuracy: 0.9988
Epoch 305/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0030 - accuracy: 0.9988
Epoch 306/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0030 - accuracy: 0.9988
Epoch 307/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0029 - accuracy: 0.9988
Epoch 308/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0029 - accuracy: 0.9988
Epoch 309/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0029 - accuracy: 0.9988
Epoch 310/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0030 - accuracy: 0.9988
Epoch 311/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0029 - accuracy: 0.9987
Epoch 312/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0031 - accuracy: 0.9988
Epoch 313/500
```

```
156/156 [==============================] - 3s 19ms/step - loss: 0.0032 - accuracy: 0.9988
Epoch 314/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0036 - accuracy: 0.9988
Epoch 315/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0192 - accuracy: 0.9947
Epoch 316/500
156/156 [==============================] - 3s 20ms/step - loss: 0.0498 - accuracy: 0.9855
Epoch 317/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0185 - accuracy: 0.9952
Epoch 318/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0079 - accuracy: 0.9979
Epoch 319/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0045 - accuracy: 0.9987
Epoch 320/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0034 - accuracy: 0.9987
Epoch 321/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0031 - accuracy: 0.9987
Epoch 322/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0030 - accuracy: 0.9988
Epoch 323/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0029 - accuracy: 0.9988
Epoch 324/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0030 - accuracy: 0.9988
Epoch 325/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0029 - accuracy: 0.9988
Epoch 326/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0029 - accuracy: 0.9988
Epoch 327/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0029 - accuracy: 0.9988
Epoch 328/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0029 - accuracy: 0.9988
Epoch 329/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0028 - accuracy: 0.9987
Epoch 330/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0028 - accuracy: 0.9988
Epoch 331/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0028 - accuracy: 0.9988
Epoch 332/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0029 - accuracy: 0.9988
Epoch 333/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0028 - accuracy: 0.9988
Epoch 334/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0031 - accuracy: 0.9988
Epoch 335/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0031 - accuracy: 0.9987
Epoch 336/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0034 - accuracy: 0.9987
Epoch 337/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0069 - accuracy: 0.9981
Epoch 338/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0313 - accuracy: 0.9912
```

```
Epoch 339/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0254 - accuracy: 0.9927
Epoch 340/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0108 - accuracy: 0.9972
Epoch 341/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0054 - accuracy: 0.9985
Epoch 342/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0037 - accuracy: 0.9987
Epoch 343/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0031 - accuracy: 0.9988
Epoch 344/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0030 - accuracy: 0.9987
Epoch 345/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0028 - accuracy: 0.9988
Epoch 346/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0028 - accuracy: 0.9989
Epoch 347/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0028 - accuracy: 0.9988
Epoch 348/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0028 - accuracy: 0.9988
Epoch 349/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 350/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 351/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 352/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0027 - accuracy: 0.9987
Epoch 353/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0028 - accuracy: 0.9987
Epoch 354/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 355/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0028 - accuracy: 0.9987
Epoch 356/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0028 - accuracy: 0.9988
Epoch 357/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0028 - accuracy: 0.9988
Epoch 358/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0029 - accuracy: 0.9988
Epoch 359/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0113 - accuracy: 0.9966
Epoch 360/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0397 - accuracy: 0.9884
Epoch 361/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0206 - accuracy: 0.9943
Epoch 362/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0083 - accuracy: 0.9977
Epoch 363/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0046 - accuracy: 0.9985
Epoch 364/500
```

```
156/156 [==============================] - 3s 18ms/step - loss: 0.0032 - accuracy: 0.9988
Epoch 365/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0029 - accuracy: 0.9988
Epoch 366/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0028 - accuracy: 0.9988
Epoch 367/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 368/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 369/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 370/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 371/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 372/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 373/500
156/156 [==============================] - 3s 20ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 374/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 375/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 376/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 377/500
156/156 [==============================] - 3s 20ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 378/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 379/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 380/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 381/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 382/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 383/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0028 - accuracy: 0.9987
Epoch 384/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0028 - accuracy: 0.9988
Epoch 385/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0086 - accuracy: 0.9973
Epoch 386/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0511 - accuracy: 0.9847
Epoch 387/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0213 - accuracy: 0.9940
Epoch 388/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0086 - accuracy: 0.9976
Epoch 389/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0042 - accuracy: 0.9986
```

```
Epoch 390/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0031 - accuracy: 0.9988
Epoch 391/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0028 - accuracy: 0.9988
Epoch 392/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0028 - accuracy: 0.9988
Epoch 393/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 394/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 395/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 396/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 397/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 398/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 399/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 400/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 401/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 402/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0027 - accuracy: 0.9987
Epoch 403/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9987
Epoch 404/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 405/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 406/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9989
Epoch 407/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 408/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9987
Epoch 409/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0029 - accuracy: 0.9987
Epoch 410/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0198 - accuracy: 0.9939
Epoch 411/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0355 - accuracy: 0.9896
Epoch 412/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0152 - accuracy: 0.9957
Epoch 413/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0070 - accuracy: 0.9980
Epoch 414/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0042 - accuracy: 0.9986
Epoch 415/500
```

```
156/156 [==============================] - 3s 19ms/step - loss: 0.0031 - accuracy: 0.9988
Epoch 416/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0028 - accuracy: 0.9988
Epoch 417/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 418/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 419/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 420/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9989
Epoch 421/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 422/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 423/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 424/500
156/156 [==============================] - 3s 20ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 425/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9987
Epoch 426/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 427/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9989
Epoch 428/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 429/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 430/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 431/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 432/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 433/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 434/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 435/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 436/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0027 - accuracy: 0.9987
Epoch 437/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 438/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0029 - accuracy: 0.9988
Epoch 439/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0399 - accuracy: 0.9881
Epoch 440/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0299 - accuracy: 0.9913
```

```
Epoch 441/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0102 - accuracy: 0.9971
Epoch 442/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0046 - accuracy: 0.9986
Epoch 443/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0031 - accuracy: 0.9987
Epoch 444/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0028 - accuracy: 0.9987
Epoch 445/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 446/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 447/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 448/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 449/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0025 - accuracy: 0.9987
Epoch 450/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0025 - accuracy: 0.9989
Epoch 451/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 452/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9989
Epoch 453/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0024 - accuracy: 0.9988
Epoch 454/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 455/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 456/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0024 - accuracy: 0.9988
Epoch 457/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 458/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9989
Epoch 459/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 460/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 461/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 462/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 463/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 464/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0024 - accuracy: 0.9988
Epoch 465/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 466/500
```

```
156/156 [==============================] - 3s 18ms/step - loss: 0.0075 - accuracy: 0.9976
Epoch 467/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0507 - accuracy: 0.9848
Epoch 468/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0198 - accuracy: 0.9942
Epoch 469/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0071 - accuracy: 0.9978
Epoch 470/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0035 - accuracy: 0.9987
Epoch 471/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0032 - accuracy: 0.9987
Epoch 472/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0035 - accuracy: 0.9987
Epoch 473/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0029 - accuracy: 0.9988
Epoch 474/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0027 - accuracy: 0.9988
Epoch 475/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 476/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0075 - accuracy: 0.9983
Epoch 477/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0036 - accuracy: 0.9987
Epoch 478/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0029 - accuracy: 0.9988
Epoch 479/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0026 - accuracy: 0.9988
Epoch 480/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 481/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0024 - accuracy: 0.9988
Epoch 482/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0024 - accuracy: 0.9989
Epoch 483/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0024 - accuracy: 0.9988
Epoch 484/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0024 - accuracy: 0.9988
Epoch 485/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 486/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 487/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9987
Epoch 488/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9987
Epoch 489/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0023 - accuracy: 0.9989
Epoch 490/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 491/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9988
```

```
Epoch 492/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0024 - accuracy: 0.9988
Epoch 493/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0024 - accuracy: 0.9988
Epoch 494/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0025 - accuracy: 0.9987
Epoch 495/500
156/156 [==============================] - 3s 19ms/step - loss: 0.0025 - accuracy: 0.9988
Epoch 496/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0029 - accuracy: 0.9987
Epoch 497/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0308 - accuracy: 0.9908
Epoch 498/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0257 - accuracy: 0.9925
Epoch 499/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0105 - accuracy: 0.9968
Epoch 500/500
156/156 [==============================] - 3s 18ms/step - loss: 0.0047 - accuracy: 0.9985
```

Out[44]:

```
<keras.src.callbacks.History at 0x7e4d50098700>
```

In [46]:

```
%cd /content/drive/MyDrive/NLP_PROJECT2/Models
model.save('seq2seq_model2.keras')
```

```
/content/drive/MyDrive/NLP_PROJECT2/Models
```

In [38]:

```
# from tensorflow.keras.models import load_model
# %cd /content/drive/MyDrive/NLP_PROJECT2/Models
# # Load model from Google Drive
# model_path = '/content/drive/MyDrive/NLP_PROJECT2/Models/seq2seq_model1.keras'
# model = load_model(model_path)
```

```
/content/drive/MyDrive/NLP_PROJECT2/Models
```

In [47]:

```
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input

LSTM_UNITS = 300
# Define the encoder model

encoder_model = Model([encoder_inp], encoder_states1)
```

```
# Define input layers for the decoder

decoder_state_input_h = Input(shape=(LSTM_UNITS,))
decoder_state_input_c = Input(shape=(LSTM_UNITS,))

dec_model_states_inputs = [decoder_state_input_h, decoder_state_input_c]

# Define the decoder outputs and states

decoder_outputs, state_h, state_c = decoder_lstm(decoder_embed ,
                                    initial_state=dec_model_states_inputs)


decoder_states = [state_h, state_c]

# Define the decoder model

decoder_model = Model([decoder_inp]+ dec_model_states_inputs,
                                      [decoder_outputs]+ decoder_states)
```

TESTING OUT THE TRAINED SEQ2SEQ MODEL

In [49]:

```
import numpy as np
from keras.preprocessing.sequence import pad_sequences

print("*******************************************")
print("*          Welcome to JestMaster ʕ•ᴥ•ʔ!         *")
print("*         Where Every Line is a Punchline       *")
print("*******************************************")

def preprocess_input(user_input):
    """
    Preprocesses the user input by cleaning, tokenizing, encoding, and padding it for model input.

    Args:
    - user_input (str): The raw user input text.

    Returns:
    - numpy.ndarray: The preprocessed and padded input sequence as a 2D array.
    """
    preprocessed_input = clean_sent(user_input)
    # print("preprocessed_input1",preprocessed_input)
    preprocessed_input=[preprocessed_input]
    # print("preprocessed_input2",preprocessed_input)
    encoded_user_input=[]
    for inp in preprocessed_input:
        x=[]
```

```
        for i in inp.split():
            try:
                x.append(vocab_dict[i])
            except:
                x.append(vocab_dict['<OUT>'])
        encoded_user_input.append(x)
    # print("encoded_user_input",encoded_user_input)
    padded_user_input = pad_sequences(encoded_user_input, maxlen=MAX_SEQUENCE_LENGTH, padding='post')
    # print("padded_user_input",padded_user_input)
    # encoded_input = [vocab_dict.get(word, vocab_dict['<OUT>']) for word in preprocessed_input.split()]
    return padded_user_input

def generate_punch_line(input_sequence):
    """
    Generate a punchline based on the input sequence using the trained encoder-decoder model.

    Args:
    - input_sequence (numpy.ndarray): The preprocessed input sequence encoded as integers.

    Returns:
    - str: The generated punchline.
    """
    # print("input_sequence",input_sequence)
    # print("input_sequence type",type(input_sequence))
    # print("input_sequence shape",input_sequence.shape)
    humor_states = encoder_model.predict(input_sequence)
    punchline_sequence = np.zeros((1, 1))
    punchline_sequence[0, 0] = vocab_dict['<SOS>']
    joke = ''
    laugh_limit = False

    while not laugh_limit:
        dec_output, h, c = decoder_model.predict([punchline_sequence] + humor_states)
        dec_input1 = dense(dec_output)
        sampled_joke_index=np.argmax(dec_input1[0,-1,:])
        joke_word = inv_vocab[sampled_joke_index] + ' '
        if joke_word != '<EOS> ':
            joke += joke_word
        if joke_word == '<EOS> ' or len(joke.split()) > MAX_SEQUENCE_LENGTH:
            laugh_limit = True
        punchline_sequence = np.zeros((1, 1))
        punchline_sequence[0, 0] = sampled_joke_index
        humor_states = [h, c]

    joke = joke.replace("<PAD>", "")
    return joke

while True:
    user_prompt = input("You: ")
    if user_prompt == 'exit':
        break
```

```python
    processed_padded_prompt = preprocess_input(user_prompt)
    # print("padded user prompt",processed_padded_prompt)
    punchline = generate_punch_line(processed_padded_prompt)
    print("JestMaster: ", punchline)
    print("========================================")
```

```
****************************************
*          Welcome to JestMaster ʕ•ᴥ•ʔ!          *
*         Where Every Line is a Punchline       *
****************************************
You: hello
1/1 [==============================] - 0s 347ms/step
1/1 [==============================] - 0s 347ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
JestMaster:  hello there im jestmaster step right up to jestmaster your goto for witty banter and hilarious punchlines get ready to lol as i dish out funny answers guaranteed
========================================
You: how's it going?
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
```

```
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
JestMaster:  im doing well how about you
==========================================
You: who created you?
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
JestMaster:  i emerged from the digital ether a quirky creation by caroline p
==========================================
You: what can you do?
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
```

```
1/1 [==============================] - 0s 19ms/step
JestMaster:  ah the ageold question im here to sprinkle a dash of humor into your day with punchy punchlines and witty banter so as
k away and brace yourself for
==========================================
You: do you like corny jokes?
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
JestMaster:  its not they both ill be in the breasts all this one of you
==========================================
You: do you guys like corny jokes?
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
JestMaster:  because i have some absolutely amaizeing ones
==========================================
You: are you a moment of inertia?
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
JestMaster:  because youre mr squared
==========================================
You: Bye
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
JestMaster:  see you again
==========================================
```

```
==========================================
You: exit
```

IMPLEMENTED CHATBOT WITH USER MODELS

In [50]:

```
import spacy
NER = spacy.load("en_core_web_sm")
```

In [59]:

```
##### USER MODEL txt files

google_drive_folder = '/content/drive/MyDrive/NLP_PROJECT2/user_models'
%cd /content/drive/MyDrive/NLP_PROJECT2/user_models

import numpy as np
from keras.preprocessing.sequence import pad_sequences
import os
import json

print("*****************************************")
print("*          Welcome to JestMaster ʕ•ᴥ•ʔ!         *")
print("*         Where Every Line is a Punchline       *")
print("*****************************************")

def preprocess_input(user_input):
    """
    Preprocesses the user input by cleaning, tokenizing, encoding, and padding it for model input.

    Args:
    - user_input (str): The raw user input text.

    Returns:
    - numpy.ndarray: The preprocessed and padded input sequence as a 2D array.
    """
    preprocessed_input = clean_sent(user_input)
    # print("preprocessed_input1",preprocessed_input)
    preprocessed_input=[preprocessed_input]
    # print("preprocessed_input2",preprocessed_input)
    encoded_user_input=[]
    for inp in preprocessed_input:
        x=[]
        for i in inp.split():
            try:
                x.append(vocab_dict[i])
            except:
                x.append(vocab_dict['<OUT>'])
```

```python
        encoded_user_input.append(x)
    # print("encoded_user_input",encoded_user_input)
    padded_user_input = pad_sequences(encoded_user_input, maxlen=MAX_SEQUENCE_LENGTH, padding='post')
    # print("padded_user_input",padded_user_input)
    # encoded_input = [vocab_dict.get(word, vocab_dict['<OUT>']) for word in preprocessed_input.split()]
    return padded_user_input

def generate_punch_line(input_sequence):
    """
    Generate a punchline based on the input sequence using the trained encoder-decoder model.

    Args:
    - input_sequence (numpy.ndarray): The preprocessed input sequence encoded as integers.

    Returns:
    - str: The generated punchline.
    """
    # print("input_sequence",input_sequence)
    # print("input_sequence type",type(input_sequence))
    # print("input_sequence shape",input_sequence.shape)
    humor_states = encoder_model.predict(input_sequence)
    punchline_sequence = np.zeros((1, 1))
    punchline_sequence[0, 0] = vocab_dict['<SOS>']
    joke = ''
    laugh_limit = False

    while not laugh_limit:
        dec_output, h, c = decoder_model.predict([punchline_sequence] + humor_states)
        dec_input1 = dense(dec_output)
        sampled_joke_index=np.argmax(dec_input1[0,-1,:])
        joke_word = inv_vocab[sampled_joke_index] + ' '
        if joke_word != '<EOS> ':
            joke += joke_word
        if joke_word == '<EOS> ' or len(joke.split()) > MAX_SEQUENCE_LENGTH:
            laugh_limit = True
        punchline_sequence = np.zeros((1, 1))
        punchline_sequence[0, 0] = sampled_joke_index
        humor_states = [h, c]

    joke = joke.replace("<PAD>", "")
    return joke

def get_user_name(name_input_text):
    """
        Extract the user's name from the input text.
        :param name_input_text: The input text containing the user's name
        :return: The extracted name as a string
    """
    name=""
    tagged_input=NER(name_input_text)
    for item in tagged_input:
```

```
        if item.pos=="PROPN":
            name+=item.text+""
    name=name[:-1]
    return name

def create_user_model(user_name):
    """
        Create a user model file for the given user name.
        :param user_name: The user's name
        :return: A welcome message as a string
    """
    user_model_file = os.path.join(google_drive_folder, f"{user_name}.txt")
    with open(user_model_file, "w") as f:
        f.write(f"Name: {user_name}\n")
    return f"JestMaster: Nice to meet you, {user_name}. Welcome to JestMaster ʕ•ᴥ•ʔ! As your personal joke assistant, I specialize
in delivering hilarious one-liners and punchlines. Just like a seasoned comedian, I'm here to make you laugh until your sides ache!
But before we dive into the comedy ocean, I'd love to understand your sense of humor better. So, tell me, what tickles your funny b
one? Whether it's witty puns, clever wordplay, or quirky anecdotes, I'm here to tailor the humor to your tastes! Let's embark on a
laughter-filled journey together!"


def check_user_model(user_name):
    """
        Check if a user model file exists for the given user name.
        :param user_name: The user's name
        :return: 1 if the user model exists, -1 otherwise
    """
    user_model_file = os.path.join(google_drive_folder, f"{user_name}.txt")
    if os.path.exists(user_model_file):
        return 1
    else:
        return -1

def update_user_information(user_name, key, value):
    """
        Update user information in the user model file.
        :param user_name: The user's name
        :param key: The key to update
        :param value: The value to update
    """
    user_model_file = os.path.join(google_drive_folder, f"{user_name}.txt")
    with open(user_model_file, "a") as f:
        f.write(f"{key}: {value}\n")


def get_personalized_remark(user_name):
    """
        Provide a personalized remark based on the user's name and preferences.
        :param user_name: The user's name
        :return: A personalized remark as a string
    """
```

```
    user_model_file = os.path.join(google_drive_folder, f"{user_name}.txt")
    likes = []
    with open(user_model_file, "r") as f:
        for line in f:
            if line.startswith("Likes:"):
                likes = line.strip().split(": ")[1].split(", ")
    if likes:
        return f"JestMaster: I see that you like {', '.join(likes)}. So wait no more, give me a joke prompt of your choice for me t
o deliver a hilarious punch line XD"
    else:
        return f"JestMaster::I'm Jest Master, your personal joke assistant specializing in delivering hilarious one-liners and punc
hlines. So wait no more, give me a joke prompt for me to deliver a hilarious punch line XD "


def save_conversation(user_name, user_input, jest):
    """
    Saves the conversation between the user and JestMaster in a text file.

    Args:
    - user_name (str): The name of the user.
    - user_input (str): The input provided by the user.
    - jest (str): The response generated by JestMaster.

    Returns:
    - None
    """
    user_model_file = os.path.join(google_drive_folder, f"{user_name}.txt")
    with open(user_model_file, "a") as f:
        f.write(f"User: {user_input}\n")
        f.write(f"JestMaster: {jest}\n")


def main():
    user_name_input = input("JestMaster: I'm Jest Master, your personal joke assistant specializing in delivering hilarious one-lin
ers and punchlines. Before we get started, What's your name? ")
    name_result=get_user_name(user_name_input)
    if name_result == "":
        print("JestMaster: I'm unable to get your name from your input. Can you please tell me your name and just enter your name b
y itself?")
        name_result=input()
    #check if name already exists, if not create an entry
    check=check_user_model(name_result)
    if check==1:
      print(f"JestMaster: Welcome back, {name_result}!" )
    elif check==-1:
      welcome_message=create_user_model(name_result)
      print(welcome_message)
      likes = input("JestMaster: What are the kind of jokes you like? (Enter as comma-separated list): ").split(",")
      update_user_information(name_result, "Likes", ", ".join([like.strip() for like in likes]))

      # Update user's dislikes
```

```
      dislikes = input("JestMaster: What are the kind of jokes you dislike?? (Enter as comma-separated list) ").split(",")
      update_user_information(name_result, "Dislikes", ", ".join([dislike.strip() for dislike in dislikes]))
    print(get_personalized_remark(name_result))

    while True:
        user_prompt = input("You: ")
        # print(user_prompt)
        if user_prompt == 'exit':
            break
        processed_padded_prompt = preprocess_input(user_prompt)
        # print("padded user prompt",processed_padded_prompt)
        # print("padded user prompt",processed_padded_prompt)
        punchline = generate_punch_line(processed_padded_prompt)
        print("JestMaster:",punchline)
        # print("JestMaster: ", punchline)
        save_conversation(name_result,user_prompt, punchline)
        print("==========================================")


if __name__ == "__main__":
  main()
```

```
/content/drive/MyDrive/NLP_PROJECT2/user_models
******************************************
*          Welcome to JestMaster ʕ•ᴥ•ʔ!         *
*         Where Every Line is a Punchline       *
******************************************
JestMaster: I'm Jest Master, your personal joke assistant specializing in delivering hilarious one-liners and punchlines. Before we
get started, What's your name? Rakshitha
JestMaster: I'm unable to get your name from your input. Can you please tell me your name and just enter your name by itself?
Rakshitha
JestMaster: Nice to meet you, Rakshitha. Welcome to JestMaster ʕ•ᴥ•ʔ! As your personal joke assistant, I specialize in delivering h
ilarious one-liners and punchlines. Just like a seasoned comedian, I'm here to make you laugh until your sides ache! But before we
dive into the comedy ocean, I'd love to understand your sense of humor better. So, tell me, what tickles your funny bone? Whether it
's witty puns, clever wordplay, or quirky anecdotes, I'm here to tailor the humor to your tastes! Let's embark on a laughter-filled
journey together!
JestMaster: What are the kind of jokes you like? (Enter as comma-separated list): dark jokes, puns
JestMaster: What are the kind of jokes you dislike?? (Enter as comma-separated list) dad jokes
JestMaster: I see that you like dark jokes, puns. So wait no more, give me a joke prompt of your choice for me to deliver a hilario
us punch line XD
You: hello
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
```

```
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
JestMaster: hello there im jestmaster step right up to jestmaster your goto for witty banter and hilarious punchlines get ready to
lol as i dish out funny answers guaranteed
==========================================
You: who created you?
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
JestMaster: i emerged from the digital ether a quirky creation by caroline p
==========================================
You: why did C++ look down at C?
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 21ms/step
```

```
1/1 [==============================] - 0s 21ms/step
JestMaster: it thought c lacked class ill see myself out
==========================================
You: Why did Hitler suicide?
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
JestMaster: he got the gas bill
==========================================
You: That was funny. Thanks. you're so funny
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
JestMaster: because the hardware store clerk was a 300 pound flaming homosexual and i was afraid to ask him some trouble hed ask to
read some
==========================================
You: ok bye
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
JestMaster: bye
==========================================
You: exit
```

In [ ]:

```

```