In [1]:

```
!gdown --id 1OurDQUtbWQacvT32HMqFL7vIUrSMllOp
```

```
Downloading...
From: https://drive.google.com/uc?id=1OurDQUtbWQacvT32HMqFL7vIUrSMllOp
To: /content/preprocessed_data.csv
100% 300k/300k [00:00<00:00, 43.8MB/s]
```

In [2]:

```
#Importing necessary libraries
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

In [3]:

```
df=pd.read_csv('preprocessed_data.csv')#reading the data into DataFrame
```

In [4]:

```
df.head(4)#displaying top 4 four data
```

Out[4]:

	Unnamed: 0	source	target
0	0	U wan me to "chop" seat 4 u nt?\n	Do you want me to reserve seat for you or not?\n
1	1	Yup. U reaching. We order some durian pastry a...	Yeap. You reaching? We ordered some Durian pas...
2	2	They become more ex oredi... Mine is like 25.....	They become more expensive already. Mine is li...
3	3	I'm thai. what do u do?\n	I'm Thai. What do you do?\n

In [5]:

```
def preprocess(x):#removing the last character
  x=x[:-1]
  return x
```

In [6]:

```
df['source']=df['source'].apply(preprocess)#preprocessing on source data
df['target']=df['target'].apply(preprocess)#perprocessing on target data
```

In [7]:

```
df=df[['source','target']]
df.head()
```

Out[7]:

	source	target
0	U wan me to "chop" seat 4 u nt?	Do you want me to reserve seat for you or not?
1	Yup. U reaching. We order some durian pastry a...	Yeap. You reaching? We ordered some Durian pas...
2	They become more ex oredi... Mine is like 25.....	They become more expensive already. Mine is li...
3	I'm thai. what do u do?	I'm Thai. What do you do?
4	Hi! How did your week go? Haven heard from you...	Hi! How did your week go? Haven't heard from y...

In [8]:

```
df.shape#shape of the data
```

Out[8]:

```
(2000, 2)
```

In [9]:

```
df=df[df['source'].apply(len)<170]#removing source datapoints having length greater than equal to 170
df=df[df['target'].apply(len)<200]#removing target datapoints having length greater than equal to 200
```

In [10]:

```
df.shape#shape of the data
```

Out[10]:

```
(1990, 2)
```

In [11]:

```
from sklearn.model_selection import train_test_split
X=df['source']
y=df['target']
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.01)#splitting the data
print(X_train.shape)
```

```
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(1970,)
(20,)
(1970,)
(20,)
```

In [57]:

```
X_train.to_csv('X_train.csv')
y_train.to_csv('y_train.csv')
X_test.to_csv('X_test.csv')
y_test.to_csv('y_test.csv')
```

Target:

In [13]:

```
target_tokenizer= Tokenizer()#tokenization on target
target_tokenizer.fit_on_texts(y_train)#fitting on ytrain
target_vocab_size= len(target_tokenizer.word_index) + 1#target vocab size
print(len(target_tokenizer.word_index))
```

```
3032
```

In [14]:

```
target_encoded_docs_train = target_tokenizer.texts_to_sequences(y_train)#converting text to integers
target_encoded_docs_test = target_tokenizer.texts_to_sequences(y_test)#converting text to integers
```

In [15]:

```
target_padded_docs_train = pad_sequences(target_encoded_docs_train,padding='post')#padding to maxlength
```

In [16]:

```
target_padded_docs_train.shape
```

Out[16]:

```
(1970, 43)
```

In [17]:

```
target_padded_docs_test = pad_sequences(target_encoded_docs_test,maxlen=target_padded_docs_train.shape[1]
```

In [18]:

```
target_padded_docs_test.shape
```

Out[18]:

```
(20, 43)
```

Source:

In [19]:

```
source_tokenizer= Tokenizer()#tokenization on source
source_tokenizer.fit_on_texts(X_train)#fitting to X_train
source_vocab_size= len(source_tokenizer.word_index) + 1#source vocab size
print(len(source_tokenizer.word_index))
```

```
3703
```

In [20]:

```
source_encoded_docs_train = source_tokenizer.texts_to_sequences(X_train)#converting text to sequence
source_encoded_docs_test = source_tokenizer.texts_to_sequences(X_test)#converting text to sequence
```

In [21]:

```
source_padded_docs_train = pad_sequences(source_encoded_docs_train,maxlen=target_padded_docs_train.shape[
```

In [22]:

```
source_padded_docs_train.shape
```

Out[22]:

```
(1970, 43)
```

In [23]:

```
source_padded_docs_test = pad_sequences(source_encoded_docs_test,maxlen=target_padded_docs_train.shape[1]
```

In [24]:

```
source_padded_docs_test.shape
```

Out[24]:

```
(20, 43)
```

In [25]:

```
#we are reshaping the dataset because the sparese_categorical_crossentropy requires data to be three dim

target_padded_docs_train=target_padded_docs_train.reshape((*target_padded_docs_train.shape,1))
target_padded_docs_test=target_padded_docs_test.reshape((*target_padded_docs_test.shape,1))
```

In [26]:

```
print(target_padded_docs_train.shape)
print(target_padded_docs_test.shape)
```

```
(1970, 43, 1)
(20, 43, 1)
```

In [27]:

```
#we are reshaping the dataset because the sparese_categorical_crossentropy requires data to be three dim

source_padded_docs_train=source_padded_docs_train.reshape((*source_padded_docs_train.shape,1))
source_padded_docs_test=source_padded_docs_test.reshape((*source_padded_docs_test.shape,1))
```

In [28]:

```
print(source_padded_docs_train.shape)
print(source_padded_docs_test.shape)
```

```
(1970, 43, 1)
(20, 43, 1)
```

In [59]:

```
import pandas as pd
pd.DataFrame(source_encoded_docs_train).to_csv("source_encoded_docs_train.csv")
pd.DataFrame(source_encoded_docs_test).to_csv("source_encoded_docs_test.csv")
pd.DataFrame(target_encoded_docs_train).to_csv("target_encoded_docs_train.csv")
pd.DataFrame(target_encoded_docs_test).to_csv("target_encoded_docs_test.csv")
```

Model1:

In [32]:

```
input=tf.keras.layers.Input(shape=(43,))
embed=tf.keras.layers.Embedding(source_vocab_size,512, input_length=source_padded_docs_train.shape[1])(in
lstm1=tf.keras.layers.LSTM(128, return_sequences=True)(embed)
dense=tf.keras.layers.TimeDistributed(tf.keras.layers.Dense(512, activation='relu'))(lstm1)
drop=tf.keras.layers.Dropout(0.5)(dense)
output=tf.keras.layers.TimeDistributed(tf.keras.layers.Dense(target_vocab_size, activation='softmax'))(dr
model=tf.keras.models.Model(inputs=input,outputs=output)
model.summary()
```

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 43)]              0
_________________________________________________________________
embedding_1 (Embedding)      (None, 43, 512)           1896448
_________________________________________________________________
lstm_1 (LSTM)                (None, 43, 128)           328192
_________________________________________________________________
time_distributed_2 (TimeDist (None, 43, 512)           66048
_________________________________________________________________
dropout_1 (Dropout)          (None, 43, 512)           0
_________________________________________________________________
time_distributed_3 (TimeDist (None, 43, 3033)          1555929
=================================================================
Total params: 3,846,617
Trainable params: 3,846,617
Non-trainable params: 0
_________________________________________________________________
```

In [33]:

```
# Compile model
model.compile(optimizer=tf.keras.optimizers.Adam(0.01),
              loss='sparse_categorical_crossentropy',metrics=['accuracy'])
```

In [34]:

```
model.fit(source_padded_docs_train,target_padded_docs_train,batch_size=1024,epochs=50,
          validation_data=(source_padded_docs_test,target_padded_docs_test))
```

```
Epoch 1/50
2/2 [==============================] - 2s 682ms/step - loss: 7.6636 - accuracy: 0.3253 - val_loss:
2.8829 - val_accuracy: 0.6814
Epoch 2/50
2/2 [==============================] - 1s 296ms/step - loss: 3.3623 - accuracy: 0.6737 - val_loss:
3.7915 - val_accuracy: 0.6814
Epoch 3/50
2/2 [==============================] - 1s 298ms/step - loss: 3.5440 - accuracy: 0.6738 - val_loss:
3.1268 - val_accuracy: 0.6849
Epoch 4/50
2/2 [==============================] - 1s 295ms/step - loss: 2.9334 - accuracy: 0.6277 - val_loss:
2.1335 - val_accuracy: 0.6907
Epoch 5/50
2/2 [==============================] - 1s 297ms/step - loss: 2.2468 - accuracy: 0.6814 - val_loss:
2.0976 - val_accuracy: 0.6826
Epoch 6/50
```

```
2/2 [==============================] - 1s 299ms/step - loss: 2.2337 - accuracy: 0.6770 - val_loss:
2.0702 - val_accuracy: 0.6826
Epoch 7/50
2/2 [==============================] - 1s 295ms/step - loss: 2.1996 - accuracy: 0.6778 - val_loss:
2.0878 - val_accuracy: 0.6895
Epoch 8/50
2/2 [==============================] - 1s 298ms/step - loss: 2.1723 - accuracy: 0.6818 - val_loss:
2.0516 - val_accuracy: 0.6895
Epoch 9/50
2/2 [==============================] - 1s 295ms/step - loss: 2.1341 - accuracy: 0.6830 - val_loss:
2.0055 - val_accuracy: 0.6884
Epoch 10/50
2/2 [==============================] - 1s 297ms/step - loss: 2.1038 - accuracy: 0.6839 - val_loss:
1.9801 - val_accuracy: 0.6907
Epoch 11/50
2/2 [==============================] - 1s 296ms/step - loss: 2.0731 - accuracy: 0.6865 - val_loss:
1.9737 - val_accuracy: 0.6942
Epoch 12/50
2/2 [==============================] - 1s 298ms/step - loss: 2.0512 - accuracy: 0.6876 - val_loss:
1.9464 - val_accuracy: 0.6907
Epoch 13/50
2/2 [==============================] - 1s 297ms/step - loss: 2.0213 - accuracy: 0.6886 - val_loss:
1.9211 - val_accuracy: 0.6942
Epoch 14/50
2/2 [==============================] - 1s 301ms/step - loss: 1.9878 - accuracy: 0.6887 - val_loss:
1.9127 - val_accuracy: 0.6953
Epoch 15/50
2/2 [==============================] - 1s 298ms/step - loss: 1.9589 - accuracy: 0.6912 - val_loss:
1.8974 - val_accuracy: 0.6919
Epoch 16/50
2/2 [==============================] - 1s 293ms/step - loss: 1.9291 - accuracy: 0.6931 - val_loss:
1.8791 - val_accuracy: 0.6907
Epoch 17/50
2/2 [==============================] - 1s 297ms/step - loss: 1.8970 - accuracy: 0.6950 - val_loss:
1.8727 - val_accuracy: 0.6942
Epoch 18/50
2/2 [==============================] - 1s 298ms/step - loss: 1.8681 - accuracy: 0.6959 - val_loss:
1.8493 - val_accuracy: 0.6942
Epoch 19/50
2/2 [==============================] - 1s 299ms/step - loss: 1.8403 - accuracy: 0.6976 - val_loss:
1.8440 - val_accuracy: 0.6942
Epoch 20/50
2/2 [==============================] - 1s 298ms/step - loss: 1.8105 - accuracy: 0.6993 - val_loss:
1.8314 - val_accuracy: 0.6988
Epoch 21/50
2/2 [==============================] - 1s 299ms/step - loss: 1.7842 - accuracy: 0.7021 - val_loss:
1.8299 - val_accuracy: 0.7058
Epoch 22/50
2/2 [==============================] - 1s 297ms/step - loss: 1.7562 - accuracy: 0.7040 - val_loss:
1.8196 - val_accuracy: 0.7081
Epoch 23/50
2/2 [==============================] - 1s 298ms/step - loss: 1.7309 - accuracy: 0.7066 - val_loss:
1.8111 - val_accuracy: 0.7116
Epoch 24/50
2/2 [==============================] - 1s 301ms/step - loss: 1.7078 - accuracy: 0.7096 - val_loss:
1.8135 - val_accuracy: 0.7140
Epoch 25/50
2/2 [==============================] - 1s 304ms/step - loss: 1.6773 - accuracy: 0.7121 - val_loss:
1.8021 - val_accuracy: 0.7151
Epoch 26/50
2/2 [==============================] - 1s 300ms/step - loss: 1.6516 - accuracy: 0.7151 - val_loss:
1.7920 - val_accuracy: 0.7174
Epoch 27/50
2/2 [==============================] - 1s 299ms/step - loss: 1.6225 - accuracy: 0.7181 - val_loss:
1.8150 - val_accuracy: 0.7186
Epoch 28/50
2/2 [==============================] - 1s 301ms/step - loss: 1.5917 - accuracy: 0.7226 - val_loss:
1.7871 - val_accuracy: 0.7256
Epoch 29/50
2/2 [==============================] - 1s 304ms/step - loss: 1.5617 - accuracy: 0.7263 - val_loss:
1.7878 - val_accuracy: 0.7267
Epoch 30/50
2/2 [==============================] - 1s 299ms/step - loss: 1.5382 - accuracy: 0.7308 - val_loss:
1.7456 - val_accuracy: 0.7326
Epoch 31/50
2/2 [==============================] - 1s 301ms/step - loss: 1.5133 - accuracy: 0.7338 - val_loss:
1.8043 - val_accuracy: 0.7419
```

```
Epoch 32/50
2/2 [==============================] - 1s 298ms/step - loss: 1.4798 - accuracy: 0.7402 - val_loss:
1.7593 - val_accuracy: 0.7384
Epoch 33/50
2/2 [==============================] - 1s 302ms/step - loss: 1.4467 - accuracy: 0.7436 - val_loss:
1.7910 - val_accuracy: 0.7395
Epoch 34/50
2/2 [==============================] - 1s 308ms/step - loss: 1.4099 - accuracy: 0.7485 - val_loss:
1.7919 - val_accuracy: 0.7500
Epoch 35/50
2/2 [==============================] - 1s 296ms/step - loss: 1.3756 - accuracy: 0.7540 - val_loss:
1.7466 - val_accuracy: 0.7465
Epoch 36/50
2/2 [==============================] - 1s 299ms/step - loss: 1.3438 - accuracy: 0.7564 - val_loss:
1.8333 - val_accuracy: 0.7651
Epoch 37/50
2/2 [==============================] - 1s 299ms/step - loss: 1.3087 - accuracy: 0.7656 - val_loss:
1.7408 - val_accuracy: 0.7605
Epoch 38/50
2/2 [==============================] - 1s 298ms/step - loss: 1.2719 - accuracy: 0.7681 - val_loss:
1.7982 - val_accuracy: 0.7721
Epoch 39/50
2/2 [==============================] - 1s 303ms/step - loss: 1.2379 - accuracy: 0.7742 - val_loss:
1.7827 - val_accuracy: 0.7733
Epoch 40/50
2/2 [==============================] - 1s 299ms/step - loss: 1.2018 - accuracy: 0.7811 - val_loss:
1.7217 - val_accuracy: 0.7686
Epoch 41/50
2/2 [==============================] - 1s 298ms/step - loss: 1.1695 - accuracy: 0.7825 - val_loss:
1.8738 - val_accuracy: 0.7767
Epoch 42/50
2/2 [==============================] - 1s 299ms/step - loss: 1.1507 - accuracy: 0.7885 - val_loss:
1.7565 - val_accuracy: 0.7756
Epoch 43/50
2/2 [==============================] - 1s 296ms/step - loss: 1.1208 - accuracy: 0.7932 - val_loss:
1.7998 - val_accuracy: 0.7756
Epoch 44/50
2/2 [==============================] - 1s 297ms/step - loss: 1.0840 - accuracy: 0.7979 - val_loss:
1.8774 - val_accuracy: 0.7791
Epoch 45/50
2/2 [==============================] - 1s 300ms/step - loss: 1.0505 - accuracy: 0.8041 - val_loss:
1.7933 - val_accuracy: 0.7767
Epoch 46/50
2/2 [==============================] - 1s 297ms/step - loss: 1.0186 - accuracy: 0.8041 - val_loss:
1.9076 - val_accuracy: 0.7872
Epoch 47/50
2/2 [==============================] - 1s 298ms/step - loss: 0.9887 - accuracy: 0.8124 - val_loss:
1.7677 - val_accuracy: 0.7791
Epoch 48/50
2/2 [==============================] - 1s 297ms/step - loss: 0.9538 - accuracy: 0.8149 - val_loss:
1.9284 - val_accuracy: 0.7907
Epoch 49/50
2/2 [==============================] - 1s 303ms/step - loss: 0.9254 - accuracy: 0.8213 - val_loss:
1.7607 - val_accuracy: 0.7837
Epoch 50/50
2/2 [==============================] - 1s 298ms/step - loss: 0.8998 - accuracy: 0.8253 - val_loss:
1.9367 - val_accuracy: 0.7930
```

Out[34]:

```
<tensorflow.python.keras.callbacks.History at 0x7fdec6dfbe90>
```

In [35]:

```
x=model.predict(source_padded_docs_test[:1])[0]
```

In [36]:

```
#https://machinelearningmastery.com
index_to_words = {id: word for word, id in target_tokenizer.word_index.items()}
index_to_words[0] = '<PAD>'

' '.join([index_to_words[prediction] for prediction in np.argmax(x, 1)])
```

Out[36]:

```
'hey i am still having stuff <PAD> if you reach <PAD> <PAD> <PAD> help it and me you to <PAD> <PAD> <PAD>
<PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PA
D> <PAD> <PAD> <PAD>'
```

In [37]:

```
print(y_test[:1])
```

```
1866    I am still having breakfast. If you reach ther...
Name: target, dtype: object
```

In [38]:

```
X_test[:1]
```

Out[38]:

```
1866    Hey i am still having breakfast eh. If you rea...
Name: source, dtype: object
```

In [41]:

```
def prediction(x):

  index_to_words = {id: word for word, id in target_tokenizer.word_index.items()}
  index_to_words[0] = '<PAD>'

  y=' '.join([index_to_words[prediction] for prediction in np.argmax(x, 1)])
  return y
for i in range(20):
  print("Input text: ")
  a=list(X_test[i:i+1])
  print(a[0])

  print("Actual Output: ")
  b=list(y_test[i:i+1])
  print(b[0])

  print("Predicted Output: ")
  x=model.predict(source_padded_docs_test[i:i+1])
  y=prediction(x[0])
  y=y.split(' ')
  y_lst=[]
  for i in y:
    if i=='<PAD>':
      continue
    else:
      y_lst.append(i)
  print(' '.join(y_lst))
  print('>'*180)
```

```
Input text:
Hey i am still having breakfast eh. If you reach there first can help rebecca and me chope seats?
Actual Output:
I am still having breakfast. If you reach there first can you help me and Rebecca reserve seats?
Predicted Output:
hey i am still having stuff if you reach help it and me you to
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Huh ü take then how i take bus later... Inside got money a not...
Actual Output:
If you take then how I take bus later? Inside got money or not?
Predicted Output:
huh you take then how i take bus me have your a not
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Hi neva worry bout da truth coz the truth will lead me 2 ur heart. It's the least a unique person like u
deserve. Sleep tight or morning
Actual Output:
Hi, never worry about the truth because the truth will lead me to your heart. It's the least that a uniqu
e person like you deserve. Sleep tight or morning.
Predicted Output:
hi never worry about the because the will me to your one very the hang a you kb can you inform sleep unr
estricted or the
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Take so long
Actual Output:
Take so long.
Predicted Output:
take so long
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Hey where r ü im here liao
Actual Output:
Hey, where are you? I'm here.
```

```
Predicted Output:
hey where are you i'm here already
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Hi everyone hows ur day ?
Actual Output:
Hi everyone, how's your day?
Predicted Output:
hi everyone how's your day
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Haha- if no need make up ñ near my wkplace ñ not wk too late.can consider.tt is if ü can find such a pla
ce.ay,abt a mth ago she say she wk ere la. Hee-
Actual Output:
Haha. If no need to make up and near my workplace and does not work too late. Can consider. That is if y
ou can find such a place. AY, about a month ago, she said she worked there.
Predicted Output:
haha if no need make up and near my workplace and not new too late can that is if you can find a place s
orry about a month ago that say she next is but
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
How i noe... Last time tis one is on offer wat...
Actual Output:
How I know. Last time this one is on offer.
Predicted Output:
how i know very time this one is for it what
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
I reached already
Actual Output:
I reached already.
Predicted Output:
i reached already
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Haiyoh... It was so crowded... We didnt buy anything... Haha... Lots of pple in town. So mon we go
facial with ü then go shopping?
Actual Output:
Ouch. It was so crowded. We didn't buy anything. Haha. There are lots of people in town. So Monday we go
facial with you then go shopping?
Predicted Output:
thing was so crowded we didn't buy anything haha a of people in town so monday i go with you then go to
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
HI MERINA NICE 2 CHAT WITH U. UR HP NO PLS. WHAT IS UR RACE?
Actual Output:
Hi Merina. It's nice to chat with you. Your hand phone number please. What is your race?
Predicted Output:
hi merina nice to chat with you your handphone no please what is your girl
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Hmmm.... After my drivin den free lor... Y?
Actual Output:
After my driving then I will be free. Why?
Predicted Output:
hmm after my is late free you
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Erm anything lor...Can bring tmr? Thx =)
Actual Output:
Can anything be brought tomorrow? Thanks.
Predicted Output:
i anything can bring tomorrow that's
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Okay... they arent open on public holidays
Actual Output:
Okay. They aren't open on public holidays.
```

```
Predicted Output:
okay they accepted open on games
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Haha... I'm carrying a broom with me so really paiseh to walk into lecture with it. I'm coming straight
from home mah... Cya later then.
Actual Output:
Haha. I'm carrying a broom with me. So I'm really sorry to walk into lecture with it. I'm coming
straight from home. See you later then.
Predicted Output:
haha i'm a with me so really sorry to to into another with it i'm coming password from home i'm see you
then
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Hmmm.... I'm watchin w my frens oredi... Paiseh...
Actual Output:
Hmm. I'm watching with my friends already. It's embarrassing.
Predicted Output:
hmm i'm watching with my friends
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Hey... Ü 've got driving today? my driving at 240.
Actual Output:
Hey. You have got driving today? My driving is at 2:40.
Predicted Output:
hey you got driving today my i at at
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
then it can moisturise our skin. and rub in circular motion. u wash face, tone,then put a bit of jelly a
nd cream onto ur hand,and tap it on your face,
Actual Output:
Then it can moisturise our skin and rub in circular motion. You wash face, tone, then put a bit of jelly
and cream onto your hand, and tap it on your face.
Predicted Output:
then it can our skin and in you haircut be then a a bit a of few for for and it on your be
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
I'm pubbin now, gee, cant go online...After my drivin ah, hmmm, den where ur meetin....
Actual Output:
I'm in pub now. I can't go online. After my driving, then where are you meeting?
Predicted Output:
i'm to the i go go to my i i i is you
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Haha... Not accurate right....
Actual Output:
Haha. Not accurate, right?
Predicted Output:
haha not right
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

In [44]:

```
import nltk.translate.bleu_score as bleu

bleu_score=[]
for i in range(20):
  b=list(y_test[i:i+1])
  x=model.predict(source_padded_docs_test[i:i+1])
  y=prediction(x[0])
  y=y.split(' ')
  y_lst=[]
  for i in y:
    if i=='<PAD>':
      continue
    else:
      y_lst.append(i)
  bleu_score.append(bleu.sentence_bleu([b[0].split(),],y_lst))
print(bleu_score)
print("The Average Bleu Score is: ",sum(bleu_score)/20)
```

```
/usr/local/lib/python3.7/dist-packages/nltk/translate/bleu_score.py:490: UserWarning:
Corpus/Sentence contains 0 counts of 4-gram overlaps.
BLEU scores might be undesirable; use SmoothingFunction().
  warnings.warn(_msg)
/usr/local/lib/python3.7/dist-packages/nltk/translate/bleu_score.py:490: UserWarning:
Corpus/Sentence contains 0 counts of 2-gram overlaps.
BLEU scores might be undesirable; use SmoothingFunction().
  warnings.warn(_msg)
/usr/local/lib/python3.7/dist-packages/nltk/translate/bleu_score.py:490: UserWarning:
Corpus/Sentence contains 0 counts of 3-gram overlaps.
BLEU scores might be undesirable; use SmoothingFunction().
  warnings.warn(_msg)
[0.2581911684267368, 0.21294807603873017, 0.14463972129203845, 0.7598356856515925, 0.4671379777282001,
0.5623413251903491, 0.30207246566144663, 0.23462350320528, 0.7598356856515925, 0.15909672318073625, 0.171
22548504687662, 0.46199933699457096, 0.6389431042462724, 0.43012508513132625, 0.1828175732238544,
0.3081980909598119, 0.36177396082048563, 0.24573784957585945, 0.5502659908318907, 0]
The Average Bleu Score is:  0.3605904404428826
```

Model2:

In [49]:

```
input=tf.keras.layers.Input(shape=(43,))
embed=tf.keras.layers.Embedding(source_vocab_size,512, input_length=source_padded_docs_train.shape[1])(in
lstm1=tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(100, return_sequences=True))(embed)
output=tf.keras.layers.TimeDistributed(tf.keras.layers.Dense(target_vocab_size, activation='softmax'))(ls
model=tf.keras.models.Model(inputs=input,outputs=output)
model.summary()
```

```
Model: "model_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_4 (InputLayer)         [(None, 43)]              0
_________________________________________________________________
embedding_3 (Embedding)      (None, 43, 512)           1896448
_________________________________________________________________
bidirectional_1 (Bidirection (None, 43, 200)           490400
_________________________________________________________________
time_distributed_5 (TimeDist (None, 43, 3033)          609633
=================================================================
Total params: 2,996,481
Trainable params: 2,996,481
Non-trainable params: 0
_________________________________________________________________
```

In [50]:

```
# Compile model
model.compile(optimizer=tf.keras.optimizers.Adam(0.01),
              loss='sparse_categorical_crossentropy',metrics=['accuracy'])
```

In [51]:

```
model.fit(source_padded_docs_train,target_padded_docs_train,batch_size=1024,epochs=50,
          validation_data=(source_padded_docs_test,target_padded_docs_test))
```

```
Epoch 1/50
2/2 [==============================] - 4s 865ms/step - loss: 7.6696 - accuracy: 0.3222 - val_loss:
5.3397 - val_accuracy: 0.6814
Epoch 2/50
2/2 [==============================] - 1s 279ms/step - loss: 4.5154 - accuracy: 0.6737 - val_loss:
2.4947 - val_accuracy: 0.6814
Epoch 3/50
2/2 [==============================] - 1s 283ms/step - loss: 2.7089 - accuracy: 0.6737 - val_loss:
2.7940 - val_accuracy: 0.6814
Epoch 4/50
2/2 [==============================] - 1s 289ms/step - loss: 2.9144 - accuracy: 0.6737 - val_loss:
2.5952 - val_accuracy: 0.6814
Epoch 5/50
2/2 [==============================] - 1s 282ms/step - loss: 2.7359 - accuracy: 0.6742 - val_loss:
2.3767 - val_accuracy: 0.6802
Epoch 6/50
2/2 [==============================] - 1s 284ms/step - loss: 2.4859 - accuracy: 0.6713 - val_loss:
2.1571 - val_accuracy: 0.6767
Epoch 7/50
2/2 [==============================] - 1s 284ms/step - loss: 2.2431 - accuracy: 0.6746 - val_loss:
2.0513 - val_accuracy: 0.6814
Epoch 8/50
2/2 [==============================] - 1s 284ms/step - loss: 2.1384 - accuracy: 0.6804 - val_loss:
2.0430 - val_accuracy: 0.6907
Epoch 9/50
```

```
2/2 [==============================] - 1s 279ms/step - loss: 2.1238 - accuracy: 0.6848 - val_loss:
2.1171 - val_accuracy: 0.6872
Epoch 10/50
2/2 [==============================] - 1s 282ms/step - loss: 2.1175 - accuracy: 0.6847 - val_loss:
2.0871 - val_accuracy: 0.6872
Epoch 11/50
2/2 [==============================] - 1s 285ms/step - loss: 2.1044 - accuracy: 0.6849 - val_loss:
2.0475 - val_accuracy: 0.6907
Epoch 12/50
2/2 [==============================] - 1s 284ms/step - loss: 2.0834 - accuracy: 0.6873 - val_loss:
1.9701 - val_accuracy: 0.6907
Epoch 13/50
2/2 [==============================] - 1s 284ms/step - loss: 2.0519 - accuracy: 0.6911 - val_loss:
1.9469 - val_accuracy: 0.6895
Epoch 14/50
2/2 [==============================] - 1s 283ms/step - loss: 2.0203 - accuracy: 0.6912 - val_loss:
1.9451 - val_accuracy: 0.6872
Epoch 15/50
2/2 [==============================] - 1s 286ms/step - loss: 1.9913 - accuracy: 0.6902 - val_loss:
1.9212 - val_accuracy: 0.6860
Epoch 16/50
2/2 [==============================] - 1s 287ms/step - loss: 1.9605 - accuracy: 0.6923 - val_loss:
1.8990 - val_accuracy: 0.6907
Epoch 17/50
2/2 [==============================] - 1s 285ms/step - loss: 1.9280 - accuracy: 0.6943 - val_loss:
1.8813 - val_accuracy: 0.6942
Epoch 18/50
2/2 [==============================] - 1s 285ms/step - loss: 1.8964 - accuracy: 0.6971 - val_loss:
1.8585 - val_accuracy: 0.6953
Epoch 19/50
2/2 [==============================] - 1s 286ms/step - loss: 1.8631 - accuracy: 0.7001 - val_loss:
1.8362 - val_accuracy: 0.7000
Epoch 20/50
2/2 [==============================] - 1s 292ms/step - loss: 1.8290 - accuracy: 0.7022 - val_loss:
1.8113 - val_accuracy: 0.7023
Epoch 21/50
2/2 [==============================] - 1s 285ms/step - loss: 1.7940 - accuracy: 0.7051 - val_loss:
1.7887 - val_accuracy: 0.7058
Epoch 22/50
2/2 [==============================] - 1s 285ms/step - loss: 1.7584 - accuracy: 0.7073 - val_loss:
1.7681 - val_accuracy: 0.7070
Epoch 23/50
2/2 [==============================] - 1s 281ms/step - loss: 1.7228 - accuracy: 0.7107 - val_loss:
1.7467 - val_accuracy: 0.7070
Epoch 24/50
2/2 [==============================] - 1s 288ms/step - loss: 1.6850 - accuracy: 0.7161 - val_loss:
1.7267 - val_accuracy: 0.7174
Epoch 25/50
2/2 [==============================] - 1s 289ms/step - loss: 1.6486 - accuracy: 0.7207 - val_loss:
1.7093 - val_accuracy: 0.7233
Epoch 26/50
2/2 [==============================] - 1s 281ms/step - loss: 1.6112 - accuracy: 0.7266 - val_loss:
1.6898 - val_accuracy: 0.7302
Epoch 27/50
2/2 [==============================] - 1s 281ms/step - loss: 1.5730 - accuracy: 0.7339 - val_loss:
1.6672 - val_accuracy: 0.7372
Epoch 28/50
2/2 [==============================] - 1s 288ms/step - loss: 1.5344 - accuracy: 0.7421 - val_loss:
1.6461 - val_accuracy: 0.7442
Epoch 29/50
2/2 [==============================] - 1s 280ms/step - loss: 1.4959 - accuracy: 0.7480 - val_loss:
1.6250 - val_accuracy: 0.7512
Epoch 30/50
2/2 [==============================] - 1s 285ms/step - loss: 1.4567 - accuracy: 0.7534 - val_loss:
1.6075 - val_accuracy: 0.7593
Epoch 31/50
2/2 [==============================] - 1s 285ms/step - loss: 1.4181 - accuracy: 0.7596 - val_loss:
1.5918 - val_accuracy: 0.7640
Epoch 32/50
2/2 [==============================] - 1s 283ms/step - loss: 1.3798 - accuracy: 0.7655 - val_loss:
1.5755 - val_accuracy: 0.7674
Epoch 33/50
2/2 [==============================] - 1s 287ms/step - loss: 1.3417 - accuracy: 0.7719 - val_loss:
1.5621 - val_accuracy: 0.7767
Epoch 34/50
2/2 [==============================] - 1s 286ms/step - loss: 1.3036 - accuracy: 0.7783 - val_loss:
1.5477 - val_accuracy: 0.7767
```

```
Epoch 35/50
2/2 [==============================] - 1s 290ms/step - loss: 1.2660 - accuracy: 0.7841 - val_loss:
1.5371 - val_accuracy: 0.7756
Epoch 36/50
2/2 [==============================] - 1s 289ms/step - loss: 1.2291 - accuracy: 0.7900 - val_loss:
1.5269 - val_accuracy: 0.7779
Epoch 37/50
2/2 [==============================] - 1s 282ms/step - loss: 1.1923 - accuracy: 0.7963 - val_loss:
1.5147 - val_accuracy: 0.7791
Epoch 38/50
2/2 [==============================] - 1s 293ms/step - loss: 1.1564 - accuracy: 0.8019 - val_loss:
1.5045 - val_accuracy: 0.7791
Epoch 39/50
2/2 [==============================] - 1s 281ms/step - loss: 1.1213 - accuracy: 0.8075 - val_loss:
1.4951 - val_accuracy: 0.7802
Epoch 40/50
2/2 [==============================] - 1s 291ms/step - loss: 1.0868 - accuracy: 0.8132 - val_loss:
1.4849 - val_accuracy: 0.7814
Epoch 41/50
2/2 [==============================] - 1s 284ms/step - loss: 1.0525 - accuracy: 0.8183 - val_loss:
1.4780 - val_accuracy: 0.7826
Epoch 42/50
2/2 [==============================] - 1s 282ms/step - loss: 1.0193 - accuracy: 0.8238 - val_loss:
1.4697 - val_accuracy: 0.7849
Epoch 43/50
2/2 [==============================] - 1s 283ms/step - loss: 0.9866 - accuracy: 0.8288 - val_loss:
1.4615 - val_accuracy: 0.7837
Epoch 44/50
2/2 [==============================] - 1s 285ms/step - loss: 0.9550 - accuracy: 0.8334 - val_loss:
1.4558 - val_accuracy: 0.7849
Epoch 45/50
2/2 [==============================] - 1s 285ms/step - loss: 0.9244 - accuracy: 0.8370 - val_loss:
1.4431 - val_accuracy: 0.7826
Epoch 46/50
2/2 [==============================] - 1s 283ms/step - loss: 0.8950 - accuracy: 0.8409 - val_loss:
1.4413 - val_accuracy: 0.7860
Epoch 47/50
2/2 [==============================] - 1s 284ms/step - loss: 0.8660 - accuracy: 0.8443 - val_loss:
1.4315 - val_accuracy: 0.7895
Epoch 48/50
2/2 [==============================] - 1s 289ms/step - loss: 0.8381 - accuracy: 0.8480 - val_loss:
1.4301 - val_accuracy: 0.7872
Epoch 49/50
2/2 [==============================] - 1s 287ms/step - loss: 0.8116 - accuracy: 0.8509 - val_loss:
1.4286 - val_accuracy: 0.7895
Epoch 50/50
2/2 [==============================] - 1s 294ms/step - loss: 0.7862 - accuracy: 0.8535 - val_loss:
1.4302 - val_accuracy: 0.7907
```

Out[51]:

```
<tensorflow.python.keras.callbacks.History at 0x7fdd6569ab90>
```

In [52]:

```
x=model.predict(source_padded_docs_test[7:8])[0]
```

In [53]:

```
index_to_words = {id: word for word, id in target_tokenizer.word_index.items()}
index_to_words[0] = '<PAD>'

' '.join([index_to_words[prediction] for prediction in np.argmax(x, 1)])
```

Out[53]:

```
"how i know last time this one is on father's what <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
<PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PA
D> <PAD> <PAD> <PAD> <PAD> <PAD>"
```

In [54]:

```
print(y_test[7:8])
```

```
128    How I know. Last time this one is on offer.
Name: target, dtype: object
```

In [55]:

```
X_test[7:8]
```

Out[55]:

```
128    How i noe... Last time tis one is on offer wat...
Name: source, dtype: object
```

In [60]:

```python
def prediction(x):

  index_to_words = {id: word for word, id in target_tokenizer.word_index.items()}
  index_to_words[0] = '<PAD>'

  y=' '.join([index_to_words[prediction] for prediction in np.argmax(x, 1)])
  return y
for i in range(20):
  print("Input text: ")
  a=list(X_test[i:i+1])
  print(a[0])

  print("Actual Output: ")
  b=list(y_test[i:i+1])
  print(b[0])

  print("Predicted Output: ")
  x=model.predict(source_padded_docs_test[i:i+1])
  y=prediction(x[0])
  y=y.split(' ')
  y_lst=[]
  for i in y:
    if i=='<PAD>':
      continue
    else:
      y_lst.append(i)
  print(' '.join(y_lst))
  print('>'*180)
```

```
Input text:
Hey i am still having breakfast eh. If you reach there first can help rebecca and me chope seats?
Actual Output:
I am still having breakfast. If you reach there first can you help me and Rebecca reserve seats?
Predicted Output:
hey i am still having work ask if you reach there first can help nap and me to and
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Huh ü take then how i take bus later... Inside got money a not...
Actual Output:
If you take then how I take bus later? Inside got money or not?
Predicted Output:
huh you take then how i to bus later you got money a or
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Hi neva worry bout da truth coz the truth will lead me 2 ur heart. It's the least a unique person like u
deserve. Sleep tight or morning
Actual Output:
Hi, never worry about the truth because the truth will lead me to your heart. It's the least that a uniqu
e person like you deserve. Sleep tight or morning.
Predicted Output:
hi i worry about the because the will me to your like it's the school a you person like you lend sleep e
njoy or number
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Take so long
Actual Output:
Take so long.
Predicted Output:
take so long
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Hey where r ü im here liao
Actual Output:
Hey, where are you? I'm here.
Predicted Output:
hey where are you are here you
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Hi everyone hows ur day ?
Actual Output:
Hi everyone, how's your day?
Predicted Output:
hi everyone how's your day
```

hi everyone how's your day
>>
>>>
Input text:
Haha- if no need make up ñ near my wkplace ñ not wk too late.can consider.tt is if ü can find such a pla
ce.ay,abt a mth ago she say she wk ere la. Hee-
Actual Output:
Haha. If no need to make up and near my workplace and does not work too late. Can consider. That is if y
ou can find such a place. AY, about a month ago, she said she worked there.
Predicted Output:
haha if no need make up and near my workplace and is house too late can just that is if you can find a a
of about a month ago she she she week a she
>>
>>>
Input text:
How i noe... Last time tis one is on offer wat...
Actual Output:
How I know. Last time this one is on offer.
Predicted Output:
how i know last time this one is on father's what
>>
>>>
Input text:
I reached already
Actual Output:
I reached already.
Predicted Output:
i reached already
>>
>>>
Input text:
Haiyoh... It was so crowded... We didnt buy anything... Haha... Lots of pple in town. So mon we go
facial with ü then go shopping?
Actual Output:
Ouch. It was so crowded. We didn't buy anything. Haha. There are lots of people in town. So Monday we go
facial with you then go shopping?
Predicted Output:
i it was so i'll we didn't buy anything haha today of people in town so monday we go with you then go go
to
>>
>>>
Input text:
HI MERINA NICE 2 CHAT WITH U. UR HP NO PLS. WHAT IS UR RACE?
Actual Output:
Hi Merina. It's nice to chat with you. Your hand phone number please. What is your race?
Predicted Output:
hi merina nice to chat with you your hand no please please is your say
>>
>>>
Input text:
Hmmm.... After my drivin den free lor... Y?
Actual Output:
After my driving then I will be free. Why?
Predicted Output:
hmm after my then free you haha
>>
>>>
Input text:
Erm anything lor...Can bring tmr? Thx =)
Actual Output:
Can anything be brought tomorrow? Thanks.
Predicted Output:
i anything anything can bring tomorrow tomorrow
>>
>>>
Input text:
Okay... they arent open on public holidays
Actual Output:
Okay. They aren't open on public holidays.
Predicted Output:
okay they nama open on public is
>>
>>>
Input text:
Haha... I'm carrying a broom with me so really paiseh to walk into lecture with it. I'm coming straight
from home mah... Cya later then.
Actual Output:

```
Haha. I'm carrying a broom with me. So I'm really sorry to walk into lecture with it. I'm coming
straight from home. See you later then.
Predicted Output:
haha i'm a with me so really sorry to for the lecture with it i'm coming rather from home i you later th
en
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Hmmm.... I'm watchin w my frens oredi... Paiseh...
Actual Output:
Hmm. I'm watching with my friends already. It's embarrassing.
Predicted Output:
hmm i'm watching with my friends already friends
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Hey... Ü 've got driving today? my driving at 240.
Actual Output:
Hey. You have got driving today? My driving is at 2:40.
Predicted Output:
hey you got have today my driving at 2
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
then it can moisturise our skin. and rub in circular motion. u wash face, tone,then put a bit of jelly a
nd cream onto ur hand,and tap it on your face,
Actual Output:
Then it can moisturise our skin and rub in circular motion. You wash face, tone, then put a bit of jelly
and cream onto your hand, and tap it on your face.
Predicted Output:
then it can our let's and in you having one be a a a a and quite a first and it on a one
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
I'm pubbin now, gee, cant go online...After my drivin ah, hmmm, den where ur meetin....
Actual Output:
I'm in pub now. I can't go online. After my driving, then where are you meeting?
Predicted Output:
i'm i now now can't then i after my later haha haha then where you you
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Input text:
Haha... Not accurate right....
Actual Output:
Haha. Not accurate, right?
Predicted Output:
haha not right
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

In [61]:

```python
import nltk.translate.bleu_score as bleu

bleu_score=[]
for i in range(20):
  b=list(y_test[i:i+1])
  x=model.predict(source_padded_docs_test[i:i+1])
  y=prediction(x[0])
  y=y.split(' ')
  y_lst=[]
  for i in y:
    if i=='<PAD>':
      continue
    else:
      y_lst.append(i)
  bleu_score.append(bleu.sentence_bleu([b[0].split(),],y_lst))
print(bleu_score)
print("The Average Bleu Score is: ",sum(bleu_score)/20)
```

```
/usr/local/lib/python3.7/dist-packages/nltk/translate/bleu_score.py:490: UserWarning:
Corpus/Sentence contains 0 counts of 4-gram overlaps.
BLEU scores might be undesirable; use SmoothingFunction().
  warnings.warn(_msg)
/usr/local/lib/python3.7/dist-packages/nltk/translate/bleu_score.py:490: UserWarning:
Corpus/Sentence contains 0 counts of 2-gram overlaps.
BLEU scores might be undesirable; use SmoothingFunction().
  warnings.warn(_msg)
/usr/local/lib/python3.7/dist-packages/nltk/translate/bleu_score.py:490: UserWarning:
Corpus/Sentence contains 0 counts of 3-gram overlaps.
BLEU scores might be undesirable; use SmoothingFunction().
  warnings.warn(_msg)
[0.27447938256311044, 0.22718709780542318, 0.3312232008397192, 0.7598356856515925, 0.4671379777282001,
0.5623413251903491, 0.30228791143745415, 0.3508439695638686, 0.7598356856515925, 0.16527975033438158,
0.17795291340072017, 0.5494128986804837, 0.6147881529512643, 0.4111336169005197, 0.22265046674893665,
0.3050975216056289, 0.6537993517025207, 0.3201518925576873, 0.37991784282579627, 0]
The Average Bleu Score is:  0.39176783220696243
```