In [5]:

```
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.layers import Embedding, LSTM, Dense
from tensorflow.keras.models import Sequential
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.optimizers import Adam
import pickle
import numpy as np
import os
```

In [9]:

```
from google.colab import files
file = files.upload()
file = open("dataset.txt","r")
lines = []

for i in file:
    lines.append(i)

print(lines[0])
print(lines[1000])
```

Choose files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving dataset.txt to dataset (4).txt
It was the best of times, it was the worst of times, it was the age of

“A likely thing, too!” replied the strong woman. “If it was ever
```

In [10]:

```
#Cleaning dataset
data = ""

for i in lines:
    data = ' '. join(lines)

data = data.replace('\n', '').replace('\r', '').replace('\ufeff', '')
data[:360]
```

Out[10]:

```
'It was the best of times, it was the worst of times, it was the age of wisdom, it
was the age of foolishness, it was the epoch of belief, it was the epoch of incred
ulity, it was the season of Light, it was the season of Darkness, it was the sprin
g of hope, it was the winter of despair, we had everything before us, we had nothi
ng before us, we were all going '
```

In [13]:

```
import string

translator = str.maketrans(string.punctuation, ' '*len(string.punctuation)) #map pu
new_data = data.translate(translator)

new_data[:150]
```

Out[13]:

```
'It was the best of times  it was the worst of times  it was the age of wisdom  it
was the age of foolishness  it was the epoch of belief  it was the e'
```

In [14]:

```
z = []

for i in data.split():
    if i not in z:
        z.append(i)

data = ' '.join(z)
data[:150]
```

Out[14]:

```
'It was the best of times, it worst age wisdom, foolishness, epoch belief, incredu
lity, season Light, Darkness, spring hope, winter despair, we had eve'
```

In [15]:

```python
#Tokenization
tokenizer = Tokenizer()
tokenizer.fit_on_texts([data])

# saving the tokenizer for predict function.
pickle.dump(tokenizer, open('tokenizer.pkl', 'wb'))

sequence_data = tokenizer.texts_to_sequences([data])[0]
sequence_data[:10]
```

Out[15]:

```
[5, 31, 6, 505, 52, 370, 5, 1141, 753, 3916]
```

In [16]:

```python
vocab_size = len(tokenizer.word_index) + 1
print(vocab_size)
```

```
10619
```

In [17]:

```python
sequences = []

for i in range(1, len(sequence_data)):
    words = sequence_data[i-1:i+1]
    sequences.append(words)

print("The Length of sequences are: ", len(sequences))
sequences = np.array(sequences)
sequences[:10]
```

Out[17]:

```
The Length of sequences are:  22558
array([[   5,   31],
       [  31,    6],
       [   6,  505],
       [ 505,   52],
       [  52,  370],
       [ 370,    5],
       [   5, 1141],
       [1141,  753],
       [ 753, 3916],
       [3916, 3917]])
```

In [18]:

```python
X = []
y = []

for i in sequences:
    X.append(i[0])
    y.append(i[1])

X = np.array(X)
y = np.array(y)
```

In [19]:

```python
print("The Data is: ", X[:5])
print("The responses are: ", y[:5])
```

```
The Data is:  [  5  31   6 505  52]
The responses are:  [ 31   6 505  52 370]
```

In [20]:

```python
y = to_categorical(y, num_classes=vocab_size)
y[:5]
```

Out[20]:

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]], dtype=float32)
```

In [29]:

```
#Creating the model
model = Sequential()
model.add(Embedding(vocab_size, 10, input_length=1))
model.add(LSTM(50, return_sequences=True))
model.add(LSTM(50))
model.add(Dense(50, activation="relu"))
model.add(Dense(vocab_size, activation="softmax"))
```

In [30]:

```
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 embedding_2 (Embedding)     (None, 1, 10)             106190    
                                                                 
 lstm_4 (LSTM)               (None, 1, 50)             12200     
                                                                 
 lstm_5 (LSTM)               (None, 50)                20200     
                                                                 
 dense_4 (Dense)             (None, 50)                2550      
                                                                 
 dense_5 (Dense)             (None, 10619)             541569    
                                                                 
=================================================================
Total params: 682,709
Trainable params: 682,709
Non-trainable params: 0
_________________________________________________________________
```

In [31]:

```
#Callbacks
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.callbacks import ReduceLROnPlateau
from tensorflow.keras.callbacks import TensorBoard

checkpoint = ModelCheckpoint("nextword1.h5", monitor='loss', verbose=1,
    save_best_only=True, mode='auto')

reduce = ReduceLROnPlateau(monitor='loss', factor=0.2, patience=3, min_lr=0.0001, 

logdir='logsnextword1'
tensorboard_Visualization = TensorBoard(log_dir=logdir)
```

In [33]:

```
#Compile model
model.compile(loss="categorical_crossentropy", optimizer=Adam(lr=0.001))
```

```
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/adam.py:105: UserWarnin
g: The `lr` argument is deprecated, use `learning_rate` instead.
  super(Adam, self).__init__(name, **kwargs)
```

In [35]:

```
#Fitting the Model
model.fit(X, y, epochs=20, batch_size=64, callbacks=[checkpoint, reduce, tensorboar
Epoch 1/20
353/353 [==============================] - ETA: 0s - loss: 8.3472
Epoch 00001: loss improved from 8.61569 to 8.34720, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 8.3472 - lr: 0.001
0
Epoch 2/20
351/353 [============================>.] - ETA: 0s - loss: 8.2409
Epoch 00002: loss improved from 8.34720 to 8.24102, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 8.2410 - lr: 0.001
0
Epoch 3/20
352/353 [============================>.] - ETA: 0s - loss: 8.1820
Epoch 00003: loss improved from 8.24102 to 8.18142, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 8.1814 - lr: 0.001
0
Epoch 4/20
352/353 [============================>.] - ETA: 0s - loss: 8.1350
Epoch 00004: loss improved from 8.18142 to 8.13467, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 8.1347 - lr: 0.001
0
Epoch 5/20
351/353 [============================>.] - ETA: 0s - loss: 8.0966
Epoch 00005: loss improved from 8.13467 to 8.09620, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 8.0962 - lr: 0.001
0
Epoch 6/20
353/353 [==============================] - ETA: 0s - loss: 8.0638
Epoch 00006: loss improved from 8.09620 to 8.06378, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 8.0638 - lr: 0.001
0
Epoch 7/20
353/353 [==============================] - ETA: 0s - loss: 8.0345
Epoch 00007: loss improved from 8.06378 to 8.03448, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 8.0345 - lr: 0.001
0
Epoch 8/20
351/353 [============================>.] - ETA: 0s - loss: 8.0109
Epoch 00008: loss improved from 8.03448 to 8.01057, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 8.0106 - lr: 0.001
0
Epoch 9/20
353/353 [==============================] - ETA: 0s - loss: 7.9892
Epoch 00009: loss improved from 8.01057 to 7.98919, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 7.9892 - lr: 0.001
0
Epoch 10/20
353/353 [==============================] - ETA: 0s - loss: 7.9683
Epoch 00010: loss improved from 7.98919 to 7.96829, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 7.9683 - lr: 0.001
0
Epoch 11/20
353/353 [==============================] - ETA: 0s - loss: 7.9471
Epoch 00011: loss improved from 7.96829 to 7.94715, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 7.9471 - lr: 0.001
0
Epoch 12/20
351/353 [============================>.] - ETA: 0s - loss: 7.9250
Epoch 00012: loss improved from 7.94715 to 7.92541, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 7.9254 - lr: 0.001
0
Epoch 13/20
353/353 [==============================] - ETA: 0s - loss: 7.9026
Epoch 00013: loss improved from 7.92541 to 7.90262, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 7.9026 - lr: 0.001
```

```
0
Epoch 14/20
352/353 [============================>.] - ETA: 0s - loss: 7.8787
Epoch 00014: loss improved from 7.90262 to 7.87851, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 7.8785 - lr: 0.001
0
Epoch 15/20
351/353 [============================>.] - ETA: 0s - loss: 7.8494
Epoch 00015: loss improved from 7.87851 to 7.84957, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 7.8496 - lr: 0.001
0
Epoch 16/20
352/353 [============================>.] - ETA: 0s - loss: 7.8104
Epoch 00016: loss improved from 7.84957 to 7.80999, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 7.8100 - lr: 0.001
0
Epoch 17/20
353/353 [==============================] - ETA: 0s - loss: 7.7503
Epoch 00017: loss improved from 7.80999 to 7.75030, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 7.7503 - lr: 0.001
0
Epoch 18/20
351/353 [============================>.] - ETA: 0s - loss: 7.6824
Epoch 00018: loss improved from 7.75030 to 7.68278, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 7.6828 - lr: 0.001
0
Epoch 19/20
351/353 [============================>.] - ETA: 0s - loss: 7.6183
Epoch 00019: loss improved from 7.68278 to 7.61779, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 7.6178 - lr: 0.001
0
Epoch 20/20
353/353 [==============================] - ETA: 0s - loss: 7.5594
Epoch 00020: loss improved from 7.61779 to 7.55942, saving model to nextword1.h5
353/353 [==============================] - 7s 20ms/step - loss: 7.5594 - lr: 0.001
0
```

Out[35]:

```
<keras.callbacks.History at 0x7f9c8517ccd0>
```

In [42]:

```python
#Prediction Script
# Importing the Libraries

from tensorflow.keras.models import load_model
import numpy as np
import pickle

# Load the model and tokenizer
model = load_model('nextword1.h5')
tokenizer = pickle.load(open('tokenizer.pkl', 'rb'))

def Predict_Next_Words(model, tokenizer, text):
    for i in range(3):
        sequence = tokenizer.texts_to_sequences([text])[0]
        sequence = np.array(sequence)

        preds = model.predict_classes(sequence)
        predicted_word = ""

        for key, value in tokenizer.word_index.items():
            if value == preds:
                predicted_word = key
                break
        print(predicted_word)
        return predicted_word
```

In [ ]:

```
while(True):

    text = input("Enter your line: ")

    if text == "stop the script":
        print("Ending The Program.....")
        break

    else:
        try:
            text = text.split(" ")
            text = text[-1]

            text = ''.join(text)
            Predict_Next_Words(model, tokenizer, text)

        except:
            continue
```

```
Enter your line: at the dull
weather
Enter your line: collection of textile
samples
Enter your line: what a strenuous
career
Enter your line: stop the script
Ending The Program.....
```