Final Project Code

Aashay Jain - 201112212

Yash hardenia - 201112228


```
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

from google.colab import files
uploaded = files.upload()

#Saving Pride_and_Prejudice.txt to Pride_and_Prejudice.txt

file = open("Pride_and_Prejudice.txt", "r", encoding = "utf8")

# store file in list
lines = []
for i in file:
    lines.append(i)

# Convert list to string
data = ""
for i in lines:
  data = ' '. join(lines)

#replace unnecessary stuff with space
data = data.replace('\n', '').replace('\r', '').replace('\ufeff', '').repl
ace('‾','').replace('”','')  #new line, carriage return, unicode character
 --> replace by space

#remove unnecessary spaces
data = data.split()
data = ' '.join(data)
```

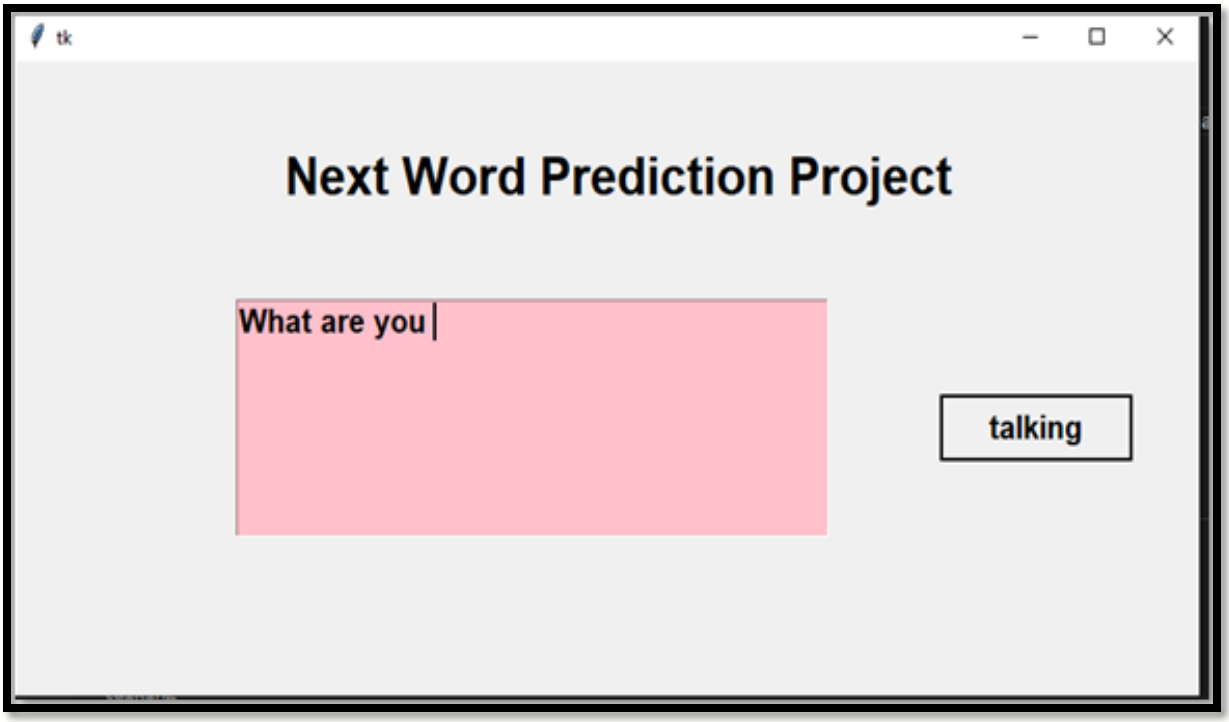
```
data[:500]

len(data)


tokenizer = Tokenizer()
tokenizer.fit_on_texts([data])

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

sequence_data = tokenizer.texts_to_sequences([data])[0]
sequence_data[:15]


len(sequence_data)

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

sequences = []

for i in range(3, len(sequence_data)):
    words = sequence_data[i-3:i+1]
    sequences.append(words)

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

X = []
y = []

for i in sequences:
    X.append(i[0:3])
    y.append(i[3])

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


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

model = Sequential()
model.add(Embedding(vocab_size, 10, input_length=3))
model.add(LSTM(1000, return_sequences=True))
model.add(LSTM(1000))
model.add(Dense(1000, activation="relu"))
```

```
model.add(Dense(vocab_size, activation="softmax"))

model.summary()

from tensorflow import keras
from keras.utils.vis_utils import plot_model

keras.utils.plot_model(model, to_file='plot.png', show_layer_names=True)

from tensorflow.keras.callbacks import ModelCheckpoint

checkpoint = ModelCheckpoint("next_words.h5", monitor='loss', verbose=1, s
ave_best_only=True)
model.compile(loss="categorical_crossentropy", optimizer=Adam(learning_rat
e=0.001),metrics=['accuracy'])
history=model.fit(X, y, epochs=70, batch_size=64, callbacks=[checkpoint])
print(model)

import matplotlib.pyplot as plt


def plot_graphs(history, string):
    plt.plot(history.history[string])
    plt.xlabel("Epochs")
    plt.ylabel(string)
    plt.show()

plot_graphs(history, 'accuracy')


from google.colab import files
files.download('next_words.h5')

files.download('token.pkl')
```

 # After Downloding the model, it is integrated with GUI in local system for real time next Word Prediction.

```
from tkinter import *
from tensorflow.keras.models import load_model
import numpy as np
import pickle

# Load the model and tokenizer
model = load_model('C:\\Users\\123\\Downloads\\next_words.h5')
tokenizer = pickle.load(open('C:\\Users\\123\\Downloads\\token.pkl', 'rb'))

def Predict_Next_Words(model, tokenizer, text):

  sequence = tokenizer.texts_to_sequences([text])
  sequence = np.array(sequence)
  preds = np.argmax(model.predict(sequence))
  predicted_word = " "
  count=0
  for key, value in tokenizer.word_index.items():
      if value == preds:
          predicted_word=key
          break

  return predicted_word



def f(event):
    text=E.get("1.0","end")
    print(type(text))
    x=text
    text=text.split(" ")
    if len(text)>=3:
        text = text[-3:]
        text[-1]=text[-1].strip()

        #print(text)
        ans=Predict_Next_Words(model, tokenizer, text)
        #ans=str(ans)
        print(ans)
        if len(ans)>0:
            l2.config(text=ans)
```

```
def append():
    x=l2['text']
    y=E.get("1.0","end")
    y=y.strip()
    E.delete("1.0","end")
    E.insert("1.0", y+" "+x)


root=Tk()
root.geometry("800x400")
X=StringVar()
Y=StringVar()
Label(text="Next Word Prediction Project",font='Helvatical 25
bold').place(x=180,y=50)

E=Text(root,bg="pink",font='Helvatical 15 bold')
E.place(x=150,y=150,width=400,height=150)
E.bind("<space>",f)


l2=Button(text="",font='Helvatical 15
bold',borderwidth=2,relief="solid",width=10,command=append)
l2.place(x=625,y=210)

t=Label(root,textvariable=Y)
t.pack()

root.mainloop()
```


tk
Next Word Prediction Project
What are you
talking