Date: 16 October 2023

Team ID: 329 PROJECT

ID: Proj\_227277\_Team\_1

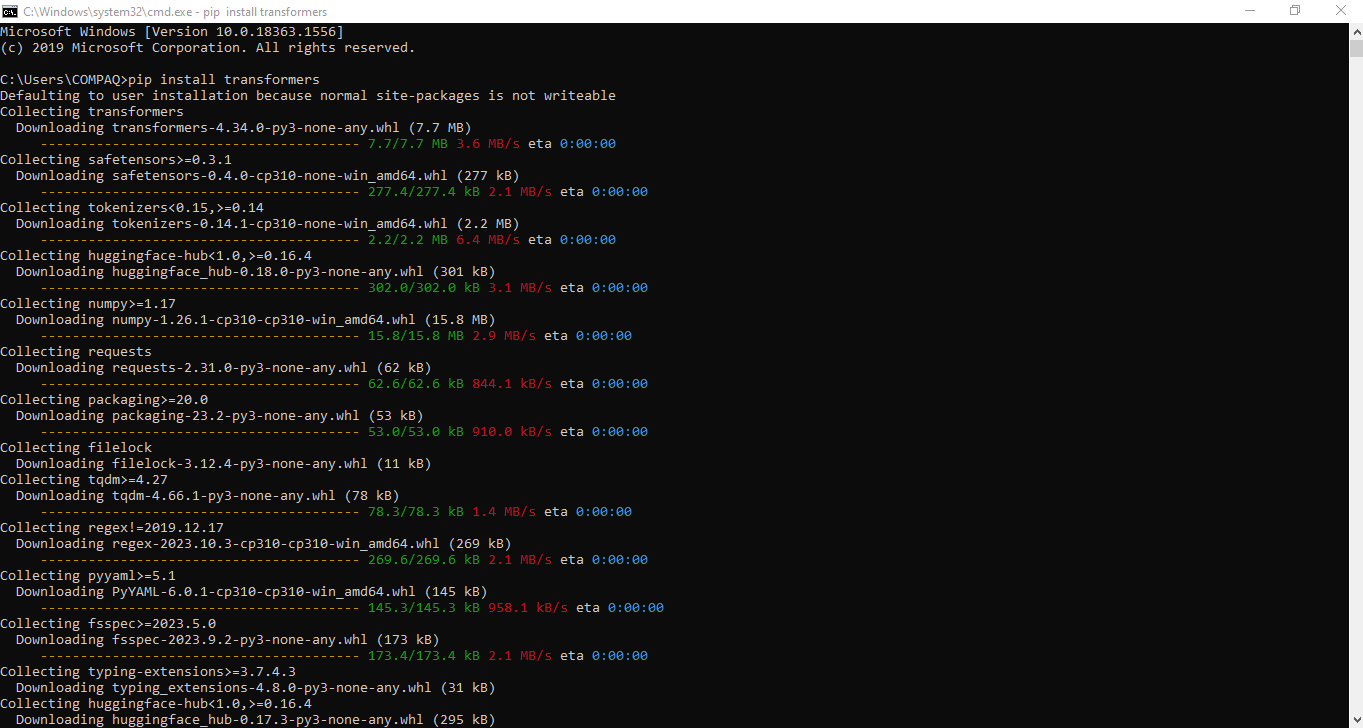
NAME: Gunasekar B

**Installing Packages**

1. Package name: transformers

Use: For GPT-3 integration

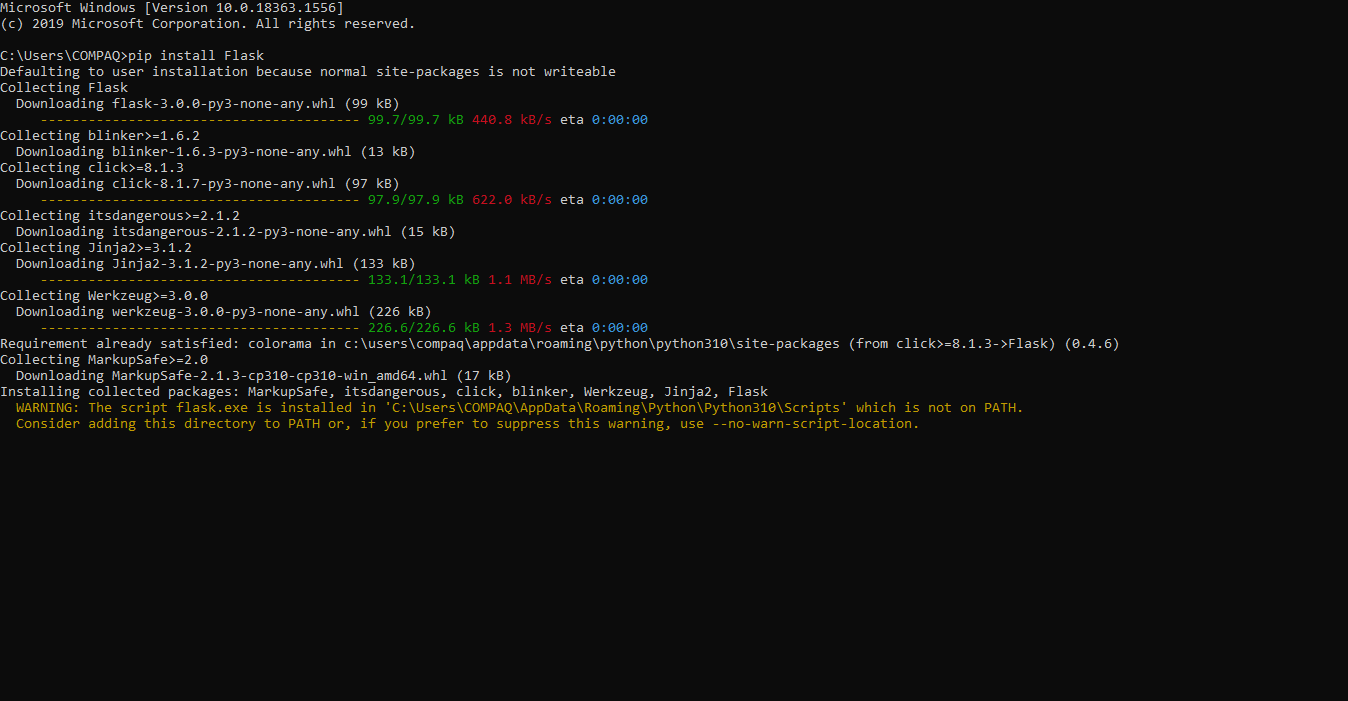
Command to install: pip install transformers



1. Package name: Flask

Use: For web app development

Command to install: pip install Flask

****

**Program for basic chat bot conversation**

I provided source code file called “AI\_Phase3\_source\_code.ipynb” in my git hub repository

# import all required libraries

import numpy as np

import string

from nltk.corpus import stopwords

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.neural\_network import MLPClassifier

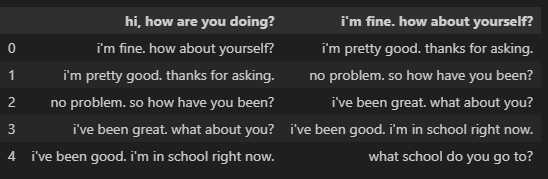
from sklearn.feature\_extraction.text import TfidfTransformer,TfidfVectorizer

from sklearn.pipeline import Pipeline

# importing the dataset

df = pd.read\_csv(r"C:\Users\COMPAQ\Desktop\IBM\dataset\dialogs.txt", sep='\t')

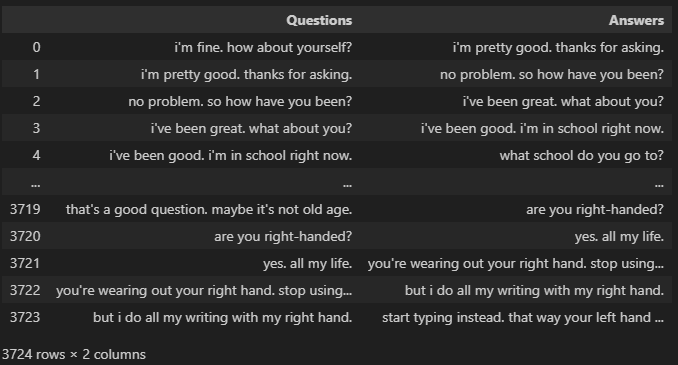
df.head()



#add column names

df.columns=['Questions','Answers']

df



# Data Preprocessing

# Assuming you have a DataFrame 'df' with 'Questions' and 'Answers' columns

df['question tokens'] = df['Questions'].apply(lambda x: len(x.split()))

df['answer tokens'] = df['Answers'].apply(lambda x: len(x.split()))

plt.style.use('fivethirtyeight')

fig, ax = plt.subplots(nrows=1, ncols=3, figsize=(20, 5))

sns.set\_palette('Set2')

# Create bar plots for question tokens and answer tokens

sns.barplot(x='question tokens', y=df.index, data=df, ax=ax[0])

ax[0].set\_xlabel('Question Tokens')

ax[0].set\_ylabel('Index')

ax[0].set\_title('Question Tokens Bar Plot')

sns.barplot(x='answer tokens', y=df.index, data=df, ax=ax[1])

ax[1].set\_xlabel('Answer Tokens')

ax[1].set\_ylabel('Index')

ax[1].set\_title('Answer Tokens Bar Plot')

# Create a scatter plot with a regression line for the relationship between question and answer tokens

sns.regplot(x='question tokens', y='answer tokens', data=df, ax=ax[2], scatter\_kws={'alpha':0.5})

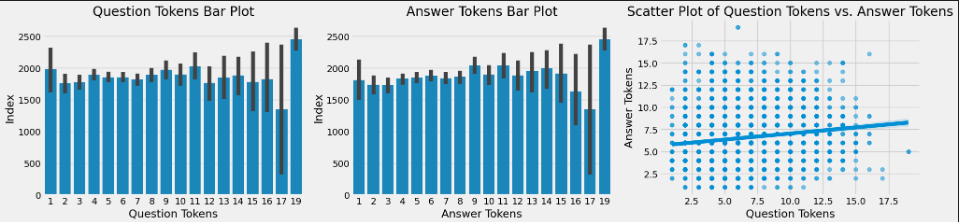
ax[2].set\_xlabel('Question Tokens')

ax[2].set\_ylabel('Answer Tokens')

ax[2].set\_title('Scatter Plot of Question Tokens vs. Answer Tokens')

plt.tight\_layout()

plt.show()



#Function for converting upper to lower case

def cleaner(x):

return [a for a in (''.join([a for a in x if a not in string.punctuation])).lower().split()]

#Model

Pipe = Pipeline([

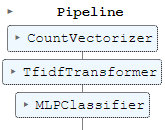
('bow',CountVectorizer(analyzer=cleaner)),

('tfidf',TfidfTransformer()),

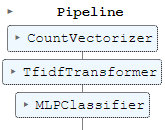
('classifier',MLPClassifier())

])

Pipe.fit(df['Questions'],df['Answers'])



Pipe.fit(df['Questions'],df['Answers'])



#Text

Pipe.predict(['like how clear the sky gets after it rains.'])[0]

5.png

Pipe.predict(['i want this trip to be perfect, i hope it stays warm.'])[0]

6.png

Pipe.predict(['it would not be good if it got cold this weekend.'])[0]



Pipe.predict(['it would be nice if the weather would never change.'])[0]



Pipe.predict(['why is that?'])[0]



Pipe.predict(['What are you doing'])[0]

