**Phase-3:**

**Transformers for GPT-3 integration and flask for web app development**.

**Introduction**:

Development Part 1In this pivotal phase of your project, Phase 3: Development Part 1, you embark on the journey of bringing your chatbot concept to life. Here, the focus is on laying the foundational elements that will enable your chatbot to interact with users seamlessly. This phase can be summarized into two essential tasks: preparing the groundwork for data and environment.

**Firstly**, you will delve into the world of data by loading and preprocessing your dataset. The dataset serves as the wellspring of knowledge that your chatbot will draw upon to provide meaningful responses. It is here that you'll shape and mold your data into a format that your chatbot can readily understand and use.

**Secondly**, you will create the necessary environment to develop and host your chatbot. This involves installing critical libraries that are the building blocks of your project. Notably, you'll set up the Transformers library, which opens the door to integrating powerful GPT-3 models into your chatbot. Additionally, Flask, a versatile Python web framework, will be employed to construct the user-friendly interface through which your chatbot will engage with its audience.By

**1. Environment Setup:**

\*To create an isolated environment for your project, you can use Python virtual environments. This ensures that your project's dependencies do not interfere with your system-wide Python installation.

\* You can create a new environment using a command like:

\* python -m venv chatbot-env

Activate the environment:

On Windows: `chatbot-env\Scripts\activate`

-On macOS and Linux: `source chatbot-env/bin/activate`

**2. Install Required Libraries:**

\*Install the libraries you'll need using `pip` or `conda`. Here's how you can install Transformers and Flask:

pip install transformers

pip install Flask

**3. Dataset Loading and Preprocessing:**

- \*Depending on your project's objectives, you may need a dataset to fine-tune your GPT-3 model. Datasets can be in various formats, such as text files, JSON, or CSV. Load and preprocess this data to make it compatible with your model.

**4. GPT-3 Integration:**

\*Hugging Face Transformers library provides easy integration with GPT-3 models. You can use code like this to load a pre-trained model:

python

from transformers import GPT3Tokenizer, GPT3LMHeadModel

model\_name = "EleutherAI/gpt-neo-1.3B"

tokenizer = GPT3Tokenizer.from\_pretrained(model\_name)

model = GPT3LMHeadModel.from\_pretrained(model\_name)

**5. Flask Web App Setup:**

\*Create a Flask application for your chatbot. You can start with a basic structure, defining routes and views for the web app. Here's a simple example:

python

from flask import Flask

app = Flask(\_\_name)

@app.route('/')

def index():

# Your code to render the chat interface

return "Chatbot Interface"

if \_\_name\_\_ == '\_\_main\_\_':

app.run()

**6. Basic User Interactions:**

\*Implement basic user interactions within your Flask app. This includes capturing user input from the web interface and sending it to the GPT-3 model for generating responses. Then, display the model's response back to the user on the interface.

**7. Testing and Debugging:**

\*Thoroughly test your chatbot and web app to ensure they function as intended. Debug any errors or issues that you encounter during testing.

**8. Deployment:**

\* Once your chatbot is working well and you're satisfied with the results, you can deploy it to a web server or a hosting platform for public access

\*These steps are the foundation for developing your chatbot project. As you progress, you can add more features, enhance user interactions, and continually improve your chatbot's performance based on user feedback and usage data.

**Import Libraries**

import tensorflow as tf

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from tensorflow.keras.layers import TextVectorization

import re,string

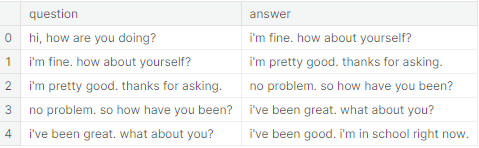
from tensorflow.keras.layers import LSTM,Dense,Embedding,Dropout,LayerNormalization

df=pd.read\_csv('/kaggle/input/simple-dialogs-for-chatbot/dialogs.txt',sep='\t',names=['question','answer'])

print(f'Dataframe size: {len(df)}')

df.head()

**Dataframe size: 3725**



**Data Preprocessing & Data Visualization:**

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

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

plt.style.use('fivethirtyeight')

fig,ax=plt.subplots(nrows=1,ncols=2,figsize=(20,5))

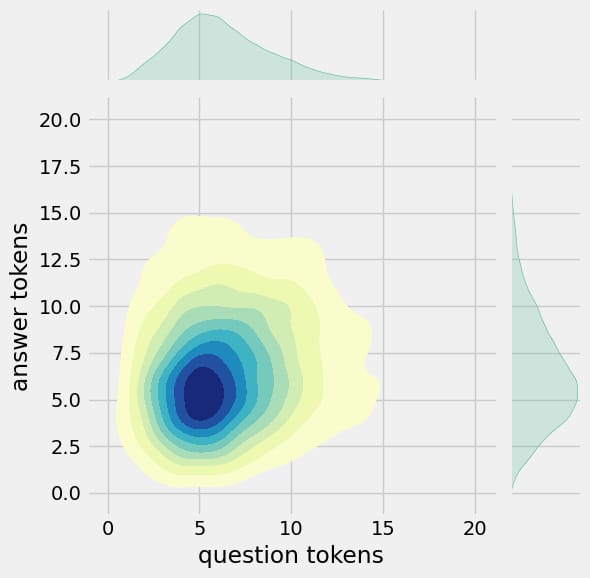
sns.set\_palette('Set2')

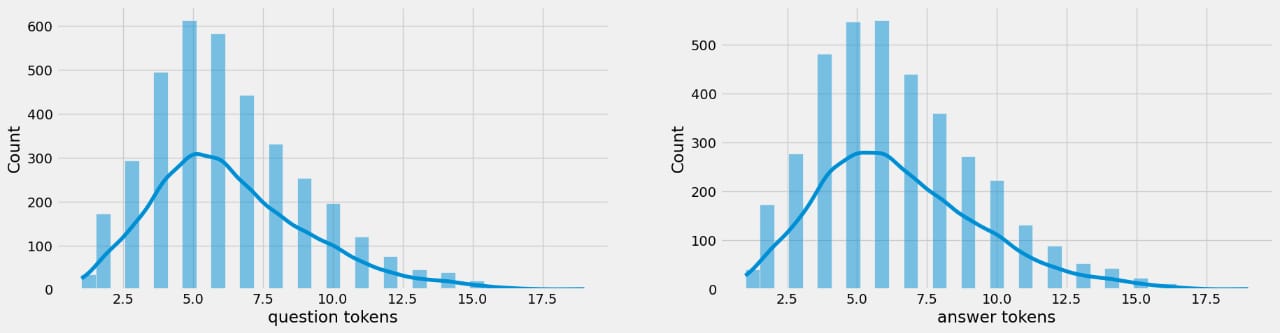
sns.histplot(x=df['question tokens'],data=df,kde=True,ax=ax[0])

sns.histplot(x=df['answer tokens'],data=df,kde=True,ax=ax[1])

sns.jointplot(x='question tokens',y='answer tokens',data=df,kind='kde',fill=True,cmap='YlGnBu')

plt.show()

****



**Conclusion:**

The end of this phase, your chatbot project will have taken its first steps towards becoming a fully functional and interactive virtual assistant. The groundwork laid here will set the stage for more intricate development and, ultimately, the deployment of your chatbot to the world.