Create a chatbot in Python

Phase 5: Project Documentation & Submission

Introduction:

To build a chatbot using GPT-3 (or any other version of GPT), you'll need to follow a few steps. I'll outline the process for you, including setting up your environment, installing required libraries, and implementing basic user interactions. In this example, we'll use the Hugging Face Transformers library for GPT-3 integration and Flask for web app development.

Ways to improve quality of GPT-3:

Here are some ways you can leverage pre-trained language models to enhance response quality:

1. Text Generation:

- GPT-3 can generate coherent and contextually relevant text. You can use it to automatically generate responses for chatbots, virtual assistants, or content generation.

2. Sentiment Analysis:

- You can use GPT-3 to analyze the sentiment of text input. This can be helpful for assessing customer feedback, social media sentiment analysis, or monitoring brand reputation.

3. Content Summarization:

- Pre-trained models can be employed to summarize long articles or documents efficiently. This can save time for users and help them quickly grasp the main points of a text.

4. Language Translation:

GPT-3 can be used for language translation tasks. It can translate text From one language to another, making it valuable for international Communication and content localization.

5. Question Answering:

- Pre-trained models can answer questions based on the context

Provided. You can use them to build intelligent question-answering systems

Or assist users in finding information.

6. Personalization:

- You can personalize responses by fine-tuning a pre-trained model with

User-specific data. This can result in more tailored and relevant

Interactions.

7. Conversational Agents:

- GPT-3 can be integrated into chatbots or virtual assistants to engage in

Natural and dynamic conversations with users. It can handle multi-turn

Conversations and maintain context effectively.

8. Content Recommendations:

- Analyze user preferences and behavior to recommend content or

Products. Pre-trained models can help in understanding user intent and

Suggesting relevant items.

9. Automatic Tagging and Categorization:

- Use pre-trained models to automatically tag and categorize text content,

Making it easier to organize and search for information.

10. Text Summarization for Search:

Enhance the quality of search results by generating concise summaries

Of search results. This can help users quickly assess which links to click on.

11. Creative Writing and Content Generation:

- Generate creative content, such as stories, poems, or marketing copy,

Using pre-trained models. They can assist in brainstorming ideas and

Crafting engaging narratives.

12. Text Completion and Suggestions:

- Offer text completion suggestions as users type, making their writing

More efficient and error-free.

Necessary step to follow:

1.Environment Setup:

First, you'll need to set up your development environment. You can use

A virtual environment to keep your project dependencies isolated.

You'll need Python installed on your system.

CODE:

Create a virtual environment

Python -m venv chatbot-env

Activate the virtual environment (on Windows)

Chatbot-env\Scripts\activate

1. Install Required Libraries:

Need to install the necessary libraries for chatbot:

Hugging Face Transformers: This library provides pre-trained models,

Including GPT-3.

Flask: A lightweight web framework for building the chatbot's user

Interface.

CODE:

Pip install ntlk

Pip install scikit-learn

Pip install pandas

Pip install numpy

Importance of loading and processing dataset:

 Loading and preprocessing the dataset is an important first

Step in

Building any machine learning model. However, it is especially

Important for house price prediction models, as house price

Datasets are often complex and noisy.

 By loading and preprocessing the dataset, we can ensure

That the

Machine learning algorithm is able to learn from the data

Effectively and accurately.

Sample Program:

```
Import random
Import pandas as pd
# Define a dataset of user input and chatbot responses
Dataset = pd.read_csv('E:\ab.csv')
# Function to generate a response
Def generate_response(user_input):
 User_input = user_input.lower()
 For key in dataset:
 If key in user_input:
 Return random.choice(dataset[key])
 Return "I'm sorry, I don't understand."
# Main chat loop
While True:
 User_input = input("You: ")
 If user_input.lower() == "exit":
 Print("Chatbot: Goodbye!")
 Break
 Response = generate_response(user_input)
 Print("Chatbot:", response)
```

Dataset Link: https://www.kaggle.com/datasets/grafstor/simple-dialogs-for-

Chatbot

Here's a list of tools and software commonly used in the

Process:

1. Programming Language:

- Python is the most popular language for machine learning due to

Its extensive libraries and frameworks. You can use libraries like NumPy,

Pandas, scikit-learn, and more.

2. Integrated Development Environment (IDE):

- Choose an IDE for coding and running machine learning

Experiments. Some popular options include Jupyter Notebook, Google

Colab, or traditional IDEs like PyCharm.

3. Data Visualization Tools:

- Tools like Matplotlib, Seaborn, or Plotly are essential for data

Exploration and visualization.

4. Data Preprocessing Tools:

- Libraries like pandas help with data cleaning, manipulation, and

Preprocessing.

Program:

```
Import numpy as np

Import pandas as pd

Import neattext.functions as ntf

Import string

Import nltk

Import random

From sklearn.feature_extraction.text import

TfidfVectorizer,CountVectorizer,TfidfTransformer

From sklearn.pipeline import Pipeline

From sklearn.tree import DecisionTreeClassifier

Sn=pd.read_csv("C:\Users\Deepak\Desktop\My_Project IBM\IBM-Team—Group-

5\dialogs.txt",sep='\t',names=['questions','answers'])

Sn

User_greetings=["hello","hi','hai','hiii','hii robo','hai robo','hello how are you doing']

Greeting_response=['hai','hello','hi thanks for asking how are you doing']
```

```
Sn['clean_questions']=sn['questions'].apply(ntf.remove_userhandles)
Sn['clean_questions']=sn['clean_questions'].apply(ntf.remove_punctuations)
Sn['clean_answers']=sn['answers'].apply(ntf.remove_userhandles)
Sn['clean_answers']=sn['clean_answers'].apply(ntf.remove_punctuations)
Sn
Pipe=Pipeline(steps=[('bow',CountVectorizer()),('tfidf',TfidfTransformer()),('dt',DecisionTr
eeClassifier())])
pipe.fit(sn['clean_questions'],sn['clean_answers'])
print("Robo: Wellcome I am Robo I am here to help you regarding all yours questions")
print('—'*30)
user=input("Enter your response:")
for I in list(user_greetings):
 if user==i:
 print(random.choice(greeting_response))
 print('Robo: Please type your question.If you have no question then type bye')
 while True :
 question=input('Robo: Enter your question here :')
 q1=pipe.predict([question])[0]
 print('Robo :',q1)
 if (question=='bye'or question=='thank you bye'or question=='ok bike'or
question=='bye robo'or question=='thanks robo bye'):
 print('Robo: bye thanks you for your coperation')
 break
 else :
 print('sorry I don't have any idea about your question,thanks for asking if you
have any other question start the conversation else type bye')
```

Output:

```
Robo: Wellcome I am Robo I am here to help you regarding all yours questions
Enter your response: what are you doing?
```

I’m going to change the light bulb. It burnt out.

Enter your response:bye

Goodbye

Name:Rahul S

Roll Number:420121104042

Dep:CSE -3-Year

College Name:AKT MCET

GROUP: AI GROUP 5