CREATE A CHATBOT USING PYTHON

NAME : SUBASH S

REGITER NUMBER : 622521104053

PROJECT NAME : CREATE A CHOTBOT IN PYTHON

# DOCUMENT : PHASE – 3 SUBMISSION



# CREATE CHATBOT IN PYTHON

Introduction:

we will walk through the process of creating a chatbot with GPT-3 integration using the Transformers library and developing a web interface for user interactions with Flask, a popular web framework for Python. This chatbot will be capable of engaging in dynamic and natural conversations, providing an interactive and user-friendly experience.

The document is structured to guide you through the steps necessary to set up the development environment, integrate the OpenAI GPT-3 model, and build a simple web-based interface for users to interact with the chatbot. Whether you are an experienced developer looking to expand your skill set or a beginner seeking to build your first chatbot, this guide is designed to be accessible and informative.

We will cover the following key aspects of the project:

1.Setting up the Environment

2.Integration with GPT-3

3.Building a Flask Web App

4.Customizing and Enhancing the Chatbot

Steps to create a basic chatbot with GPT-3 integration and a Flask web app, you'll need to follow these steps:

1.Setup environment

First, make sure you have Python installed on your system.

2.Create a virtual environment

Creating a virtual environment helps isolate your project's dependencies from the global Python environment.

Program:

# Create a virtual environment python -m venv chatbot\_env

# Activate the virtual environment

# On Windows

chatbot\_env\Scripts\activate

# On macOS and Linux

source chatbot\_env/bin/activate

 optional but recommended

## 3.Install required packages

Install the necessary libraries using pip, including Transformers and Flask.

Program:

pip install transformers flask

4.OpenAI API Key

To use GPT-3, you need an API key from OpenAI.

## 5.Create a Flask Web App

Here's a simple example of a Flask app that communicates with the GPT-3 API:

Program:

# Import necessary libraries from flask import Flask, request import openai # Initialize Flask app app = Flask(\_\_name\_\_) # Your OpenAI API key api\_key = "YOUR\_API\_KEY"

# GPT-3 endpoint gpt3\_endpoint= "https://api.openai.com/v1/engines/davinci/completions"

# Define a route for handling chat interactions@app.route('/chat', methods=['POST']) def chat():

data = request.get\_json() user\_message = data['message] # Call GPT-3 to generate a response response = openai.Completion.create( engine="davinci", prompt=user\_message, max\_tokens=50, api\_key=api\_key

)

bot\_message = response.choices[0].text return jsonify({'message': bot\_message})

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

app.run()

## 6.Run your Flask App

Run your Flask app using following command Program:

python your\_app\_filename.py

## 7.Access the chatbot

|  |  |  |
| --- | --- | --- |
| Your Flask app should now be running locally. You can access it by | | |
| visiting | <http://127.0.0.1:5000/> | in your web browser. You can also make POST |
|  | | |
| requests to the ‘chat’ endpoint to interact with the chatbot programmatically. | | |

## 8.Improve and customise

You can further customize and improve your chatbot by refining the interactions, handling user input, and enhancing the chat experience.

Given dataset:

Sample: hi, how are you doing? i'm fine.

how about yourself?

i'm fine. how about yourself? i'm pretty good.thanks for asking. i'm pretty good. thanks for asking. no problem. so how have you been? no problem. so how have you been? i've been great. what about you? i've been great. what about you? i've been good. i'm in school right now. what school do you go to?

i go to pcc.

And more.,

Key aspects of the project:

## 1.Setting up the environment

1.Installing Python and required libraries.

2.Creating a virtual environment for project isolation.

## 2.Integration with GTP-3

1. Obtaining an OpenAI API key.
2. Interacting with the GPT-3 model using the Transformers library.

## 3.Building a Flask Web App

1. Setting up a basic Flask application.
2. Creating routes for user interactions.
3. Displaying chatbot responses in a web interface.

## 4.Enhancing the chatbot

1. Implementing user input handling.
2. Extending functionality and customization options.
3. Error handling and improvements.

Conclusion:

In this document, we have walked through the process of creating a chatbot with GPT-3 integration and a Flask web app. We started by setting up the development environment, obtaining an OpenAI API key, and installing the necessary libraries. Then, we built a simple but functional chatbot that can engage in dynamic and natural conversations with users.

Here are the key takeaways from this guide:

1.Environment Setup: We learned how to create a virtual environment to isolate our project's dependencies and how to install the required libraries, including Transformers for GPT-3 integration and Flask for web app development.

2.GPT-3 Integration:We obtained an OpenAI API key and learned how to interact with the GPT-3 model using the Transformers library. This allowed us to generate human-like responses to user queries.

3.Flask Web App: We created a basic Flask web application to serve as the interface for our chatbot. We defined routes for user interactions and displayed chatbot responses in a web-based chat interface.

4.Customization and Expansion:While we built a simple chatbot in this guide, there are many opportunities for customization and enhancement. You can further improve the chatbot by adding more features, handling user input, and refining the user experience.

Building a chatbot is just the beginning. With this foundation, you can explore more advanced concepts such as natural language understanding, sentiment analysis, and integrating the chatbot into other

applications. The possibilities are endless, and the skills you've acquired can be applied to a wide range of projects.

P a g e |

We hope that this guide has provided you with valuable insights and a solid starting point for your chatbot development journey. Chatbots have the potential to revolutionize customer service, information retrieval, and many other domains. As you continue to work on your chatbot project, don't hesitate to explore more advanced NLP models, enhance the user interface, and, most importantly, have fun while building and refining your conversational AI.