CREATE A CHATBOT USING PYTHON:

College Code:5113

TEAM MEMBERS:

D.Aakash(511321106001)

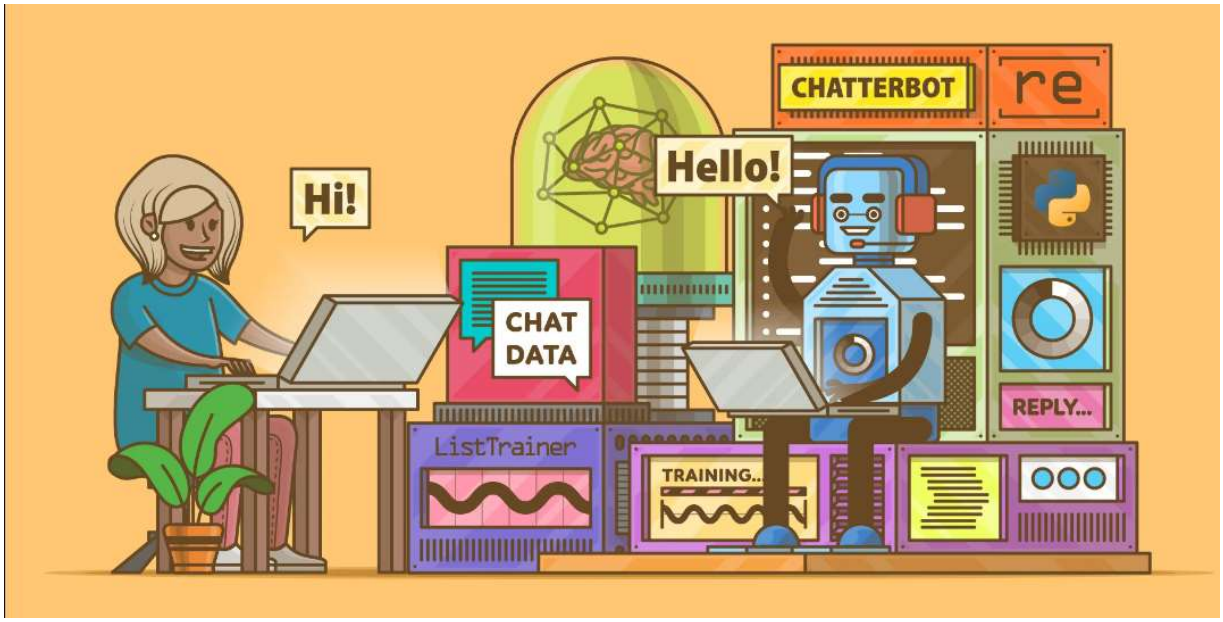
S.Chandrsekran(511321106005)

A.Dinesh(511321106010)

J.Yukendiran(511321106306)

Phase 2 Project:

Project Title: Chatbot using Python



INTRODUCTION:

- A chatbot is an automated software program that interacts with humans. A chatbot is merely a computer program that fundamentally simulates human conversations. A chatbot that functions through AI and

machine learning has an artificial neural network inspired by the neural nodes of the human brain. Chatbots are programs that can do talk like human conversations very easily. For example, Facebook has a machine learning chatbot that creates a platform for companies to interact with their consumers through the Facebook Messenger application. In 2016, chatbots became too popular on Messenger. By the consequences is noted that2016 was the entire year of chatbots. The software industry is mainly oriented on chatbots. Thousands of chatbots are invented on startups and used by the businesses to improve their customer service, keeping them hanging by a kind communication. According to research, nowadays chatbots are used to solve a number of business tasks across many industries like E-Commerce, Insurance, Banking, Healthcare,Finance, Legal, Telecom, Logistics, Retail, Auto, Leisure,Travel, Sports, Entertainment, Media and many others. Thus that was the moment to look at the chatbots as a new technology in the communication field. Nowadays various companies are using chatbots to answer quickly and efficiently some frequented asking questions from their own customers.

- AIML and LSA are used for creating chatbots. Artificial Intelligence Markup Language (AIML) and Latent Semantic Analysis (LSA) are used for developing chatbots,which are used to define general pattern-based queries. This patterncan also be used to give random responses for the samequery in the chatbot. LSA is a Latent Semantic Analysis technology in python, which is utilized to discover like nesses between words as vector representation. So that the unanswered queries by AIML will be viewed as a reply by LSA.

REQUIRMENTS:

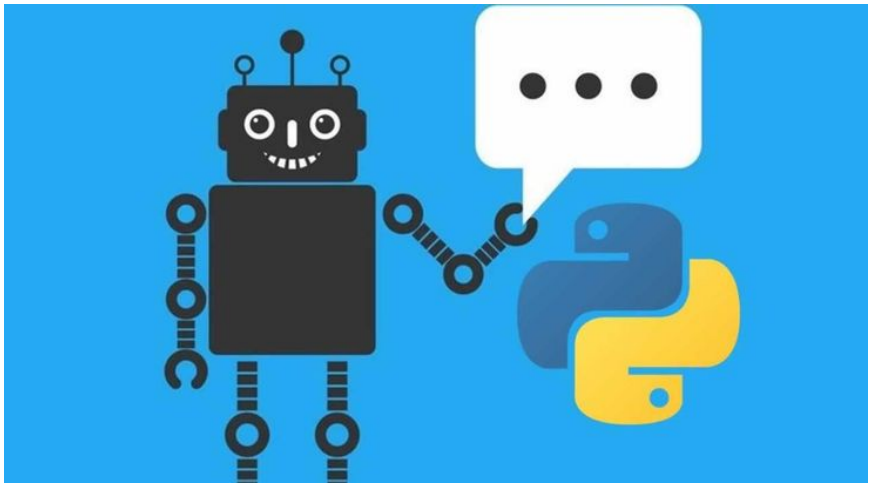
It Requires both software and hardware

Software:

- Programming languages like (Python (or) Pycharm)
- A chatbot development platform (such as Dialogflow, Microsoft Bot Framework, or IBM Watson)
- A text editor or integrated development environment (IDE)

Hardware:

- Computer or laptop (windows,mac,linux)
- Internet



Keywords: Chabot, Chabot architecture, Artificial Intelligence, NLU (Natural language understanding),NLP (Natural language processing) Artificial Intelligence Markup Language
(AIML), Latent Semantic Analysis (LSA)

1.Chatbot's Purpose and Goals:

- Determine the specific problem or task your chatbot will address.
- Identify the goals and objectives you want to achieve with the chatbot.

2. Collect and Prepare Data:

- Gather relevant data for training and fine-tuning your chatbot.
- Preprocess and clean the data to make it suitable for training the model.

3. Choose a Pre-trained Language Model:

- Select a pre-trained language model like GPT-3 or other alternatives (e.g., GPT-4, BERT) based on your project requirements and budget.

4. Set Up the Development Environment:

- Install the necessary Python libraries (e.g., **openai**, **numpy**, **tensorflow**) and dependencies for working with the chosen language model.

5. API Integration:

- Obtain API access credentials for the selected language model.
- Configure your Python code to interact with the model's API using the provided credentials.

6. Design Conversational Flows:

- Plan the conversation flows and user interactions.
- Define how the chatbot should respond to various user inputs and scenarios.

7. Implement Natural Language Processing (NLP) Techniques:

- Use NLP libraries like spaCy or NLTK for tasks such as tokenization, entity recognition, and sentiment analysis.
- Apply techniques like dialogue management to maintain context in multi-turn conversations.

8. Train and Fine-Tune the Model:

- If required, fine-tune the pre-trained model using your specific data and objectives.
- Experiment with hyperparameters and training configurations to optimize performance.

9. Test and Evaluate:

- Test the chatbot extensively to ensure it provides accurate and contextually relevant responses.
- Implement automated tests and collect user feedback for improvement.

10. Implement Advanced Features: - Consider adding advanced features like: - Multilingual support for global users. - Voice input/output for speech-based interactions. - Integration with external APIs and databases for dynamic data retrieval. - Personalization for customizing responses based on user preferences.

11. Implement User Authentication and Security: - If the chatbot handles sensitive information, implement user authentication and ensure data security.

12. Deploy the Chatbot: - Choose a hosting platform or cloud service to deploy your chatbot. - Set up scaling mechanisms to handle increased traffic.

13. Monitor and Maintain: - Continuously monitor the chatbot's performance and user interactions. - Update the model and responses as needed to adapt to changing requirements.

14. User Feedback Loop: - Implement a feedback mechanism for users to report issues and provide suggestions. - Use user feedback to iterate and improve the chatbot's performance.

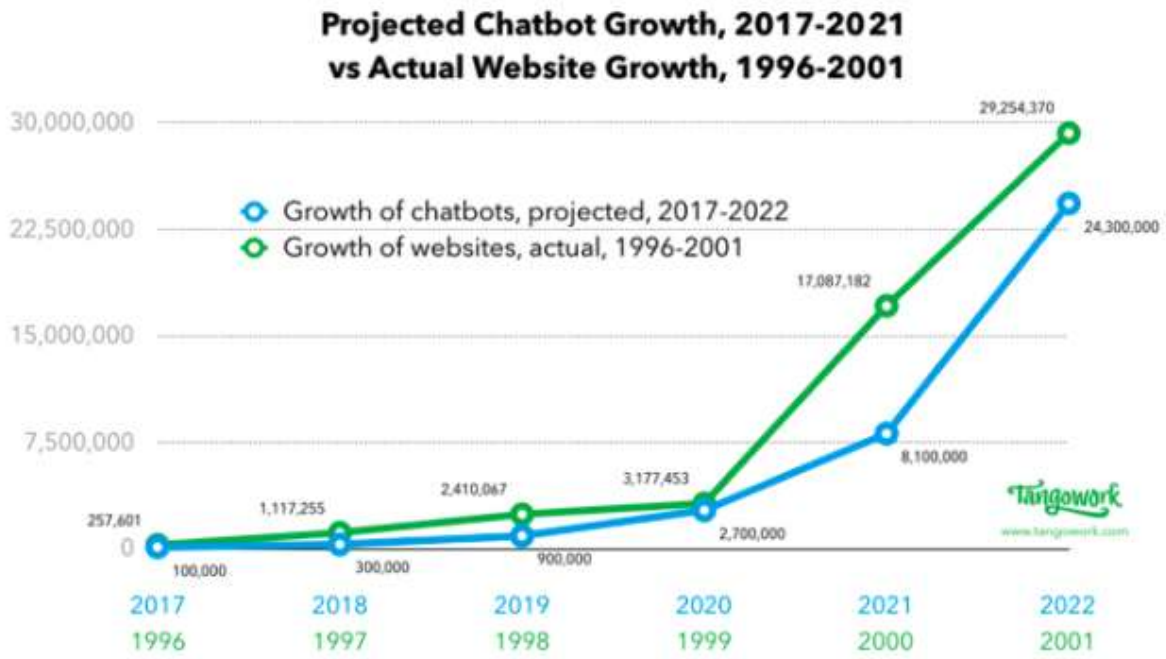
15. Compliance and Ethical Considerations: - Ensure that your chatbot complies with legal and ethical standards, including data privacy regulations.

16. Marketing and Promotion: - Develop a strategy for promoting your chatbot to potential users. - Consider social media marketing, partnerships, or app store listings.

17. Continuous Innovation: - Stay updated with the latest advancements in NLP and AI. - Continuously innovate and enhance your chatbot's capabilities to remain competitive and relevant.

Building an innovative chatbot using advanced techniques like GPT-3 requires a combination of technical expertise, creativity, and a user-centric approach. Regularly gather user feedback and iterate on your chatbot to make it increasingly valuable to your target audience.

GROWTH OF CHATBOT:



REQUIRMENTS:

1. Python 3.x
2. OpenAI GPT-3 API access
3. **openai** library (install with **pip install openai**)
4. **numpy** library (install with **pip install numpy**)

Step 1: Sign up for the OpenAI GPT-3 API and obtain your API key**.**

Step 2: Here's a more advanced example of a chatbot script:

PROGRAM:

```
import openai

import numpy as np

# Replace with your GPT-3 API key

api_key = "YOUR_API_KEY"

# Initialize the OpenAI API client
```

```
openai.api_key = api_key

def generate_response(prompt, max_tokens=50, temperature=0.7):
    response = openai.Completion.create(
        engine="davinci",  # You can use other engines like "curie" or "babbage" as well.
        prompt=prompt,
        max_tokens=max_tokens,
        temperature=temperature
    )
    return response.choices[0].text

# Initialize conversation
conversation_history = []

print("Chatbot: Hi, how can I assist you today? (Type 'exit' to end)")

while True:
    user_input = input("You: ")
    if user_input.lower() == "exit":
        print("Chatbot: Goodbye!")
        break

    # Add the user's message to the conversation history
    conversation_history.append(f"You: {user_input}")

    # Generate a response from the chatbot
    chatbot_message = generate_response("\n".join(conversation_history))

    # Display the chatbot's response
    print(f"Chatbot: {chatbot_message}")

    # Add the chatbot's message to the conversation history
    conversation_history.append(f"Chatbot: {chatbot_message}")
```

In this script:

- We initialize the OpenAI API client with your API key.
- The **generate_response** function sends a prompt to GPT-3 and retrieves a response.
- We maintain a conversation history to keep track of the chat's context.
- The chatbot continually interacts with the user, extending the conversation history as needed.

Step 3: Run the Chatbot

Save the script to a Python file (e.g., **chatbot.py**) and run it. You can have a conversation with your chatbot by entering text messages.

This is a basic example, and you can further enhance it by adding features like context management, sentiment analysis, or custom responses for specific user inputs. You can also consider using other NLP libraries like spaCy or NLTK to process and analyze user input for more sophisticated interactions.

Phase 2 CONCLUSION:

In this phase we summarized and disscused the advanced Techniques like using pre training languages models (eg.,CHATGPT-3) to enhance thequalities And upcoming sessions we are going to start builiding the chatbot by preparing the environment and implementing basic user interactions.