4222-SURYA GROUP OF INSTITUTION

VIKIRAVANDI-605 652.

NAAN MUDHALVAN PROJECT

CREATE CHATBOT IN PYTHON

Presented by:

M.PRATHIP

REGNO:422221106013

ECE DEPARTMENT

Chatbot In Python

Introduction:

At the most basic level, a chatbot is a computer program that simulates and processes human conversation (either written or spoken), allowing humans to interact with digital devices as if they were communicating with a real person. ChatterBot is a library in python which generates responses to user input. It uses a number of machine learning algorithms to produce a variety of responses. It becomes easier for the users to make chatbots using the ChatterBot library with more accurate responses.

ML used in chatpot in pythonL:

- NLP(Natural languages processing)
- Transformer-based deep learning algorithm

Scope of chatbot:

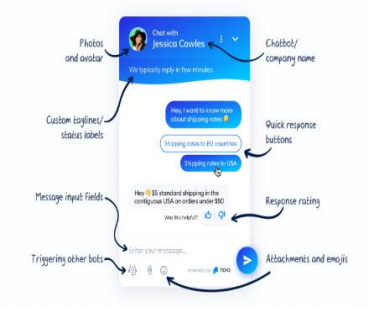
Chatbots can provide instant assistance to customers, which can help reduce wait times and improve customer satisfaction. In the future, chatbots may become even more sophisticated and be able to handle more complex customer service interactions. A chatbot is a software or computer program that simulates human conversation or "chatter" through text or voice interactions.

Users in both business-to-consumer (B2C) and business-to-business (B2B) environments increasingly use chatbot virtual assistants to handle simple tasks.

Why chatbot:

Organizations looking to increase sales or service productivity may adopt chatbots for time savings and efficiency, as artificial intelligence (AI) chatbots can converse with users and answer recurring questionsconsumers move away from traditional forms of communication, many experts expect chat-based communication methods

to rise. Organizations increasingly use chatbot-based virtual assistants to handle simple tasks, allowing human agents to focus on other responsibilities.



User Interface in Chatbot

A chatbot user interface (UI) is part of a chatbot that users see and interact with. This can include anything from the text on a screen to the buttons and menus that are used to control a chatbot. The chatbot UI is what allows users to send messages and tell it what they want it to do.

Steps taken:

- Step 1: Create a Chatbot Using Python ChatterBot
- Step 2: Begin Training Your Chatbot
- Step 3: Export a WhatsApp Chat
- Step 4: Clean Your Chat Export

- Step 5: Train Your Chatbot on Custom Data and Start Chatting

Program:

Step 1:

```
# bot.py

from chatterbot import ChatBot

chatbot = ChatBot("Chatpot")

exit_conditions = (":q", "quit", "exit")
while True:
    query = input("> ")
   if query in exit_conditions:
        break
  else:
        print(f"□ {chatbot.get_response(query)}")
```

How do chatbots offer responses?

The bot displays responses in the same order you composed them. You can apply Filters to your bot responses to trigger them only when a condition is met. You can decide how fast your chatbot should respond to a user's question using the Delay feature.

Step 2:

```
# bot.py

from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer
```

```
chatbot = ChatBot("Chatpot")

trainer = ListTrainer(chatbot)
trainer.train([
    "Hi",
    "Welcome, friend 🤗",
])
trainer.train([
    "Are you a plant?",
    "No, I'm the pot below the plant!",
])

exit_conditions = (":q", "quit", "exit")
while True:
    query = input("> ")
    if query in exit_conditions:
        break
    else:
        print(f"🪴 {chatbot.get_response(query)}")
```

Integrate chatbot on a website:

Inserting the chatbot on your site couldn't be easier. Beneath the chatbot builder, there's a shortcode that you can use to insert the chatbot into a page or post on your WordPress site. You simply copy that code and paste it where you want the chatbot to appear on the page/post.

Step 4:

```
# cleaner.py

import re

def remove_chat_metadata(chat_export_file):
    date_time = r"(\d+\/\d+\/\d+,\s\d+:\d+)"  # e.g. "9/16/22, 06:34"
    dash_whitespace = r"\s-\s"  # " - "
    username = r"([\w\s]+)"  # e.g. "Martin"
    metadata_end = r":\s"  # ": "
   pattern = date_time + dash_whitespace + username + metadata_end

   with open(chat_export_file, "r") as corpus_file:
       content = corpus_file.read()
  cleaned_corpus = re.sub(pattern, "", content)
   return tuple(cleaned_corpus.split("\n"))

if __name__ == "__main__":
  print(remove_chat_metadata("chat.txt"))
```

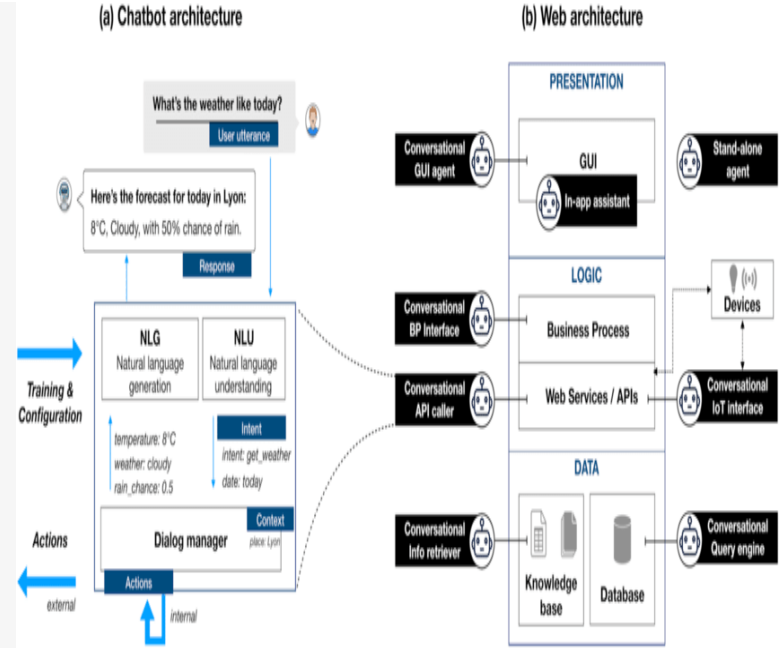
Step 5:

```
# cleaner.py

def remove_non_message_text(export_text_lines):
    messages = export_text_lines[1:-1]

    filter_out_msgs = ("<Media omitted>",)
    return tuple((msg for msg in messages if msg not in filter_out_msgs))

if __name__ == "__main__":
    message_corpus = remove_chat_metadata("chat.txt")
    cleaned_corpus = remove_non_message_text(message_corpus)
    print(cleaned_corpus)
```



Conclusion:

In this project, we have introduced a chatbot that is able to interact with users. This chatbot can answer queries in the textual user input. For this purpose, AIML with program-o has been used. The chatbot can answer only those questions which he has the answer in its AIML dataset.