IBM NAAN MUDHALVAN

CREATE A CHATBOT IN PYTHON

ATIFICIAL INTELLIGENCE

SUBMITTED BY SNEHA .P.S [AU411521104108]

11/10/2023

PHASE 2 : INNOVATION

Table of contents

SI.NO	CONTENTS	PG.NO
1.	Intoduction	2
2.	Project overview	2
3.	Requirements	3
4.	Prerequisties	3
5.	Initial implementation code	4
6.	What is a Chatbot?	4
7.	How to Make a Chatbot in Python?	5
8.	How Does the Chatbot Python Work?	5
9.	What is ChatterBot Library?	6
10.	Limitations With A Chatbot	7
11.	Final source code	7
12.	Innovation	11
13.	Conclusion	12

Introduction:

A coumpter program designed to stimulate conversation with human users, especially over the internet. It allowing human to interact with digital devices as if they were communication with a real person. There are different types of chatbots which are used for interaction between human and digital devices Menu or button-based chartbots, Rules-based chatbots, AI-powered chatbots,Voice chatbots, Generative AI chatbots.

Some example real-time bots are Mitsuku Bot, Jabberwacky etc. OR A chatbot is computer program that can learn over time how best interact with humans.And also mainly ChatGPT is an artificial intelligence(AI)

Chatbot that uses natural language processing to create humanlike conversational dialogue. The language model can respond to questions and compose various written content, including articles, social media posts, essays,code and emails. As a general rule, you can distinguish between two types of chatbots: Rule-based chatbots and AI bots.

Project overview:

The ChatterBot library combines language corpora, text processing, machine learning algorithms, and data storage and retrieval to allow you to build flexible chatbots.You can build an industry-specific chatbot by training it with relevant data. Additionally, the chatbot will remember user responses and continue building its internal graph structure to improve the responses that it can give.

While ChatterBot is still a popular open source solution for building a chatbot in Python, it hasn't been actively maintained for a while and has therefore accumulated a significant number of issues.There are multiple forks of the project that implement fixes and updates to the existing codebase, but you'll have to personally pick the fork that implements the solution you're looking for and then install it directly from GitHub. A fork might also come with additional installation instructions.To get started, however, you won't use a fork. Instead, you'll use a specific pinned version of the library, as distributed on PyPI. You'll find more information about installing ChatterBot in step one

This automated communication system is developed using Python. The project file contains a python script (main.py, trainingData.py, JSON file, and pkl file). Talking about this chatbot, it allows the user to provide suitable queries about the college and replies with suitable answers. You change the data by changing the given JSON file. Also, this is a simple cmd-based project which is easy to understand and use.

Requirements:

- *pip*
- *NumPy*
- *random*
- *nltk*
- *tensorflow*

Prerequisites:

Before you get started, make sure that you have a Python version available that works for this ChatterBot project. What version of Python you need depends on your operating system:

1. *Windows*
2. *Linux*
3. *macOS*

If you've installed the right Python version for your operating system, then you're ready to get started:

1. *Conditional statements*
2. *while loops for iteration*
1. *Lists and tuples*
2. *Python functions*
3. *Substring checks and substring replacement*
4. *File input/output*

5. *Python comprehensions and generator expressions*
6. *Regular expressions (regex) using re*

Initial implementation code:

```
from chatterbot import ChatBot

from chatterbot.trainers import ListTrainer

from cleaner import clean_corpus

CORPUS_FILE = "chat.txt"

chatbot = ChatBot("Chatpot")

trainer = ListTrainer(chatbot)

cleaned_corpus = clean_corpus(CORPUS_FILE)

trainer.train(cleaned_corpus)

exit_conditions = (":q", "quit", "exit")

while True:

    query = input("> ")

    if query in exit_conditions:

        break

    else:

        print(f"    {chatbot.get_response(query)}")
```

What is a Chatbot?

Artificial intelligence is used to construct a computer program known as "a chatbot" that simulates human chats with users. It employs a technique known as NLP to comprehend the user's inquiries and offer pertinent information. Chatbots have various functions in customer service, information retrieval, and personal support.

How to Make a Chatbot in Python?

1. Preparing the Dependencies

The right dependencies need to be established before we can create a chatbot. Python and a ChatterBot library must be installed on our machine. With Pip, the Chatbot Python package manager, we can install ChatterBot.

2. Creating and Training the Chatbot

Once the dependence has been established, we can build and train our chatbot. We will import the ChatterBot module and start a new Chatbot Python instance. If so, we might incorporate the dataset into our chatbot's design or provide it with unique chat data.

3. Communicating with the Python chatbot

We can send a message and get a response once the chatbot Python has been trained. Creating a function that analyses user input and uses the chatbot's knowledge store to produce appropriate responses will be necessary.

4. Complete Project Code

We will give you a full project code outlining every step and enabling you to start. This code can be modified to suit your unique requirements and used as the foundation for a chatbot.

How Does the Chatbot Python Work?

The main approaches to the development of chatbots are as follows:

1. Rule-Based Approach

The Chatbot Python adheres to predefined guidelines when it comprehends user questions and provides an answer. The developers often define these rules and must manually program them.

2. Self-Learning Approach:

Chatbots that learn their use of machine learning to develop better conversational skills over time. There are two categories of self-learning chatbots:

RetrievalBased Models: Based on an input question, these models can obtain predefined responses from a knowledge base. They evaluate user input and compare it to the closest equivalent response in the knowledge base.

Generative Models: Generative models create responses from scratch based on the input query. They employ approaches like sequence-to-sequence models or transformers, for example, to produce human-like answers

What is ChatterBot Library?

A Chatbot Python library called The ChatterBot makes it simpler to create chatbots. It manages the challenges of natural language processing and provides a specific API. The following are some of Chatterbot's primary features:

1. Language Independence

You can use Chatterbot to create chatbots in various languages based on your target demographic.

2. How Does ChatterBot Library Work?

Chatterbot combines a spoken language data database with an artificial intelligence system to generate a response. It uses TF-IDF (Term Frequency-Inverse Document Frequency) and cosine similarity to match user input to the proper answers.

3. How To Install ChatterBot In Python

You can launch a terminal or command prompt. Your machine needs to be configured to run Ai Chatbot Python. To check if this is the case, run the command "python version" or "python3 Version" to ensure it returns a valid Python version.

Installing Chatterbot using the Chatbot Python Package Manager that comes with your Python program. To do this, issue the command "Pip installing chatterbot."

This command will download and install the ChatterBot library and its dependencies.

Once setup is complete, add the following code to your Chatbot using Python script or interactive environment to include Chatterbot: Imported from Chatterbot is ChatBot.

You may now use Chatterbot to begin building your chatbot. Using the ChatterBot guide or other resources, you can learn how to set up and train a chatbot.

Limitations With A Chatbot:

While chatbots have come a long way, there are still some limitations to be aware of:

- *Lack of semantic understanding: Chatbots may require assistance comprehending the discourse, which could result in misinterpretation or incorrect responses.*
- *Dependency on training data: The caliber and volume of training data greatly impact chatterbotpython performance. There may be a need for more accurate or biased training data, which can result in incorrect responses.*
- *Handling complicated queries: Chatbots could encounter questions beyond simple pattern matching and call for greater comprehension or deductive reasoning.*

FINAL SOURCE CODE:

```
# for speech-to-text

import speech_recognition as sr

# for text-to-speech
```

```
from gtts import gTTS
# for language model
import transformers
import os
import time
# for data
import os
import datetime
import numpy as np
# Building the AI
class ChatBot():
    def __init__(self, name):
        print("----- Starting up", name, "-----")
        self.name = name
    def speech_to_text(self):
        recognizer = sr.Recognizer()
        with sr.Microphone() as mic:
            print("Listening...")
            audio = recognizer.listen(mic)
            self.text="ERROR"
    try:
        self.text = recognizer.recognize_google(audio)
        print("Me  -->", self.text)
    except:
```

```
            print("Me  -->  ERROR")
    @staticmethod
    def text_to_speech(text):
        print("Kim -->", text)
        speaker = gTTS(text=text, lang="en", slow=False)
        speaker.save("res.mp3")
        statbuf = os.stat("res.mp3")
        mbytes = statbuf.st_size / 1024
        duration = mbytes / 200
        os.system('start res.mp3')  #if you are using mac->afplay or else for
windows->start
        # os.system("close res.mp3")
        time.sleep(int(50*duration))
        os.remove("res.mp3")
    def wake_up(self, text):
        return True if self.name in text.lower() else False
    @staticmethod
    def action_time():
        return datetime.datetime.now().time().strftime('%H:%M')
# Running the AI
if __name__ == "__main__":
    ai = ChatBot(name="dev")
    nlp = transformers.pipeline("conversational",
model="microsoft/DialoGPT-medium")
    os.environ["TOKENIZERS_PARALLELISM"] = "true"
```

```
ex=True
while ex:
    ai.speech_to_text( )
    ## wake up
    if ai.wake_up(ai.text) is True:
        res = "Hello I am Kim  the AI, what can I do for you?"
    ## action time
        elif "time" in ai.text:
            res = ai.action_time( )
    ## respond politely
        elif any(i in ai.text for i in ["thank","thanks"]):
            res = np.random.choice(["you're welcome!","anytime!","no problem!","cool!","I'm here if you need me!","mention not"])
        elif any(i in ai.text for i in ["exit","close"]):
            res = np.random.choice(["Tata","Have a good day","Bye","Goodbye","Hope to meet soon","peace out!"])
            ex=False
    ## conversation
    else:
        if ai.text=="ERROR":
            res="Sorry, come again?"
        else:
            chat = nlp(transformers.Conversation(ai.text), pad_token_id=50256)
            res = str(chat)
            res = res[res.find("bot >> ")+6:].strip( )
```

```
    ai.text_to_speech(res)
print("----- Closing down Kim -----")
```

INNOVATION:

Customized learning :

Personalized attention to students advances their results as the tutors get to knowledge of the domain where the learners are fragile in. The availability of personal educators to individual students of different capacities can conceive larger number of professionals. Students can acquire deeper knowledge of their interests. Technology Mediated Learning(TML) is defined as "an environment in which the learner's interaction with learning materials, peers, and instructors are mediated through advanced information technologies".

Easy tasks of tutors:

It is a false assumption for teachers who think the chatbots may take up their job and they will be laid off. If only simplifies take for them by helping students with frequent queires and assessing them personally. Teachers can equip themselves with the latest research during the supplementary time they get.

Integration of chatbot to classroom:

Apart from standalone chatbots, there as been increase in the integration of this chatbots in social platform such as facebook , Google classroom and so on.

Appealing methods of online education:

How effective can chatbots be in education, also relies on it attractive design. Reeves, B& Nass , C.(1996) exemplified in their investication that most of the human consider social platform such as television, computers and internet as their fellow beings and treat them with more respect.

CONCLUSION:

Chatbot Python has gained widespread attention from both technology and business sectors in the last few years. These smart robots are so capable of imitating natural human languages and talking to humans that companies in the various industrial sectors accept them. They have all harnessed this fun utility to drive business advantages, from, e.g., the digital commerce sector to healthcare institutions.