VELAMMAL
INSTITUTE OF TECHNOLOGY

Approved by AICTE - New Delhi
Affiliated to Anna University - Chennai
Accredited by NBA & NAAC

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

TEAM MEMBERS:

BOLLEDDU KISHORE (113321106012)
CHILAKA SARATH(113321106015)
KONDURU ABHILASH(113321106046)
MIKILI MANOAJITH(113321106051)
NEERUKATTU MADHAN KUMAR(113321106061)

CREATE A CHATBOT USING PYTHON

Types of chatbots

- Chatbots are computer program that simulates human conversation through voice commands or text chats or both in natural language.
- understand the user’s intent and send responses based on the application’s business rules and data.

- They can decipher verbal or written questions and provide responses with appropriate information or direction

Types of chatbot frameworks

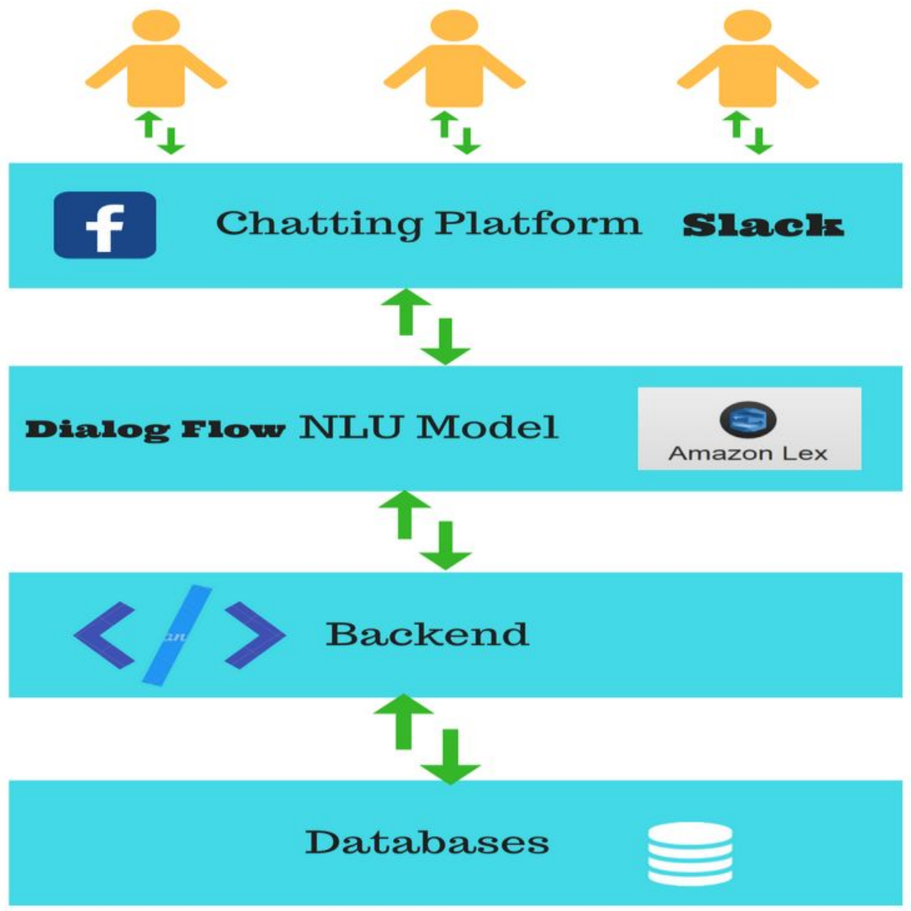
There are different types of Chatbot Frameworks such as–

1.Dialog flow
2. Alexa
3. Luis
 4.Wit
5. Rasa

- Of these frameworks Rasa is an Open Source framework for developing text-and voice-based chatbots and assistants. It comprises of mainly two things, Rasa NLU, and Rasa Core. The benefit of Rasa Stack is data privacy and zero running cost. And also being open source no cost of using the platform in Rasa, though developers cost will be higher as here you need experienced developers to develop and maintain the chatbot.
- One can choose dialog flow or alexa if one need a quick solution. But for highly scalable, data sensitive assistant Rasa Stack is better.

AI Chatbot Architecture

As in the AI bot architecture diagram say for Chatting Platform Slack the chat starts from chatting platform as input request and end at chat platform with response.
In between the message passes into so many stages.
Here NLU models are chat bot APIs which extract the information from user request.
The Web-hook back end is simple web application which provides the restful API. NLU model extract the information in the form of entity which works as parameter in these restful APIS



It produces some response that you get as robot response. The only magic comes with the *Natural Language Processing* (NLP) applies in these NLUs. Lets understand it with an example byte – ”Order 3 roses for me oncoming Monday” .

Here NLU should extract three information–

- Product -Rose
- User– Current Login
- Product Quantity-3
- Time– Date corresponding to Monday If your system is intelligent enough to extract these parameter, You can easily develop the backend for it. Chat bot can save so much time for consumer to understand the product user interface. These NLU API also provide voice interface to user. Once you speak anything to bot, it first converts speech to text. Once it gets the text, NLP comes into picture

How Chatbot Actually Works?

The chatbots work by adopting 3 classification methods:

1.Pattern Matching

• Bots use pattern matching to classify the text and produce a suitable response for the users. A standard structure of these patterns is “Artificial Intelligence Markup Language” (AIML). A simple pattern matching example is:

```
<aim l version=“1.0.1”encoding=“UTF”?>
<category>
<pattern>WHO IS ABRAHAM LINCOLN</pattern>
<template>Abraham Lincoln was the us president during American civil war.</template>
</category>

<category>
<pattern>DO YOU KNOW WHO*IS</pattern>
<template>
<srai>WHO IS<star/></srai>
</template>
</category>
</aim l>
```

2.Algorithms

For each kind of question, a unique pattern must be available in the database to provide a suitable response. With lots of combination on patterns, it creates a hierarchical structure. We use algorithms to reduce the classifiers and generate the more manageable structure. Computer scientists call it a “Reductionist” approach- in order to give a simplified solution, it reduces the problem.

Multinomial Naive Bayes is the classic algorithm for text classification and NLP. For an instance, let’s assume a set of sentences are given which are belonging to a particular class.

- With new input sentence, each word is counted for its occurrence and is accounted for its commonality and each class is assigned a score. The highest scored class is the most likely to be associated with the input sentence

Algorithms

For example Sample Training set

class: greeting

“How you doing?”

“good morning”

“hi there”

Few sample Input sentence classification:

input: “Hello good morning”

term: “hello” (no matches)

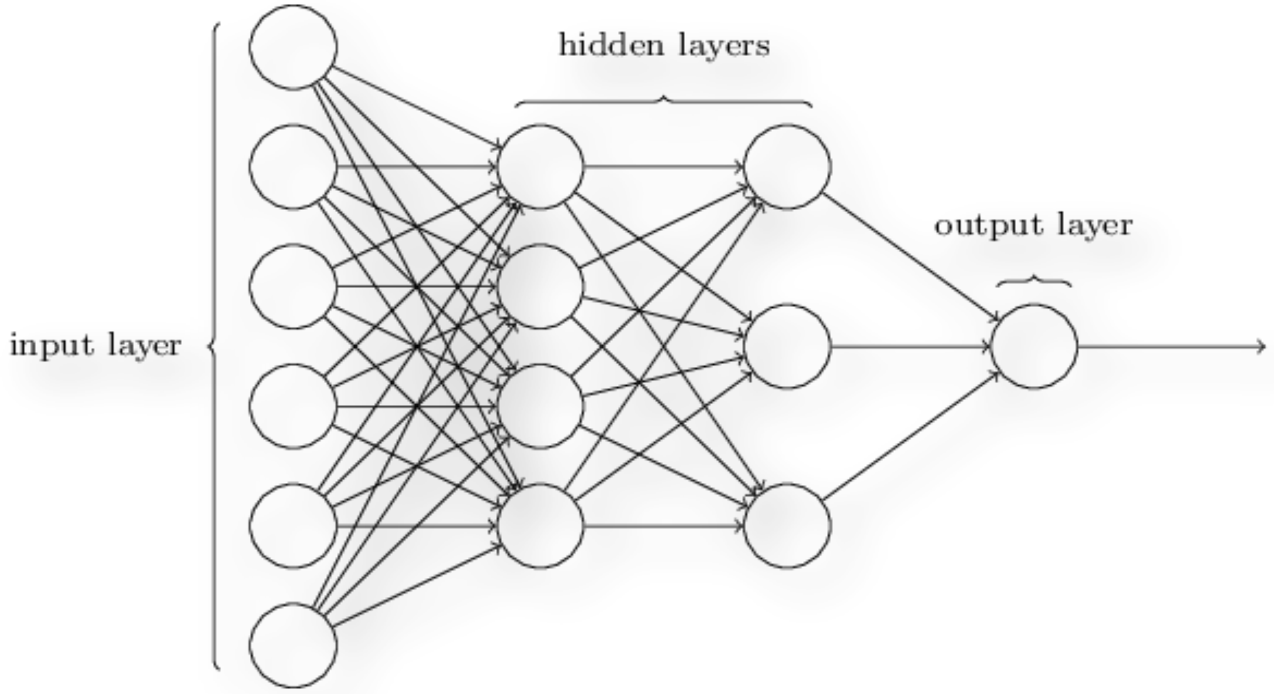
Term: “good” (class: greeting)

term: “morning” (class: greeting)

classification: greeting (score=2) Algorithms

3.Artifical neural networks

Neural Networks are a way of calculating the output from the input using weighted connections which are calculated from repeated iterations while training the data. Each step through the training data amends the weights resulting in the output with accuracy

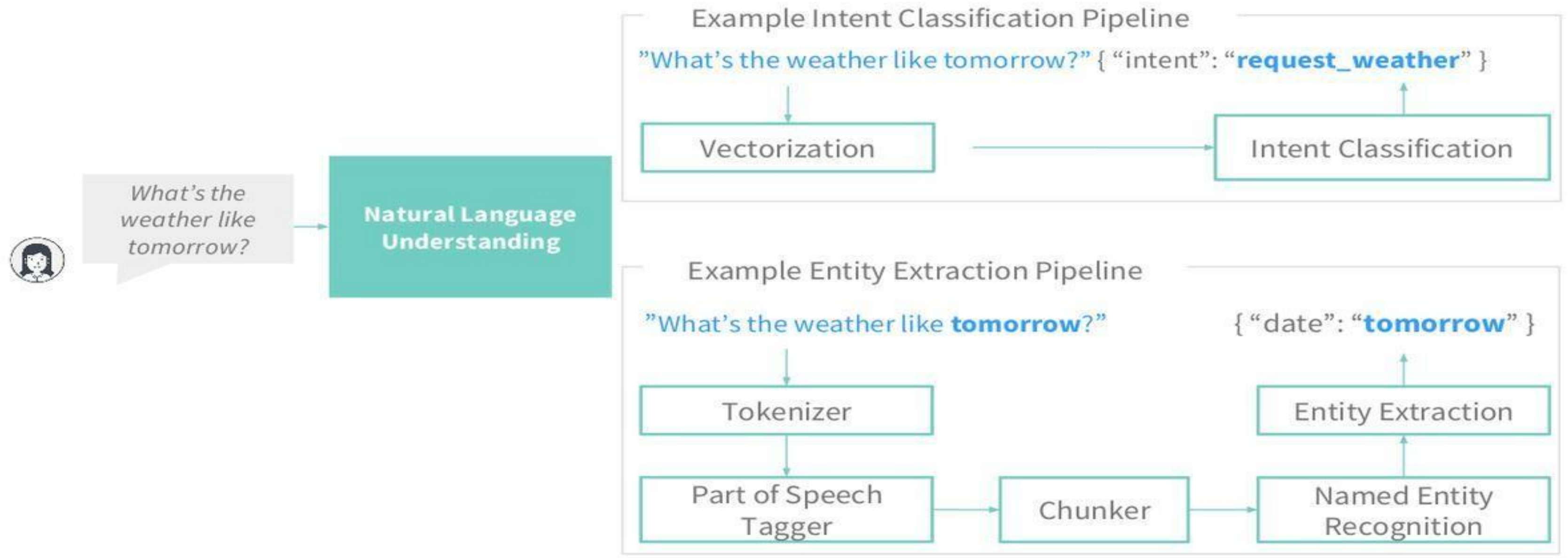


Natural Language Understanding (NLU)

It has 3 specific concepts like:

- Entities: Entity basically represents a concept in your Chatbot. It might be a payment system in Ecommerce Chatbot, NGO in NGO Darpan Portal etc.
- Intents: It is basically the action chatbot should perform when the user say something. For instance, intent can trigger same thing if user types "I want to order a red pair of shoes", "Do you have red shoes? I want to order them" or "Show me some red pair of shoes", all of these user's text show trigger single command giving users options for Red pair of shoes.
- Context: When a NLU algorithm analyzes a sentence, it does not have the history of the user conversation. It means that if it receives the answer to a question it has just asked, it will not remember the question .

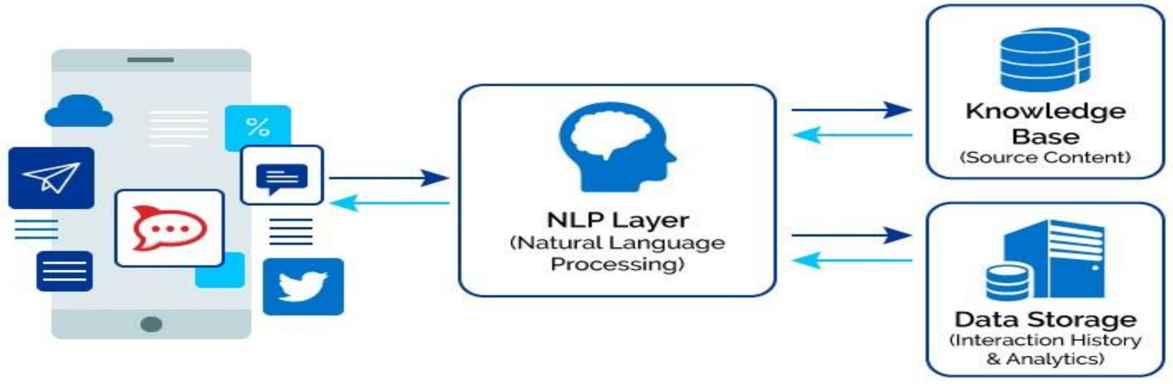
Rasa NLU: Natural Language Understanding


What's the weather like tomorrow?
Natural Language Understanding
Example Intent Classification Pipeline
"What's the weather like tomorrow?" { "intent": "request_weather" }
Vectorization
Intent Classification
Example Entity Extraction Pipeline
"What's the weather like tomorrow?"
{ "date": "tomorrow" }
Tokenizer
Entity Extraction
Part of Speech Tagger
Chunker
Named Entity Recognition

Natural Language Processing (NLP)

Some of the Natural Language Processing steps are:

- Tokenization: The NLP divides a string of words into pieces or tokens that are linguistically symbolic or are differently useful for the application.
- Named Entity Recognition: The chatbot program model looks for categories of words, like the name of the product, the user’s name or address, whatever data is required.
- Normalization: The Chatbot program model processes the text in an effort to find common spelling mistakes or typographical errors that might effect what the user intents to convey. This gives more human like effect of the Chatbot to the users.
- Dependency Parsing: The Chatbot looks for the objects and subjects- verbs, nouns and common phrases in the user’s text to find dependent and related phrases that users might be trying to convey.
- Sentiment Analysis: Tries to learn if the user is having a good experience or if after some point the chat should be forwarded to the human.


NLP Layer
(Natural Language Processing)
Knowledge Base
(Source Content)
Data Storage
(Interaction History & Analytics)

Chatbot with Rasa Stack and Python

RASA stack is an open-source AI tool and being an open source framework, it is easy to customize. Clients usually do not want to share their data and majority of the tools available are cloud-based and provide software as a service. With RASA, One can build, deploy or host Rasa internally in your server or environment with complete control on it.

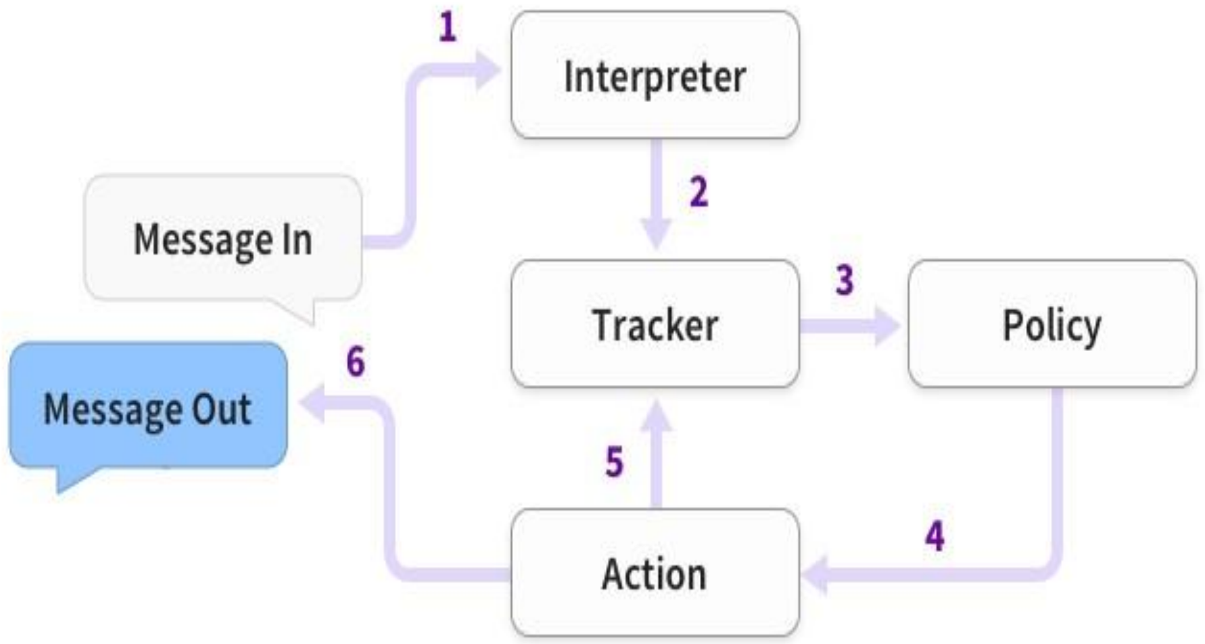
Rasa comes up with 2 components —

- Rasa NLU — a library for natural language understanding (NLU) which does the classification of intent and extract the entity from the user input and helps bot to understand what the user is saying.

- Rasa Core — a chatbot framework with machine learning-based dialogue management which takes the structured input from the NLU and predicts the next best action using a probabilistic model like LSTM neural network. NLU and Core are independent and one can use NLU without Core, and vice versa. Though Rasa recommends using both

Rasa Core

Finally, the training of Bot is considered successful where it is able to understand the natural language but we still need to build the dialogues so that bot can respond to the messages. After training the bot we need to build a dialogues management for bot to respond to the messages.



ChatterBot: Build a Chatbot With Python

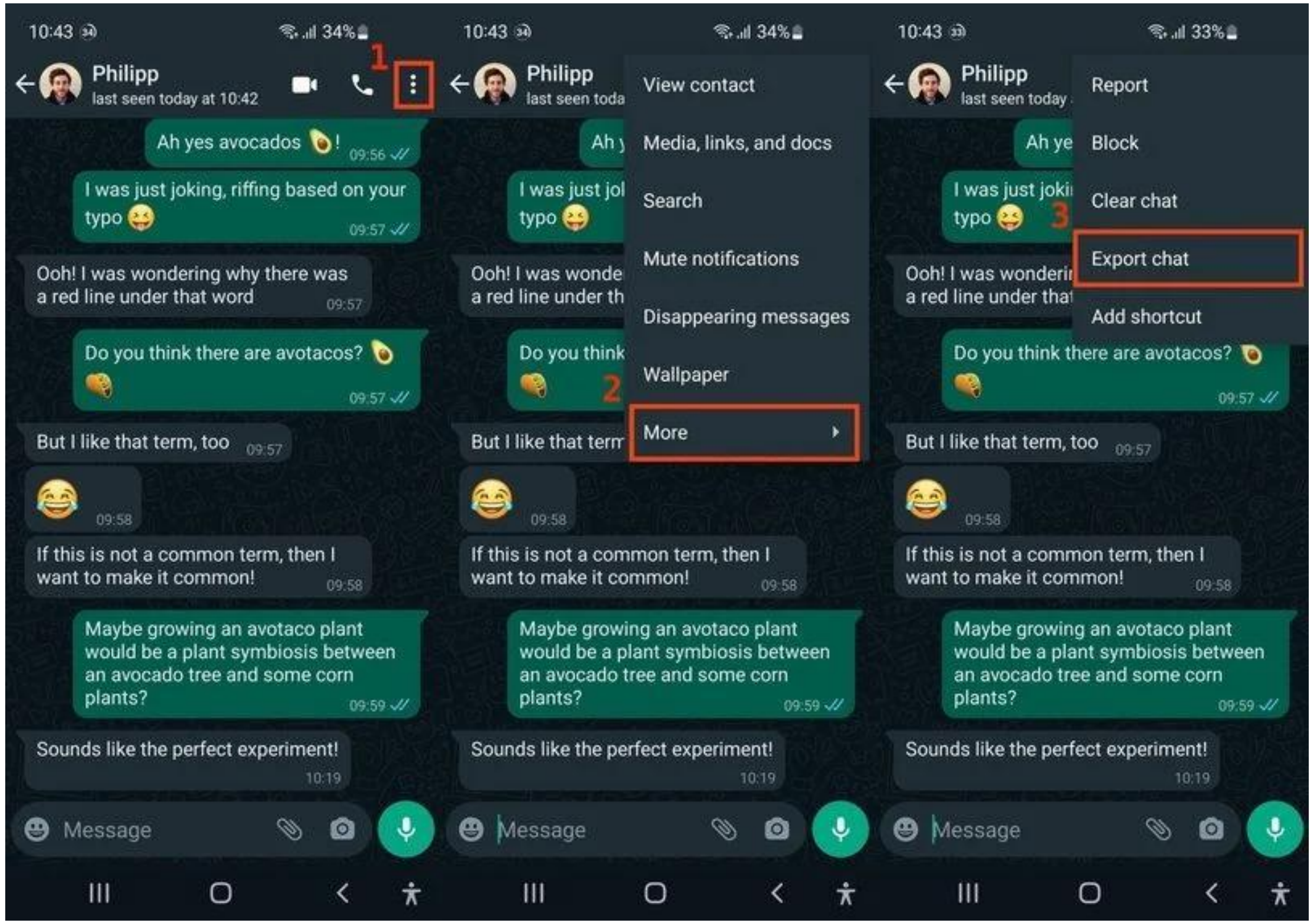
Step 1: Create a Chatbot Using Python ChatterBot

- from chatterbot import ChatBotchatbot = ChatBot("Chatpot")exit_conditions = (":q", "quit", "exit")while True: query = input("> ")   if query in exit_conditions: break   else: print(f"🪴 {chatbot.get_response(query)}")

Step 2: Begin Training Your Chatbot

- from chatterbot import ChatBotfrom chatterbot.trainers import ListTrainerchatbot = 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)}")

Step 3: Export a WhatsApp Chat



Philipp
last seen today at 10:42

Ah yes avocados 🥑! 09:56

I was just joking, riffing based on your typo 😝 09:57

Ooh! I was wondering why there was a red line under that word 09:57

Do you think there are avotacos? 🥑🌮 09:57

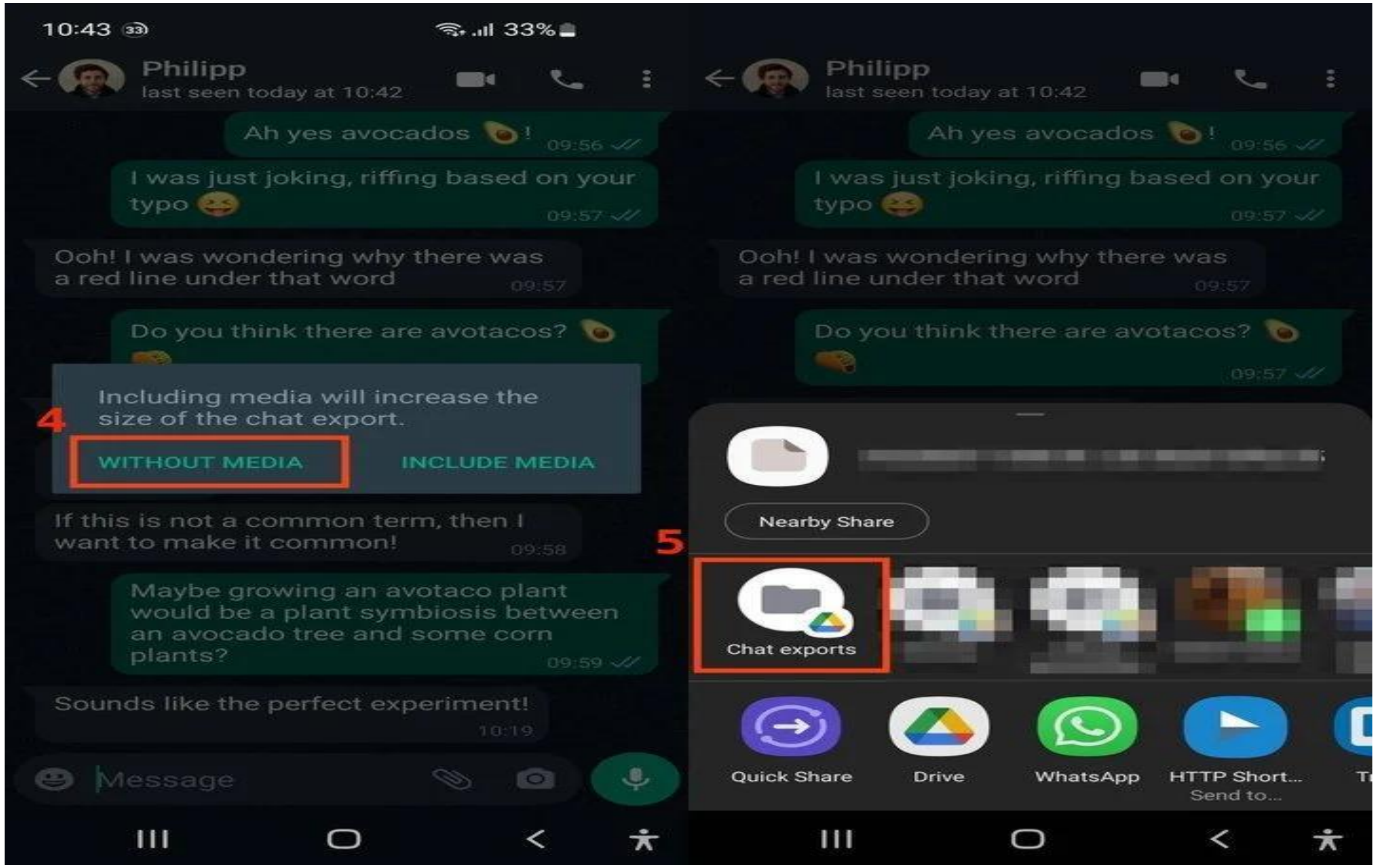
But I like that term, too 09:57

😂 09:58

If this is not a common term, then I want to make it common! 09:58

Maybe growing an avotaco plant would be a plant symbiosis between an avocado tree and some corn plants? 09:59

Sounds like the perfect experiment! 10:19



Ah yes avocados 🥑! 09:56

I was just joking, riffing based on your typo 😝 09:57

Ooh! I was wondering why there was a red line under that word 09:57

Do you think there are avotacos? 🥑🌮 09:57

Including media will increase the size of the chat export.

If this is not a common term, then I want to make it common! 09:58

Maybe growing an avocado plant would be a plant symbiosis between an avocado tree and some corn plants? 09:59

Sounds like the perfect experiment! 10:19

Ah yes avocados 🥑! 09:56

I was just joking, riffing based on your typo 😝 09:57

Ooh! I was wondering why there was a red line under that word 09:57

Do you think there are avotacos? 🥑🌮 09:57

Step 4: Clean Your Chat Export

- import redef 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"))

- 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)

- import redef clean_corpus(chat_export_file):   message_corpus = remove_chat_metadata(chat_export_file)   cleaned_corpus = remove_non_message_text(message_corpus)   return cleaned_corpus# ...# Deleted: if _name_ == "_main_":

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

- from chatterbot import ChatBotfrom chatterbot.trainers import ListTrainerfrom cleaner import clean_corpusCORPUS_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)}")

THANK YOU