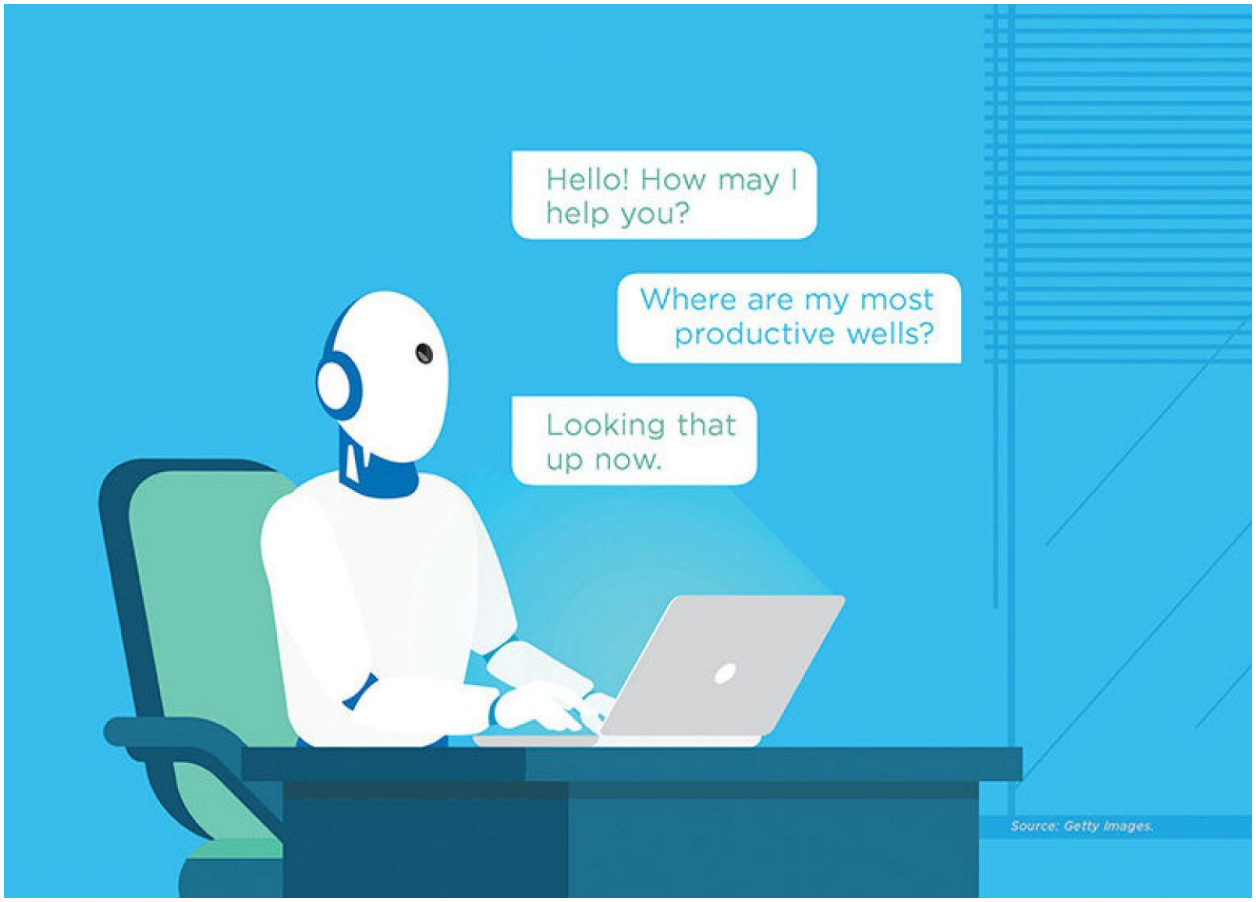


Chat Bot Using Python

LOVELY PROFESSIONAL UNIVERSITY

Project: Chatbot Using Python

Submitted By

Name: Roop Shree
Eshant Das

Reg. No: 12003336
12002058

Section: K20KP

Submitted To

Dr. Sukhvir Kaur

In partial fulfillment of the requirements of the degree of

INT213

1. 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. Chatbots can be as simple as rudimentary programs that answer a simple query with a single-line response, or as sophisticated as digital assistants that learn and evolve to deliver increasing levels of personalization as they gather and process information.

We all have probably interacted with a chatbot whether we know it or not. For example, we're at our computer researching a product, and a window pops up on the screen asking if you need help. Or perhaps we're on our way to a concert and we use your smartphone to request a ride via chat. Or we might have used voice commands to order a coffee from a neighborhood café and received a response telling us when our order will be ready and what it will cost. These are all examples of scenarios in which we could be encountering a chatbot.

In this project, we have created a simple python chatbot that gives responses according to the question/message sent to the bot. From greeting to having a small conversation. In the case of not being able to give a correct response, it refuses. Having the functionalities of text preprocessing, response generating 7 defining the conversation.

2. How do chatbots work?

Driven by AI, automated rules, natural-language processing (NLP), and machine learning (ML), chatbots process data to deliver responses to requests of all kinds. There are two main types of chatbots.

- **Task-oriented (declarative) chatbots** are single-purpose programs that focus on performing one function. Using rules, NLP, and very little ML, they generate automated but conversational responses to user inquiries. Interactions with these chatbots are highly specific and structured and are most applicable to support and service functions—think robust interactive FAQs. Task-oriented chatbots can handle common questions, such as queries about hours of business or simple transactions that don't involve a variety of variables.

Though they do use NLP so end users can experience them in a conversational way, their capabilities are fairly basic. These are currently the most commonly used chatbots.

- **Data-driven and predictive (conversational) chatbots** are often referred to as virtual assistants or digital assistants, and they are much more sophisticated, interactive, and personalized than task-oriented chatbots. These chatbots are contextually aware and leverage natural-language understanding (NLU), NLP, and ML to learn as they go. They apply predictive intelligence and analytics to enable personalization based on user profiles and past user behavior. Digital assistants can learn a user's preferences over time, provide recommendations, and even anticipate needs. In addition to monitoring data and intent, they can initiate conversations. Apple's Siri and Amazon's Alexa are examples of consumer-oriented, data-driven, predictive chatbots.

3. Why were chatbots created?

Digitization is transforming society into a "mobile-first" population. As messaging applications grow in popularity, chatbots are increasingly playing an important role in this mobility-driven transformation. Intelligent conversational chatbots are often interfaces for mobile applications and are changing the way businesses and customers interact.

4. Common chatbot uses

- Chatbots Answer Questions and Inquiries
- Book Tickets to Events/Shows with Chatbots
- Use Chatbots to Find Products, Check Inventory and Recommend Items
- Chatbots To Build Remarkable Customer Experience
- Chatbots Can Process Return and Exchange Requests
- Chatbots Can Confirm Orders and Track Shipping
- Chatbots, Help You Collect Customer Feedback Efficiently
- Chatbots Assign Customer Requests to Support Teams
- Chatbots Generate Leads with Sales Approach

- Chatbots In a Second-Net Strategy
- Chatbots, Help You Build Email Lists More Effectively
- Chatbots Help Simplify Pricing
- Chatbots Promote Products with Fun Conversations
- Chatbot Helps You Improve your Funnel Marketing
- Chatbots Can Do Quizzes, Promotions, And Contests with Customers
- Chatbots Introduce New Products/Services
- Equip Chatbots with Upsell, Down sells, and Cross-sell Tactics
- Chatbots Can Execute Drip Campaigns
- Chatbots Improve Your Product Onboarding
- Chatbots Demo and Show Requested Products

Team members with roles:

Eshant- coding of the project, adding libraries, modules.
Roop Shree- Researched what chatbot is, went through some references, created the report.

Libraries:

- **NumPy**
  NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. NumPy is a Python package. It stands for 'Numerical Python.
  NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences.

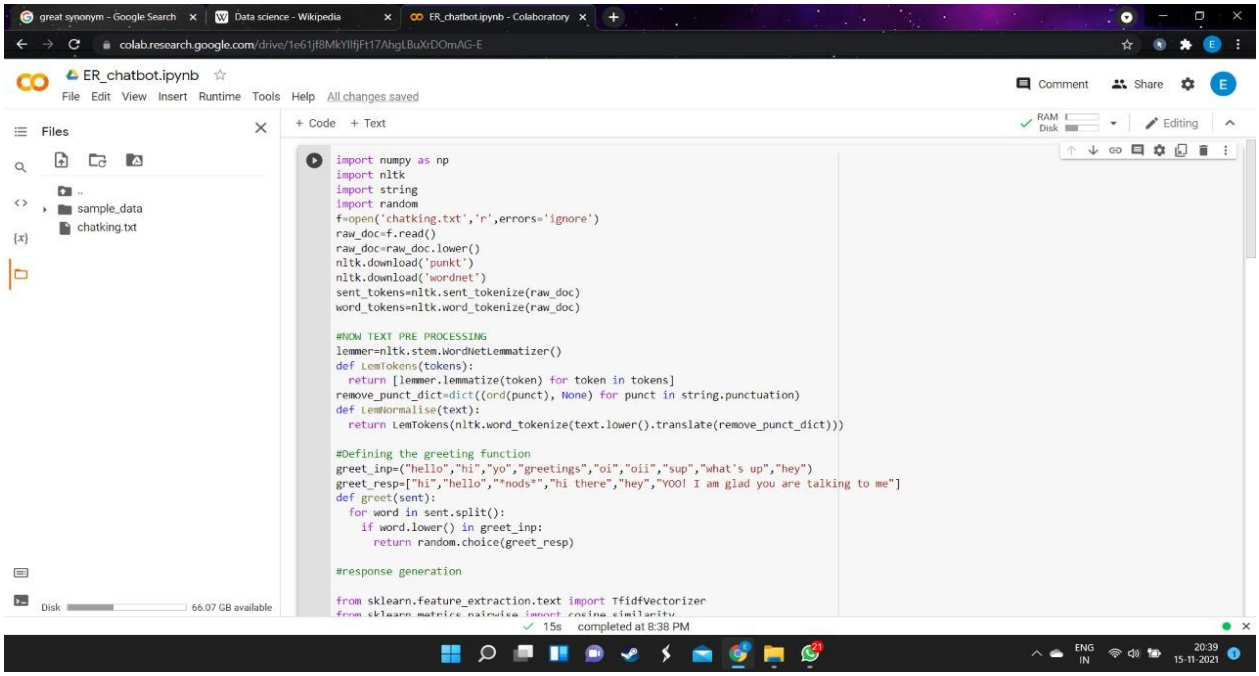
- **Nltk**
  NLTK is a standard python library with prebuilt functions and utilities for ease of use and implementation. It is one of the most used libraries for natural language processing and computational linguistics.
  NLTK is a toolkit built for working with NLP in Python. It provides us with various text processing libraries with a lot of test datasets. A variety of tasks can be performed using NLTK such as tokenizing, parse tree visualization, etc…

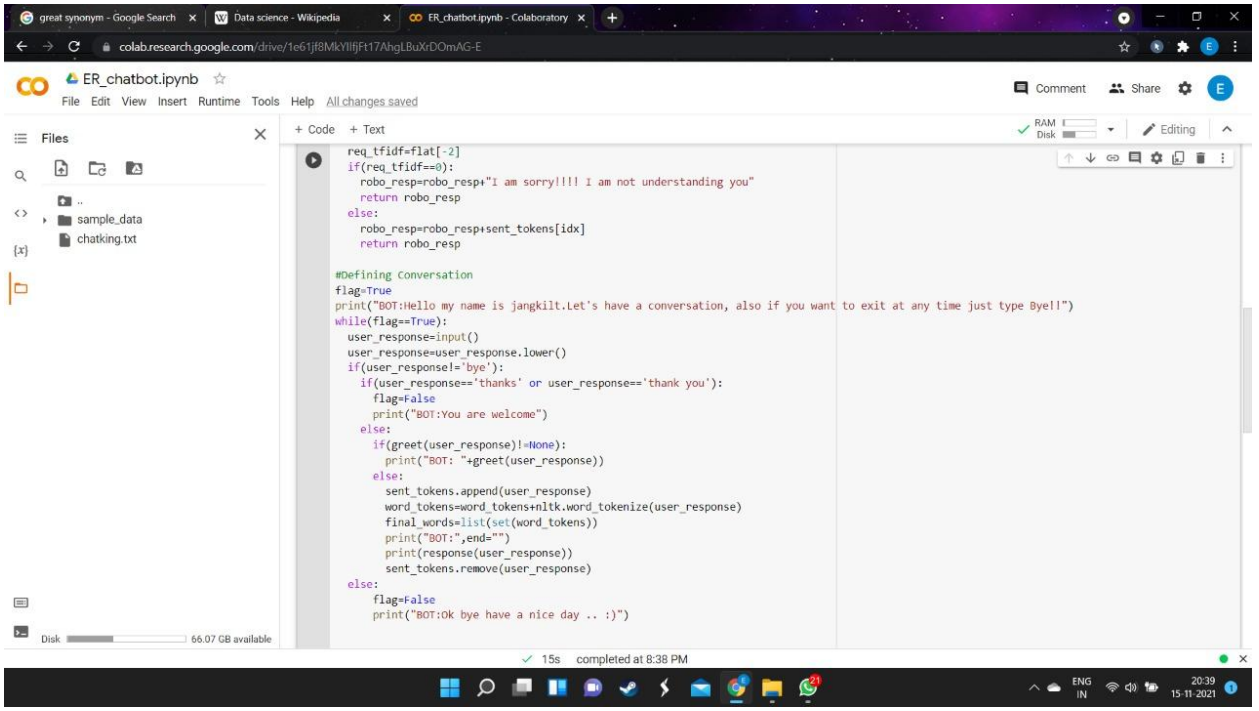
Modules Used:

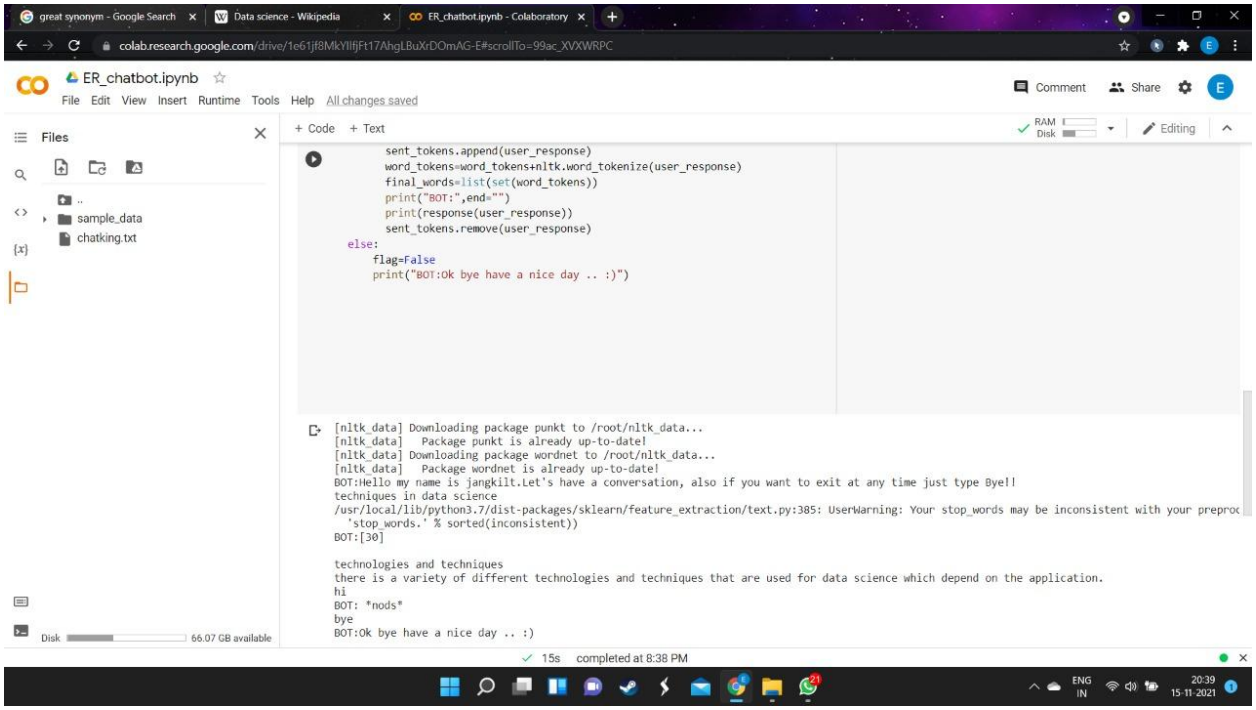
- **Random**
  Python Random module is an in-built module of Python which is used to generate random numbers. These are pseudo-random numbers means these are not truly random. This module can be used to perform random actions such as generating random numbers, printing random a value for a list or string, etc.
  The random module in python contains two interfaces(classes) of pseudorandom number generators (PRNGs). You can view it as two ways to generate random numbers. System Random uses either the /dev/random file on POSIX systems or the CryptGenRandom() function on Windows NT systems. Both are Cryptographically secure PRNGs.

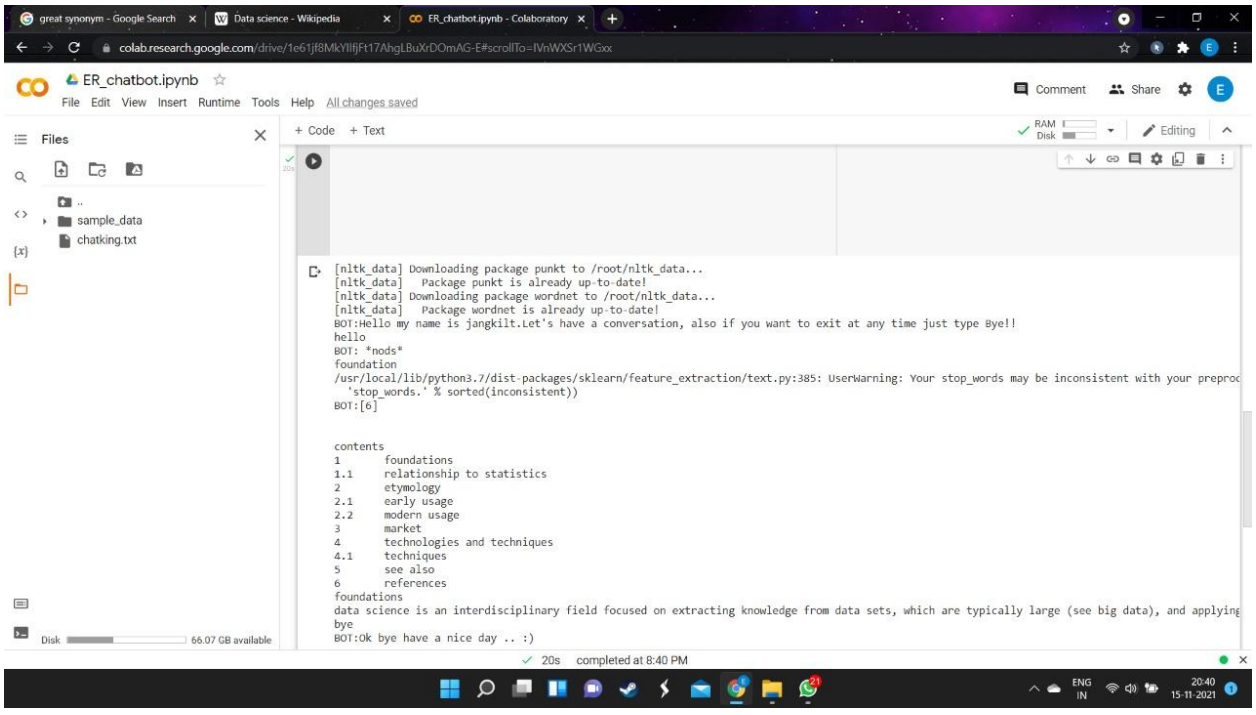
- **String**
  The string module contains a number of functions to process standard Python strings, as shown in Example 1-51. ... Many of the functions in the string module are simply wrapper functions that call the corresponding string method.
  To create a string, put the sequence of characters inside either single quotes, double quotes, or triple quotes and then assign it to a variable. You can look into how variables work in Python in the Python variables tutorial. For example, you can assign a character 'a' to a variable single quote character.

Screenshots:


great synonym - Google Search
Data science - Wikipedia
ER_chatbot.ipynb - Colaboratory
colab.research.google.com/drive/1e61jf8MkYlIjFt17AhgLBuXrDOmAG-E
ER_chatbot.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
sample_data
chatking.txt
Disk 66.07 GB available
+ Code + Text
RAM
Disk
Editing
import numpy as np
import nltk
import string
import random
f=open('chatking.txt','r',errors='ignore')
raw_doc=f.read()
raw_doc=raw_doc.lower()
nltk.download('punkt')
nltk.download('wordnet')
sent_tokens=nltk.sent_tokenize(raw_doc)
word_tokens=nltk.word_tokenize(raw_doc)
#NOW TEXT PRE PROCESSING
lemmer=nltk.stem.WordNetLemmatizer()
def LemTokens(tokens):
return [lemmer.lemmatize(token) for token in tokens]
remove_punct_dict=dict((ord(punct), None) for punct in string.punctuation)
def LemNormalise(text):
return LemTokens(nltk.word_tokenize(text.lower().translate(remove_punct_dict)))
#Defining the greeting function
greet_inp=("hello","hi","yo","greetings","oi","oii","sup","what's up","hey")
greet_resp=["hi","hello","*nods*","hi there","hey","YOO! I am glad you are talking to me"]
def greet(sent):
for word in sent.split():
if word.lower() in greet_inp:
return random.choice(greet_resp)
#response generation
from sklearn.feature_extraction.text import TfidfVectorizer
15s completed at 8:38 PM
ENG
IN
20:39
15-11-2021


great synonym - Google Search
Data science - Wikipedia
ER_chatbot.ipynb - Colaboratory
colab.research.google.com/drive/1e61jf8MkYlIjFt17AhgLBuXrDOmAG-E
ER_chatbot.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
sample_data
chatking.txt
Disk 66.07 GB available
+ Code + Text
RAM
Disk
Editing
req_tfidf=flat[-2]
if(req_tfidf==0):
robo_resp=robo_resp+"I am sorry!!! I am not understanding you"
return robo_resp
else:
robo_resp=robo_resp+sent_tokens[idx]
return robo_resp
#Defining Conversation
flag=True
print("BOT:Hello my name is jangkilt.Let's have a conversation, also if you want to exit at any time just type Bye!!")
while(flag==True):
user_response=input()
user_response=user_response.lower()
if(user_response!='bye'):
if(user_response=='thanks' or user_response=='thank you'):
flag=False
print("BOT:You are welcome")
else:
if(greet(user_response)!=None):
print("BOT: "+greet(user_response))
else:
sent_tokens.append(user_response)
word_tokens=word_tokens+nltk.word_tokenize(user_response)
final_words=list(set(word_tokens))
print("BOT:",end="")
print(response(user_response))
sent_tokens.remove(user_response)
else:
flag=False
print("BOT:Ok bye have a nice day .. :)")
15s completed at 8:38 PM
ENG
IN
20:39
15-11-2021


great synonym - Google Search
Data science - Wikipedia
ER_chatbot.ipynb - Colaboratory
colab.research.google.com/drive/1e61jf8MkYlIjFt17AhgLBuXrDOmAG-E#scrollTo=99ac_XVXWRPC
ER_chatbot.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
sample_data
chatking.txt
+ Code + Text
RAM
Disk
Editing
sent_tokens.append(user_response)
word_tokens=word_tokens+nltk.word_tokenize(user_response)
final_words=list(set(word_tokens))
print("BOT:",end="")
print(response(user_response))
sent_tokens.remove(user_response)
else:
flag=False
print("BOT:Ok bye have a nice day .. :)")
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Package wordnet is already up-to-date!
BOT:Hello my name is jangkilt.Let's have a conversation, also if you want to exit at any time just type Bye!!
techniques in data science
/usr/local/lib/python3.7/dist-packages/sklearn/feature_extraction/text.py:385: UserWarning: Your stop_words may be inconsistent with your preproc
'stop_words.' % sorted(inconsistent))
BOT:[30]
technologies and techniques
there is a variety of different technologies and techniques that are used for data science which depend on the application.
hi
BOT: *nods*
bye
BOT:Ok bye have a nice day .. :)
Disk
66.07 GB available
15s completed at 8:38 PM
ENG
IN
20:39
15-11-2021


great synonym - Google Search
Data science - Wikipedia
ER_chatbot.ipynb - Colaboratory
colab.research.google.com/drive/1e61jf8MkYlIjFt17AhgLBuXrDOmAG-E#scrollTo=IVnWXSr1WGxx
ER_chatbot.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
sample_data
chatking.txt
+ Code + Text
RAM
Disk
Editing
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Package wordnet is already up-to-date!
BOT:Hello my name is jangkilt.Let's have a conversation, also if you want to exit at any time just type Bye!!
hello
BOT: *nods*
foundation
/usr/local/lib/python3.7/dist-packages/sklearn/feature_extraction/text.py:385: UserWarning: Your stop_words may be inconsistent with your preproc
'stop_words.' % sorted(inconsistent))
BOT:[6]
contents
1 foundations
1.1 relationship to statistics
2 etymology
2.1 early usage
2.2 modern usage
3 market
4 technologies and techniques
4.1 techniques
5 see also
6 references
foundations
data science is an interdisciplinary field focused on extracting knowledge from data sets, which are typically large (see big data), and applying
bye
BOT:Ok bye have a nice day .. :)
Disk
66.07 GB available
20s completed at 8:40 PM
ENG
IN
20:40
15-11-2021

References:

Edureka U-tube channel: https://www.youtube.com/watch?v=tSjR7bk1Y9U

Great learning: https://www.youtube.com/watch?v=c_gXrw1RoKo

Fun with data science: https://www.youtube.com/watch?v=ko7KKy5WB-Y

Read the Docs: https://readthedocs.org/projects/chatterbot/downloads/pdf/latest/

Academia:
https://www.academia.edu/40419686/Building_Chatbots_with_Python_Using_Natural_Language_Processing_and_Machine_Learning_Sumit_Raj