A MINI PROJECT REPORT ON

Text-Based ChatBot

Submitted by

Vedant D. Mulherkar **Roll No. 06 -Div- 'B'**

Avinash P. Mankar **Roll No. 05 -Div- 'B'**

Ganesh C. Vispute **Roll No. 14 -Div- 'B'**

Submitted in partial fulfilment of the requirements

for the award of the degree of

Bachelor in

Computer Engineering

For the Subject

Subject: Artificial Intelligence and Robotics (410242)

For Academic Year 2021-22

MET Bhujbal Knowledge City

Institute of Engineering

Department of Computer Engineering

MET Bhujbal Knowledge City

Institute of Engineering

Department of Computer Engineering

Certificate

This is to Certify that

Vedant D. Mulherkar Roll No. 05 -Div- 'B'

Avinash P. Mankar Roll No. 06 -Div- 'B'

Ganesh C. Vispute Roll No. 14 -Div- 'B'

has completed the necessary Mini Project work and prepare the report on

Text-Based ChatBot

in satisfactory manner as a fulfillment of the requirement of the award of

degree of Bachelor of Computer Engineering in the Academic year

2021-2022

For the Subject

Artificial Intelligence and Robotics (410242)

Subject Teacher	HOD
Prof. Vaibhav Dhabada	Dr. M. U. Kharat

Acknowledgements

Every work is source which requires support from many people and areas. It gives me proud privilege to complete the Mini Project on "**Text-Based ChatBot**" under valuable guidance and encouragement of our subject teacher Prof. Vaibhav Dhabada Sir.

I am also extremely grateful to our respected H.O.D. Dr. M. U. Kharat and Principal Dr. V. P. Wani for providing all facilities and every help for smooth progress of this Mini Project.

At last, I would like to thank all the Staff members and my Colleagues who directly or indirectly supported me without which this work would not have been completed successfully.

By,

Vedant D. Mulherkar

Avinash P. Mankar

Ganesh C. Vispute

INDEX

INTRODUCTION

In the era of Artificial Intelligence every field want a system which can fulfill an ability to answer without any human support with 100% efficiency and no errors , every industry wants an agent which solves their customers queries without any compliments.
For such types of problems a system called chatbot was introduced where Chatbots are way more than simple conversational agents , which deals with the regularly asked questions without any error.
Chatbots are AI based computer programs that simulate human conversations. They are also known as digital assistants that understand human capabilities. Bots interpret the user intent, process their requests, and give prompt relevant answers.

Bots can communicate through voice as well as text and can be deployed across websites, applications, and messaging channels such as Facebook Messenger, Twitter, or Whatsapp.

II. What are chatbots?

1. Chatbots are simulations which can understand human language, process it and interact back with humans while performing specific tasks.
2. They can be connected to various APIs which will for example enable them to deal with a wider range of customers' requests.
3. They can also automate repetitive and tedious tasks like emailing prospects and customers, answering FAQs and many more.

III. History of chatbots

The first chatbot was created by Joseph Wiesenbaum in 1966, named Eliza.

1. Eliza – 1966
2. Parry – 1972
3. A.L.I.C.E – 1995
4. Smarter Child – 2001
5. SIRI – 2010
6. Google Now – 2012
7. Alexa - 2015

IV. Types of Chatbots

Important types:

1. Text based chatbot.

   A type of chatbot where the primary mode of communication is texting(i.e. chatting). Yes, this may involve media like images and videos, and UI elements like Quick Replies, Carousels, and more as well.

2. Voice based chatbot.

   Voice chatbots are **chatbots that can communicate using vocal input and output**. You can talk to the bot out loud, like you would a person. In turn, it will respond with a voice of its own. Chatbots of any kind are software programs that interact via conversation. Usually, this works through text-based input and output.

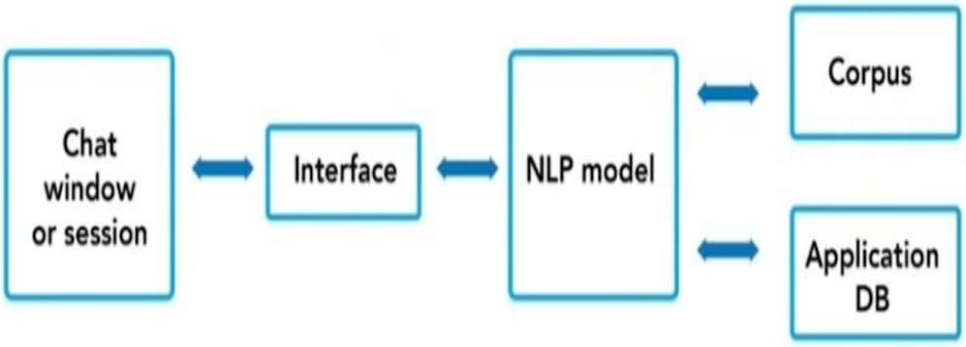
V. Top applications of chatbots

1. Helpdesk assistant
2. Email distributor
3. Home assistant
4. Operations assistant
5. Phone assistant
6. Entertainment assistant
7. Mobile assistant
8. Searching assistant
9. Navigation assistant
10. Virtual friend.

VI. Architecture of chatbots:

A typical chatbot architecture should consist of :

1. Chat window or session: The window through which chatbot interacts with us.
2. Interface: Interaction between user and chatbot.
3. NLP model: Natural Language Processing the chatbot convers our language to machine language for understanding and vice-versa so that we understand it.
4. I. Corpus
   ii Application DB



VII. HOW DOES A CHATBOT WORKS

A. Chatbots are designed using these approaches:

1. **Rule-based Chatbot:**

Rule-based chatbots also referred to as decision-tree bots, use a series of defined rules. These rules are the basis for the types of problems the chatbot is familiar with and can deliver solutions for.

Key attributes or rule based chatbots:

- These bots follow predetermined rules. So it becomes easy to use the bot for simpler scenarios.
- Interactions with rule based chatbots are highly structured and are most applicable to customer support functions.
- Rule based bots are ideally suitable for answering common queries such as an inquiry about business hours, delivery status, or tracking details.

2. **Self-learning Chatbot:**

   I . Bot that learns how to communicate using the result of a machine learning model to learn and assess current situations.

   Ii. the power of machine learning and NLP to understand the context and intent of a question before formulating a response.

   Iii. They are driven by chatbot scripts and generate their own answers to more complicated questions using natural-language responses.

   Iiv. The more you use and train these bots, the more they learn and the better they operate with the user.

Iv. The conversational communication skills of the chatbot technology empower them to deliver what customers are looking for.

Key attributes of AI enabled chatbots:

- Conversational bots can understand the context and intent of complex conversations and try to provide more relevant answers.
- AI bots apply predictive intelligence and sentiment analysis to understand customer emotions closely.
- Machine learning bots learn from user behavior and provide more personalized conversations.

B. Importing steps for the chatbot to process:

1 Import corpus.
2. Preprocess the data.
3. Text case handling.
4. Tokenization.
5. Stemming
6. Bag of words (BOW)
7. One hot encoding.

I) **Corpus:**

1. Corpus is the training data needed for the chatbot to learn.

2. Without a corpus,it is impossible for a chatbot to learn and reply to something useful back to the user.

II) **Data Preprocessing- text case handling:**

1. Convert all the data coming as an input to either upper or lower case.
2. This will avoid misrepresentation and misinterpretation of words if spelt under lower or upper cases.

III) **Tokenization:**

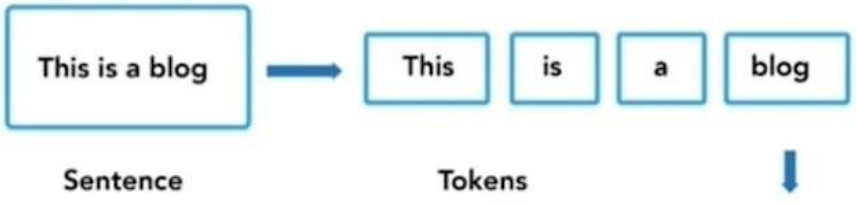
Tokenization is the structure process of converting a sentence into an individual collection of words.

IV) **Stemming** Stemming is a process of finding similarities between words with the same root words.

Original word	Root word	Similar words
Jump	Jump	Word with similar root word i.e. JUMP
Jumped	Jump	
Jumps	Jump	
Jumping	Jump	

V) **Generating Bag Of Words(BOW):**

Process of converting words into numbers by generating vector embeddings from the tokens generated.



Original word	BOW Vector			
This	1	0	0	0
is	0	1	0	0
a	0	0	1	0
blog	0	0	0	1

VI) **One hot encoding:**

One hot encoding is a process by which categorical variables are converting into the form that ML algorithms use.

Tag	One Hot encoded vector [11X11]										
This	1	0	0	0	0	0	0	0	0	0	0
is	0	1	0	0	0	0	0	0	0	0	0
a	0	0	1	0	0	0	0	0	0	0	0
blog	0	0	0	1	0	0	0	0	0	0	0
name	0	0	0	1	0	0	0	0	0	0	1

VIII. IMPLEMENTATION

Example of corpus:

artificial intelligence (AI), the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings. The term is frequently applied to the project of developing systems endowed with the intellectual processes characteristic of humans, such as the ability to reason, discover meaning, generalize, or learn from past experience. Since the development of the digital computer in the 1940s, it has been demonstrated that computers can be programmed to carry out very complex tasks—as, for example, discovering proofs for mathematical theorems or playing chess—with great proficiency. Still, despite continuing advances in computer processing speed and memory capacity, there are as yet no programs that can match human flexibility over wider domains or in tasks requiring much everyday knowledge. On the other hand, some programs have attained the performance levels of human experts and professionals in performing certain specific tasks, so that artificial intelligence in this limited sense is found in applications as diverse as medical diagnosis, computer search engines, and voice or handwriting recognition.

What is intelligence?
All but the simplest human behaviour is ascribed to intelligence, while even the most complicated insect behaviour is never taken as an indication of intelligence. What is the difference? Consider the behaviour of the digger wasp, Sphex ichneumoneus. When the female wasp returns to her burrow with food, she first deposits it on the threshold, checks for intruders inside her burrow, and only then, if the coast is clear, carries her food inside. The real nature of the wasp's instinctual behaviour is revealed if the food is moved a few inches away from the entrance to her burrow while she is inside: on emerging, she will repeat the whole procedure as often as the food is displaced. Intelligence—conspicuously absent in the case of Sphex—must include the ability to adapt to new circumstances.

Psychologists generally do not characterize human intelligence by just one trait but by the combination of many diverse abilities. Research in AI has focused chiefly on the following components of intelligence: learning, reasoning, problem solving, perception, and using language.

Learning
There are a number of different forms of learning as applied to artificial intelligence. The simplest is learning by trial and error. For example, a simple computer program for solving mate-in-one chess problems might try moves at random until mate is found. The program might then store the solution with the position so that the next time the computer encountered the same position it would recall the solution. This simple memorizing of individual items and procedures—known as rote learning—is relatively easy to implement on a computer. More challenging is the problem of implementing what is called generalization. Generalization involves applying past experience to analogous new situations. For example, a program that learns the past tense of regular English verbs by rote will not be able to produce the past tense of a word such as jump unless it previously had been presented with jumped, whereas a program that is able to generalize can learn the "add ed" rule and so form the past tense of jump based on experience with similar verbs.

AI PROJECT ON CHAT_BOT BY:

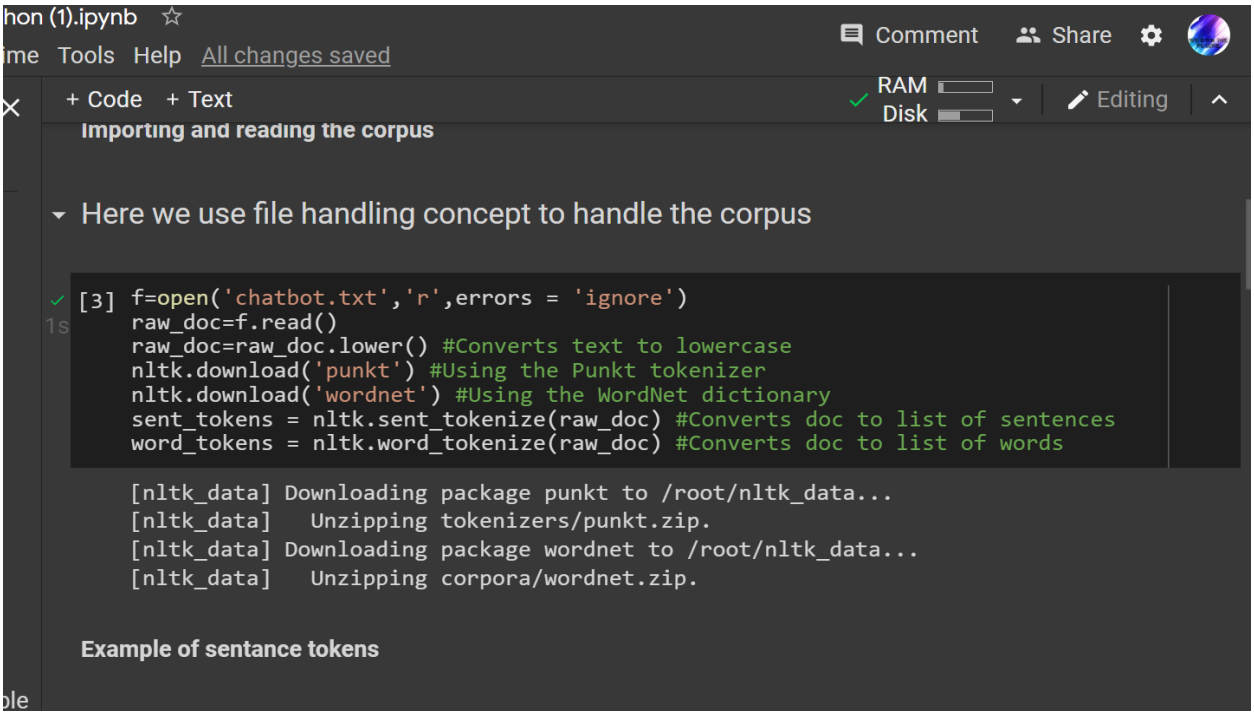
VEDANT MULHERKAR BE 6-B

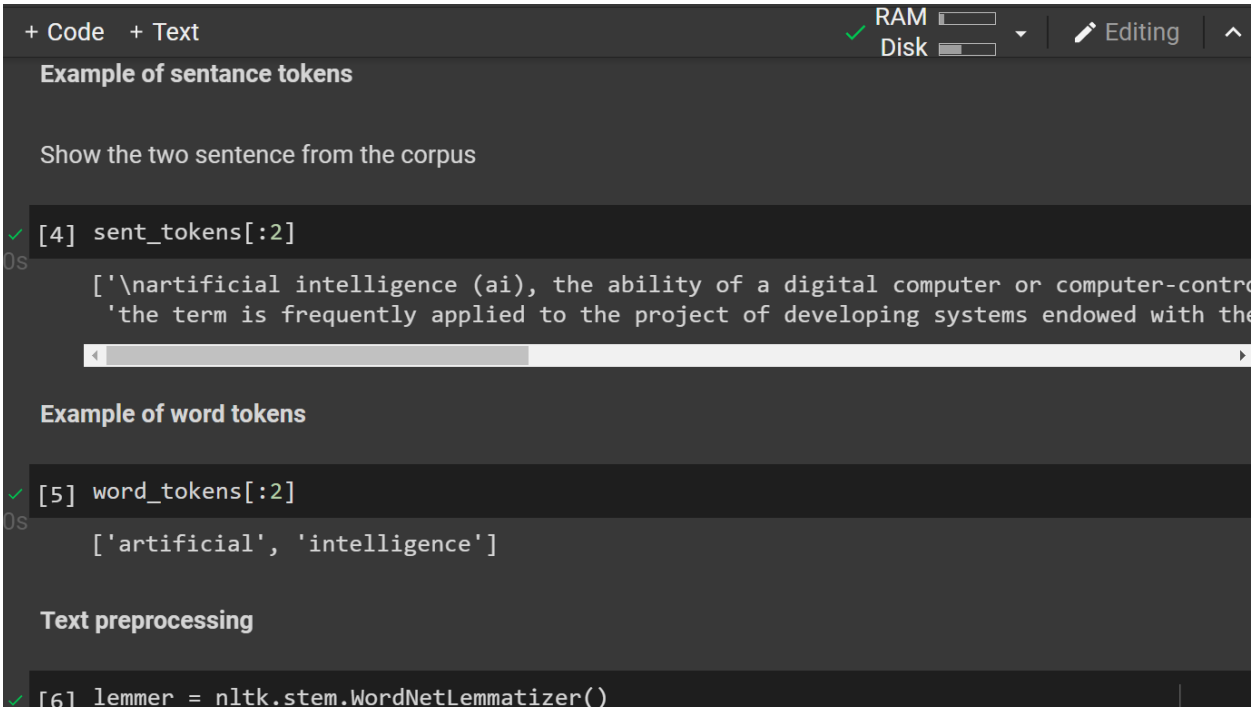
AVINASH MANKAR BE 5-B

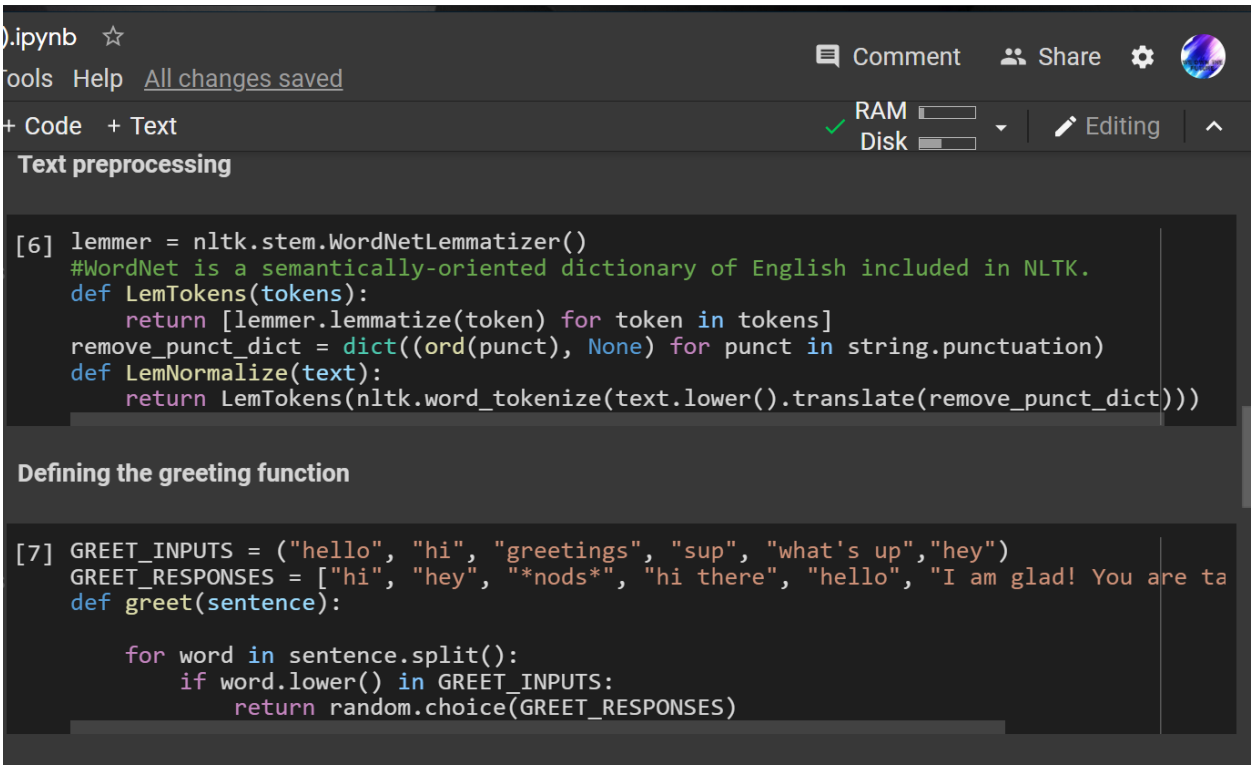
GANESH VISPUTE BE 14-B

Importing the required libraries

```
[2] import numpy as np   ## use for datastructure
    import nltk           ## it is natural language processing library
    import string         ## it is a library used to handle python string
    import random         ## use for machine learning
```




+ Code + Text
RAM
Disk
Editing
Example of sentance tokens
Show the two sentence from the corpus
[4] sent_tokens[:2]
['\nartificial intelligence (ai), the ability of a digital computer or computer-contro
'the term is frequently applied to the project of developing systems endowed with the
Example of word tokens
[5] word_tokens[:2]
['artificial', 'intelligence']
Text preprocessing
[6] lemmer = nltk.stem.WordNetLemmatizer()


.ipynb
Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
Text preprocessing
[6] lemmer = nltk.stem.WordNetLemmatizer()
#WordNet is a semantically-oriented dictionary of English included in NLTK.
def LemTokens(tokens):
return [lemmer.lemmatize(token) for token in tokens]
remove_punct_dict = dict((ord(punct), None) for punct in string.punctuation)
def LemNormalize(text):
return LemTokens(nltk.word_tokenize(text.lower().translate(remove_punct_dict)))
Defining the greeting function
[7] GREET_INPUTS = ("hello", "hi", "greetings", "sup", "what's up","hey")
GREET_RESPONSES = ["hi", "hey", "*nods*", "hi there", "hello", "I am glad! You are ta
def greet(sentence):
for word in sentence.split():
if word.lower() in GREET_INPUTS:
return random.choice(GREET_RESPONSES)

Response generation

```
[8] from sklearn.feature_extraction.text import TfidfVectorizer
    from sklearn.metrics.pairwise import cosine_similarity
```

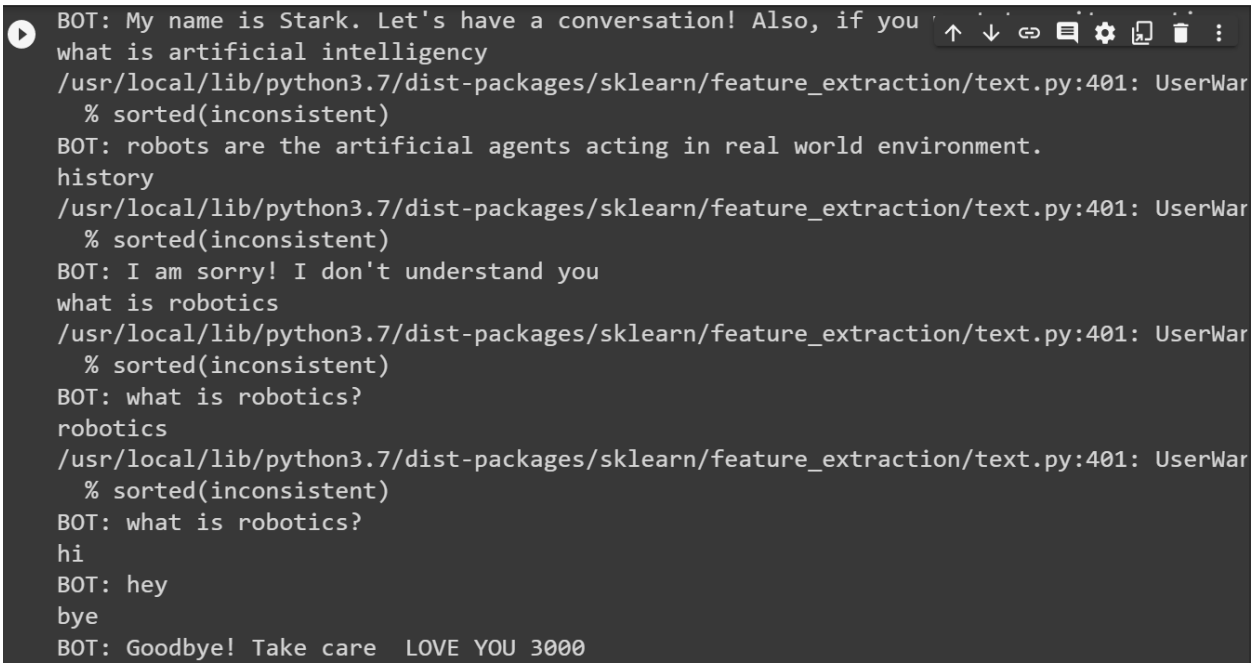
```
[9] def response(user_response):
      robo1_response=''
      TfidfVec = TfidfVectorizer(tokenizer=LemNormalize, stop_words='english')
      tfidf = TfidfVec.fit_transform(sent_tokens)
      vals = cosine_similarity(tfidf[-1], tfidf)
      idx=vals.argsort()[0][-2]
      flat = vals.flatten()
      flat.sort()
      req_tfidf = flat[-2]
      if(req_tfidf==0):
        robo1_response=robo1_response+"I am sorry! I don't understand you"
        return robo1_response
      else:
        robo1_response = robo1_response+sent_tokens[idx]
        return robo1_response
```

Disk

```
[9]     else:
          robo1_response = robo1_response+sent_tokens[idx]
          return robo1_response
```

Defining conversation start/end protocols

```
[14] flag=True
     print("BOT: My name is Stark. Let's have a conversation! Also, if you want to exit an
     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="")
```


BOT: My name is Stark. Let's have a conversation! Also, if you
what is artificial intelligency
/usr/local/lib/python3.7/dist-packages/sklearn/feature_extraction/text.py:401: UserWar
% sorted(inconsistent)
BOT: robots are the artificial agents acting in real world environment.
history
/usr/local/lib/python3.7/dist-packages/sklearn/feature_extraction/text.py:401: UserWar
% sorted(inconsistent)
BOT: I am sorry! I don't understand you
what is robotics
/usr/local/lib/python3.7/dist-packages/sklearn/feature_extraction/text.py:401: UserWar
% sorted(inconsistent)
BOT: what is robotics?
robotics
/usr/local/lib/python3.7/dist-packages/sklearn/feature_extraction/text.py:401: UserWar
% sorted(inconsistent)
BOT: what is robotics?
hi
BOT: hey
bye
BOT: Goodbye! Take care LOVE YOU 3000