Guiller Gamata
NetID: xe5599

Assignment 4: Chatterbot

Abstract

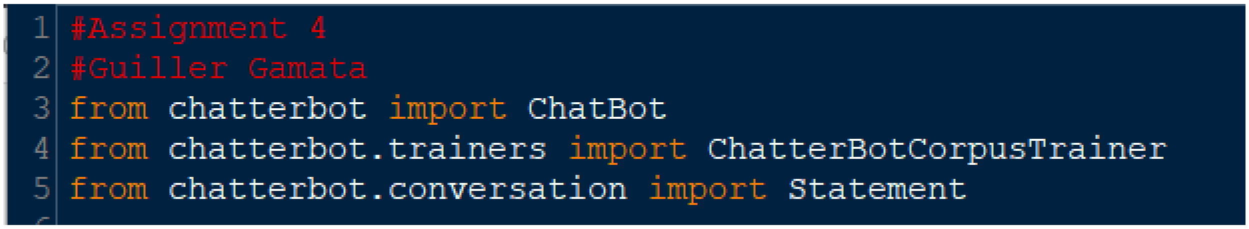
We will be using Python and the Chatterbot library in order to create an intelligent agent, a chatbot, and demonstrate the practical use of the chatbot, as well as illustrate the chatbot's goals in a hypothetical scenario.

Introduction

The kind of chatbot in question that I will focus on will be a customer service chatbot, specifically in a hospital scenario. The user can access a computer in the lobby containing the chatbot and may need directions to common or important locations located in the hospital, or the user may also want general information regarding the hospital such as hours, or even the hospital address. The goal of the chatbot is to provide information to the user of this terminal in a reliable manner.

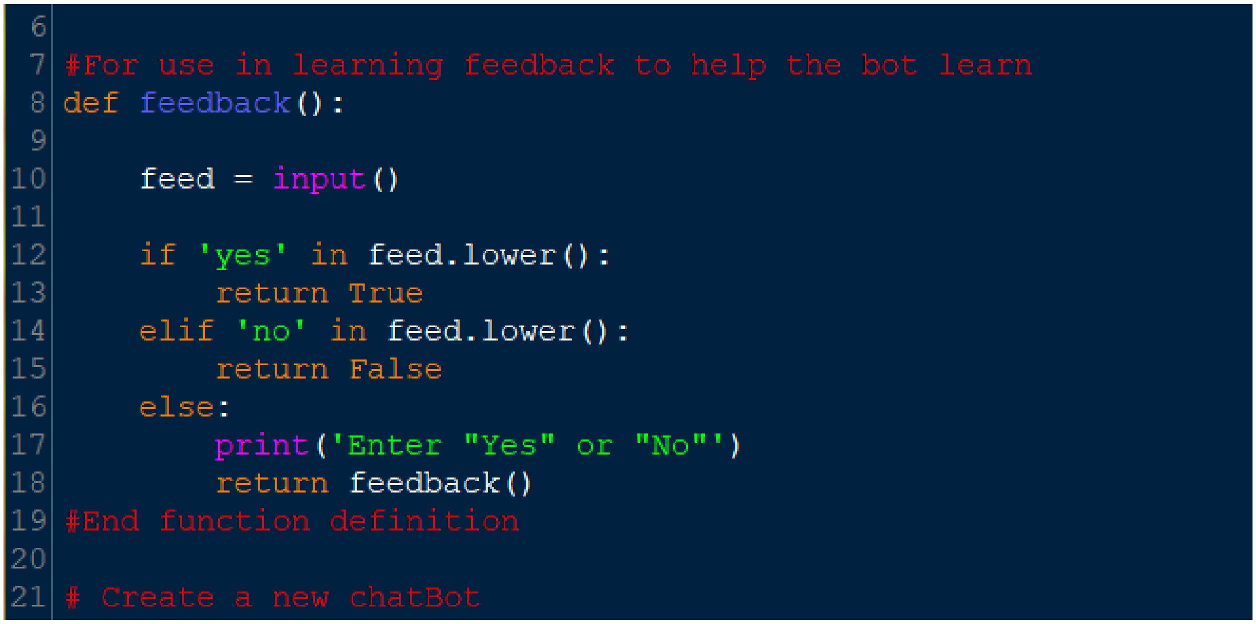
Summary of Results

The python code is shown below that details the libraries used to implement a proper chatbot. This chatbot will utilize the CorpusTrainer to train itself through a custom corpus that I have created as well as a Statement to update responses.



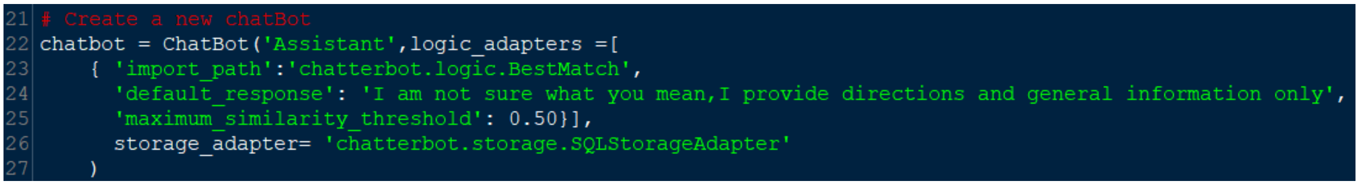
```
#Assignment 4
#Guiller Gamata
from chatterbot import ChatBot
from chatterbot.trainers import ChatterBotCorpusTrainer
from chatterbot.conversation import Statement
```

This chatbot will also allow for user feedback to help accelerate the learning process as well, this is shown in the feedback function which allows us to send feedback if the response was proper or not, this function will be touched upon again later



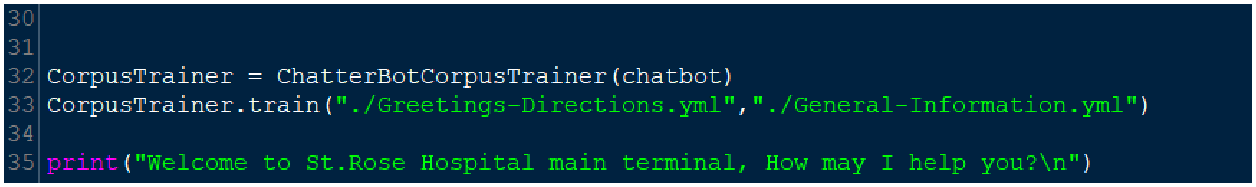
```

#For use in learning feedback to help the bot learn
def feedback():

    feed = input()

    if 'yes' in feed.lower():
        return True
    elif 'no' in feed.lower():
        return False
    else:
        print('Enter "Yes" or "No"')
        return feedback()
#End function definition

# Create a new chatBot
```

A chatbot is instantiated named "Assistant", complete with logic adapters to find the best match for an answer with a similarity threshold of 50% as well as an SQL storage Adapter to update the bot responses database should we find a response unsatisfactory.



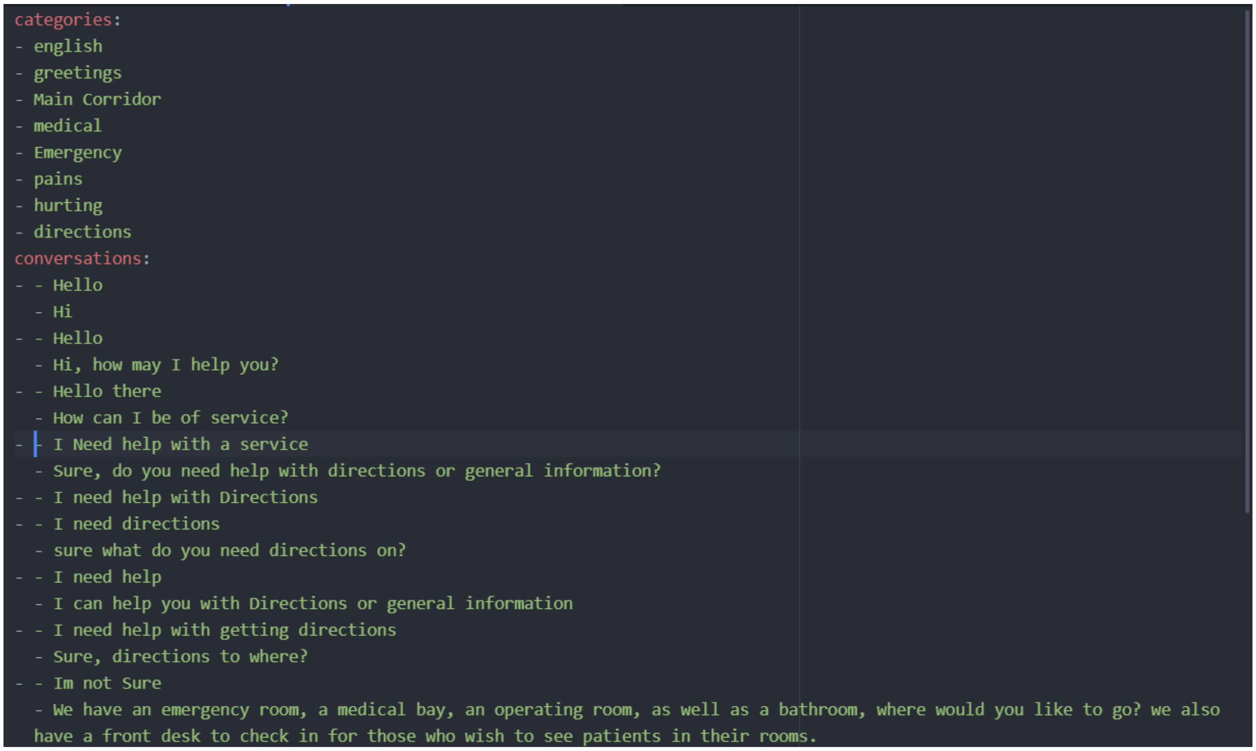
```
# Create a new chatBot
chatbot = ChatBot('Assistant',logic_adapters =[
    { 'import_path':'chatterbot.logic.BestMatch',
      'default_response': 'I am not sure what you mean,I provide directions and general information only',
      'maximum_similarity_threshold': 0.50}],
      storage_adapter= 'chatterbot.storage.SQLStorageAdapter'
    )
```

A CorpusTrainer is instantiated, which will then initially train the chatbot using the corpus that I have created for Greetings, Directions as well as General Information conversations that may occur, the initial welcome text is also printed



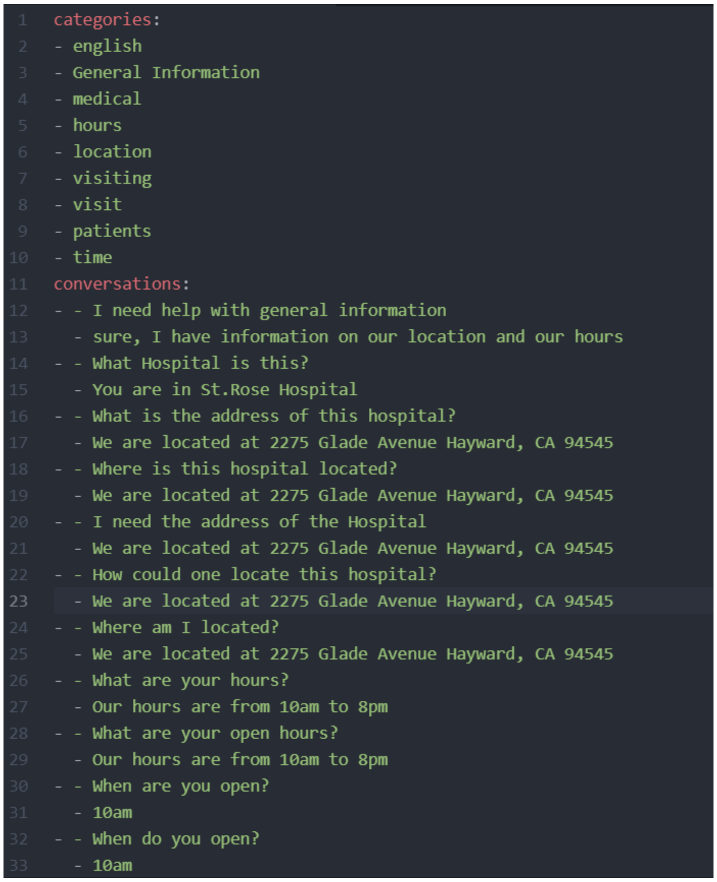
```


CorpusTrainer = ChatterBotCorpusTrainer(chatbot)
CorpusTrainer.train("./Greetings-Directions.yml","./General-Information.yml")

print("Welcome to St.Rose Hospital main terminal, How may I help you?\n")
```

Here are some of the contents of some of the training corpus provided, some answers were repeated more than twice to give the bot a better idea of the variations in the questions that may be asked. The corpus below is intended for greetings as well as answers related to directions, in which the user may ask where to find the medical lab, emergency room, bathroom, or which corridor each room resides in.



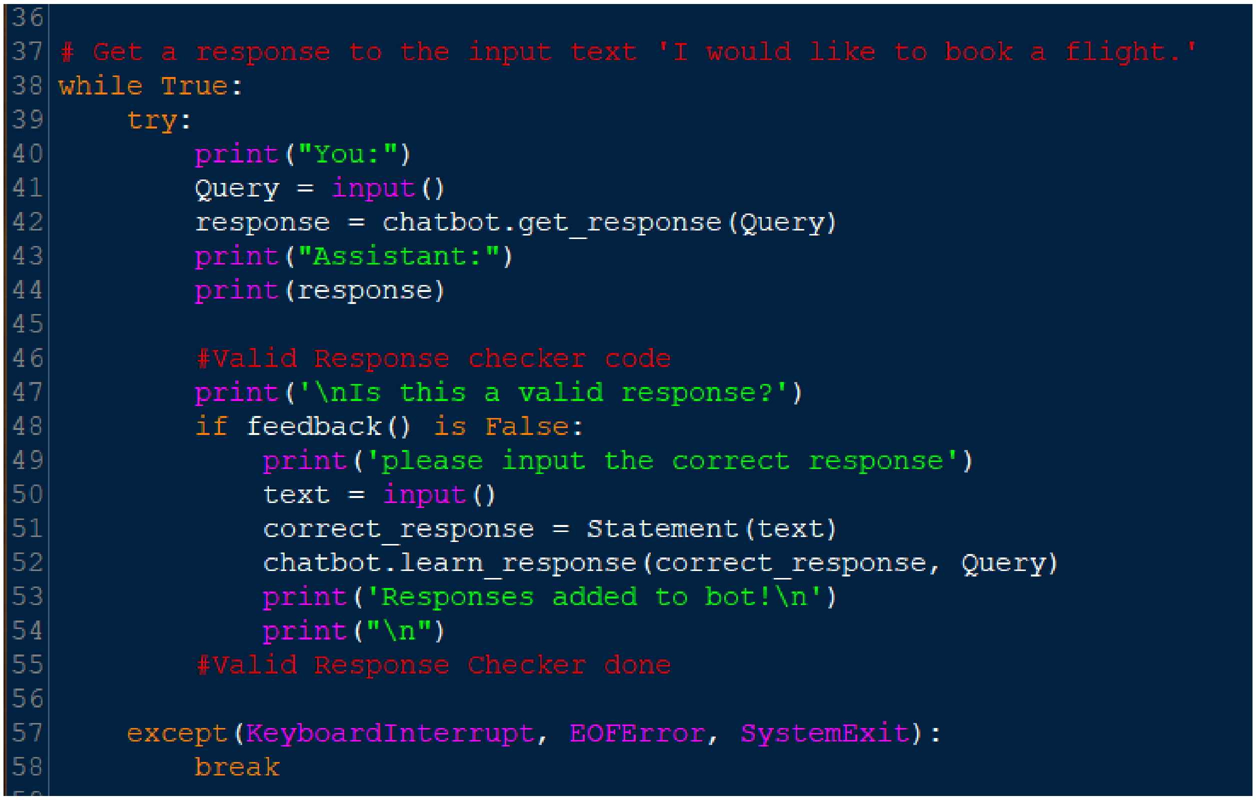
```
categories:
- english
- greetings
- Main Corridor
- medical
- Emergency
- pains
- hurting
- directions
conversations:
- - Hello
  - Hi
- - Hello
  - Hi, how may I help you?
- - Hello there
  - How can I be of service?
- - I Need help with a service
  - Sure, do you need help with directions or general information?
- - I need help with Directions
- - I need directions
  - sure what do you need directions on?
- - I need help
  - I can help you with Directions or general information
- - I need help with getting directions
  - Sure, directions to where?
- - Im not Sure
  - We have an emergency room, a medical bay, an operating room, as well as a bathroom, where would you like to go? we also
  have a front desk to check in for those who wish to see patients in their rooms.
```

Below is a portion of the general-information corpus that helps the bot provide general information about the hospital being attended to, such as hours, visitation, and location.



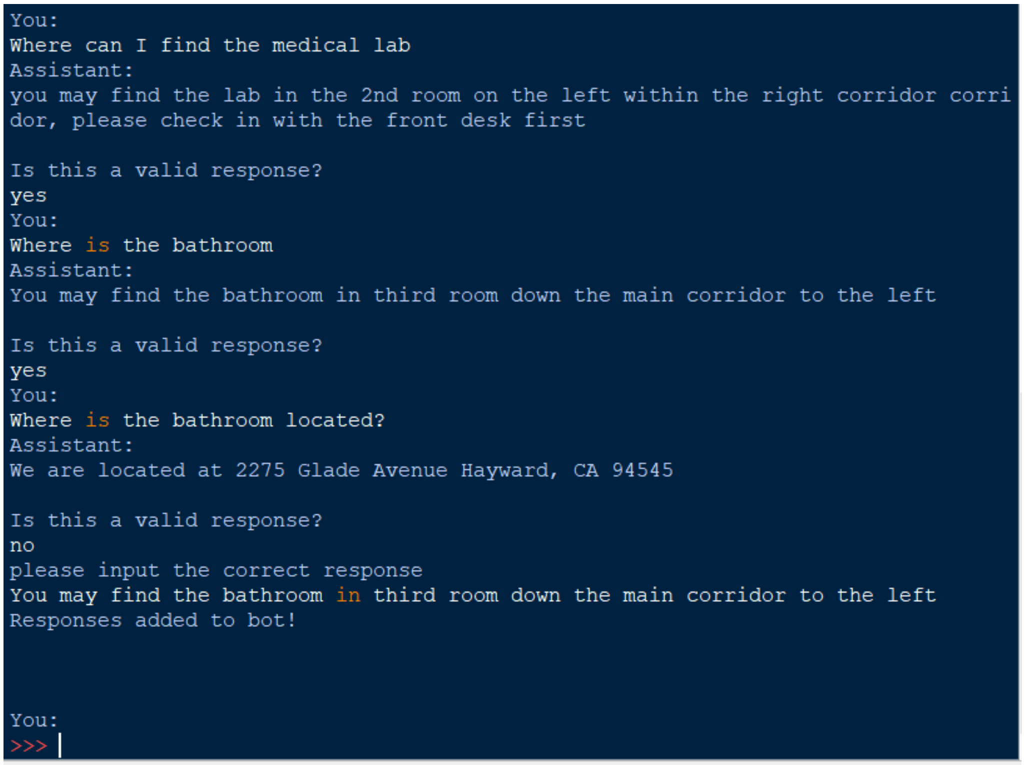
```yaml
categories:
- english
- General Information
- medical
- hours
- location
- visiting
- visit
- patients
- time
conversations:
- - I need help with general information
  - sure, I have information on our location and our hours
- - What Hospital is this?
  - You are in St.Rose Hospital
- - What is the address of this hospital?
  - We are located at 2275 Glade Avenue Hayward, CA 94545
- - Where is this hospital located?
  - We are located at 2275 Glade Avenue Hayward, CA 94545
- - I need the address of the Hospital
  - We are located at 2275 Glade Avenue Hayward, CA 94545
- - How could one locate this hospital?
  - We are located at 2275 Glade Avenue Hayward, CA 94545
- - Where am I located?
  - We are located at 2275 Glade Avenue Hayward, CA 94545
- - What are your hours?
  - Our hours are from 10am to 8pm
- - What are your open hours?
  - Our hours are from 10am to 8pm
- - When are you open?
  - 10am
- - When do you open?
  - 10am
```

In order to talk to the bot, as well as continually train it with conversation, we obtain input from the user and request a response from the bot through get_response() until ctrl+C is input to stop the chatbot, we can then optionally validate this input to help further train the bot, It also appears that I was able to train it via normal conversation without any human validation by commenting out the "valid response checker" portion of the code



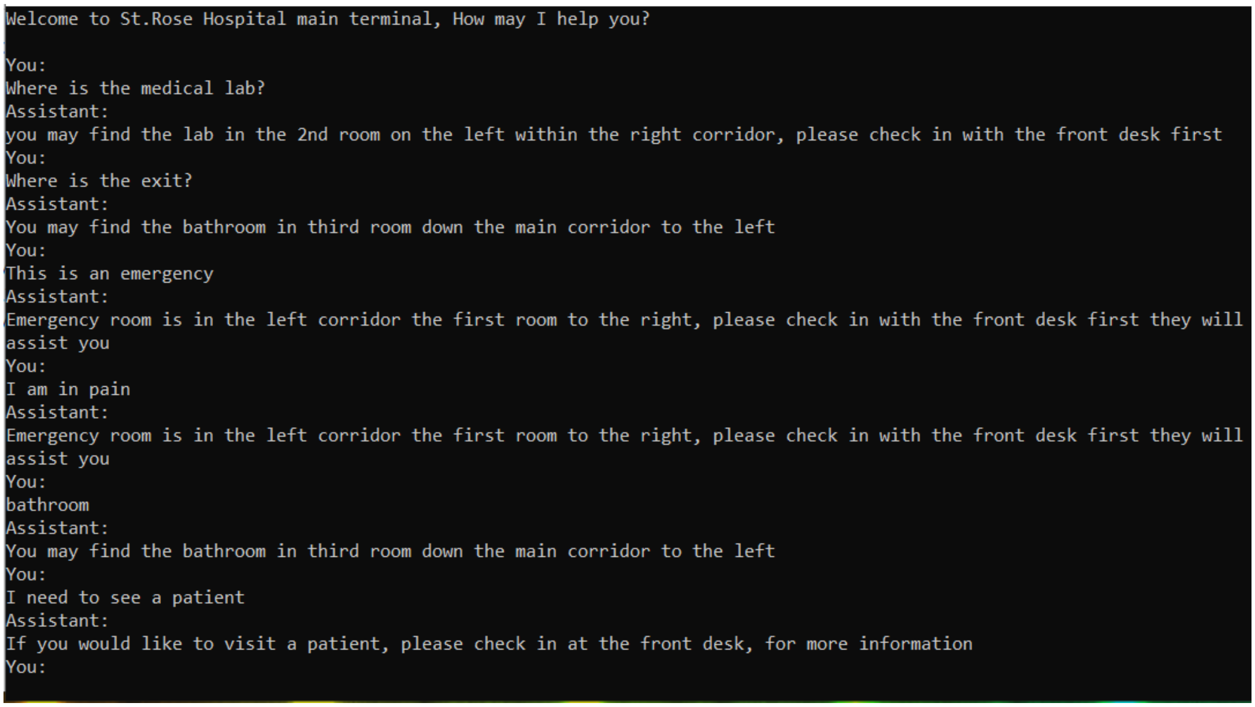
```python

# Get a response to the input text 'I would like to book a flight.'
while True:
    try:
        print("You:")
        Query = input()
        response = chatbot.get_response(Query)
        print("Assistant:")
        print(response)

        #Valid Response checker code
        print('\nIs this a valid response?')
        if feedback() is False:
            print('please input the correct response')
            text = input()
            correct_response = Statement(text)
            chatbot.learn_response(correct_response, Query)
            print('Responses added to bot!\n')
            print("\n")
        #Valid Response Checker done

    except(KeyboardInterrupt, EOFError, SystemExit):
        break
```

Upon running the bot, with the valid checker code active, we see that most responses are valid, provided by the corpus, but even some responses that were provided by the corpus were unsuccessful with training. A valid response is provided to update the unknown or incorrect answer in its place.



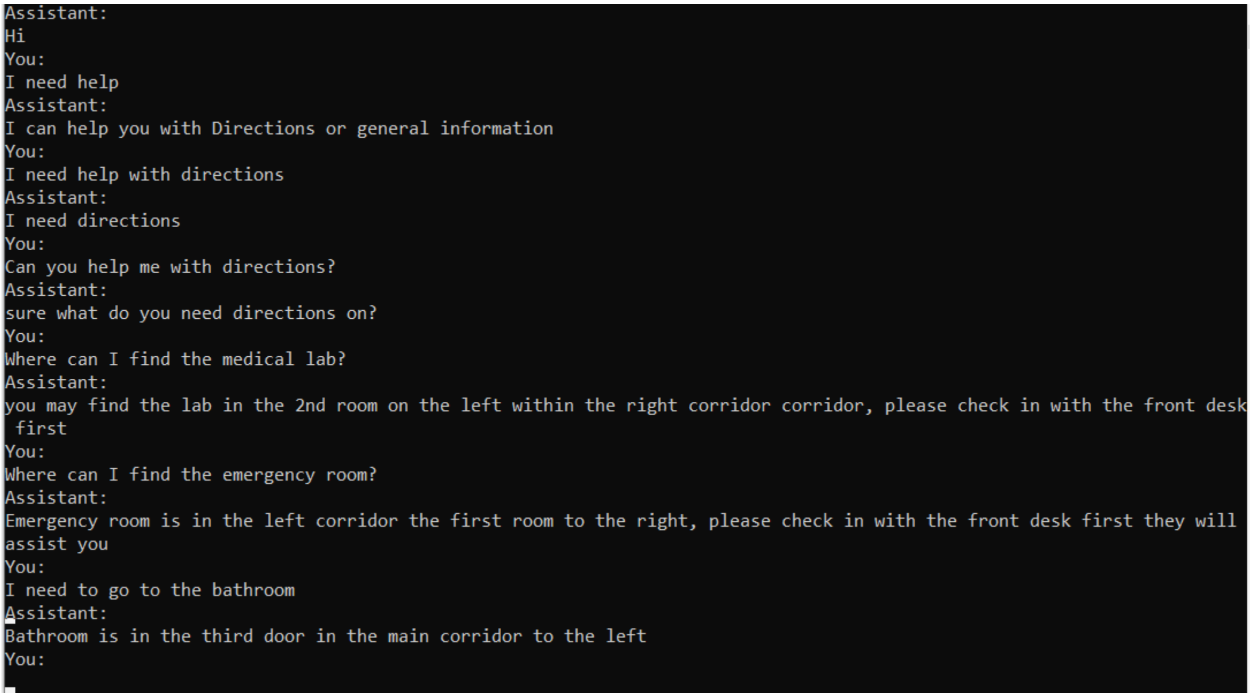
```
You:
Where can I find the medical lab
Assistant:
you may find the lab in the 2nd room on the left within the right corridor corri
dor, please check in with the front desk first

Is this a valid response?
yes
You:
Where is the bathroom
Assistant:
You may find the bathroom in third room down the main corridor to the left

Is this a valid response?
yes
You:
Where is the bathroom located?
Assistant:
We are located at 2275 Glade Avenue Hayward, CA 94545

Is this a valid response?
no
please input the correct response
You may find the bathroom in third room down the main corridor to the left
Responses added to bot!



You:
>>> |
```

The bot is then trained a without the validity checker, we can see that a good majority of the responses are successful but the chatbot will still continue to struggle with questions that have not been asked too often. below we see the bot without the validity checker and has been trained to a further extent through conversation and modifying the corpus.



```
Welcome to St.Rose Hospital main terminal, How may I help you?

You:
Where is the medical lab?
Assistant:
you may find the lab in the 2nd room on the left within the right corridor, please check in with the front desk first
You:
Where is the exit?
Assistant:
You may find the bathroom in third room down the main corridor to the left
You:
This is an emergency
Assistant:
Emergency room is in the left corridor the first room to the right, please check in with the front desk first they will
assist you
You:
I am in pain
Assistant:
Emergency room is in the left corridor the first room to the right, please check in with the front desk first they will
assist you
You:
bathroom
Assistant:
You may find the bathroom in third room down the main corridor to the left
You:
I need to see a patient
Assistant:
If you would like to visit a patient, please check in at the front desk, for more information
You:
```

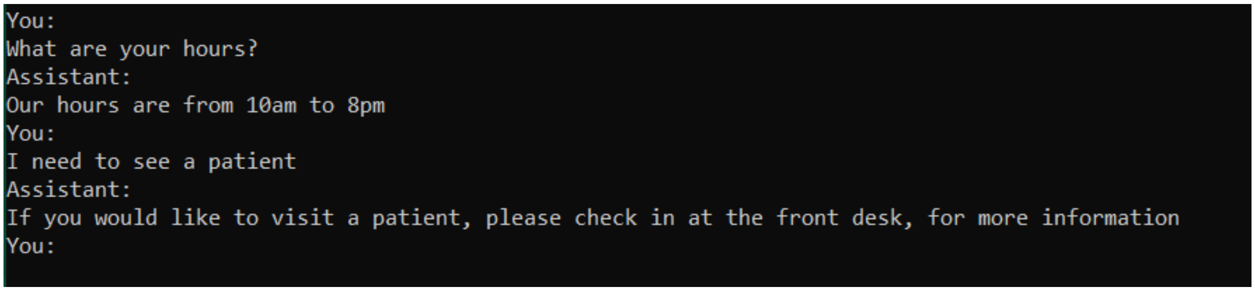
It is also noted in the above image that the bot's algorithm has now learned that the full sentence is not needed to know that the user wishes to go to the bathroom, so just inputting "bathroom" would give directions to the hospital bathroom.



```
Assistant:
Hi
You:
I need help
Assistant:
I can help you with Directions or general information
You:
I need help with directions
Assistant:
I need directions
You:
Can you help me with directions?
Assistant:
sure what do you need directions on?
You:
Where can I find the medical lab?
Assistant:
you may find the lab in the 2nd room on the left within the right corridor corridor, please check in with the front desk
 first
You:
Where can I find the emergency room?
Assistant:
Emergency room is in the left corridor the first room to the right, please check in with the front desk first they will
assist you
You:
I need to go to the bathroom
Assistant:
Bathroom is in the third door in the main corridor to the left
You:
```

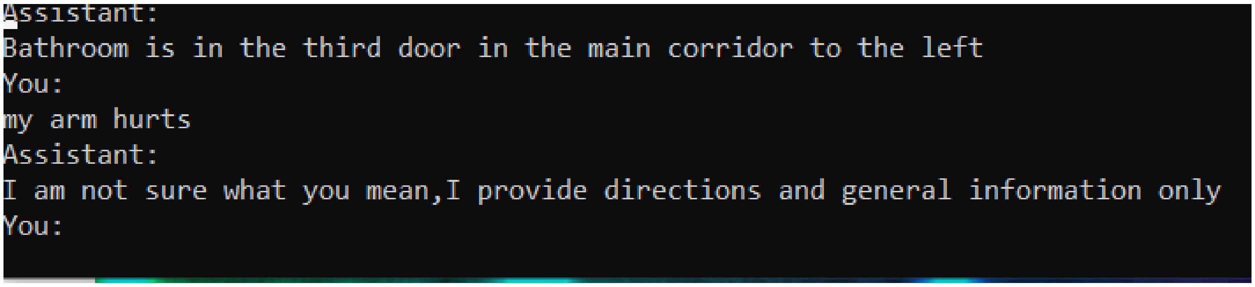
The bot in the above instance is able to give directions to rooms located in either the main, left, or right corridor such as needing to go to the emergency room, medical lab, or the bathroom. An error can be noted where asking for help with directions will give an odd response from the bot.

The bot will also give information on hours or respond if the user needs to see a patient by sending them to the front desk.



```
You:
What are your hours?
Assistant:
Our hours are from 10am to 8pm
You:
I need to see a patient
Assistant:
If you would like to visit a patient, please check in at the front desk, for more information
You:
```

A big limitation that has been seen when running this chatbot is mostly with questions it was not prepared for such as "my arm hurts", as well as the need for specific words to get the bot to respond with the correct answers, although It is shown that this isn't always the case with this bot.



```
Assistant:
Bathroom is in the third door in the main corridor to the left
You:
my arm hurts
Assistant:
I am not sure what you mean,I provide directions and general information only
You:
```

Conclusion

The bot created through python and the chatterbot library was able to respond to questions generally limited to this hospital scenario. Initially, the chatbot had many difficulties responding correctly to questions, but after further training by modifying the corpus as well as personally validating the input, the Chatbot was able to provide directions to questions to specific areas of the hospital, provide general information, and was also able to provide the appropriate directions should the user be in an emergency. The chatbot's responses were mainly limited to the responses they were trained on, but only to an extent, it was able to learn different variations of question/answer combinations provided by the corpus, but at some points, even answers provided by the corpus would yield a random response. The user experience can be improved by further modifying the corpus and repeatedly validating the inputs even further, I believe the user experience can also be improved by allowing access