The Sassbot 3000

CSC340 Artificial Intelligence

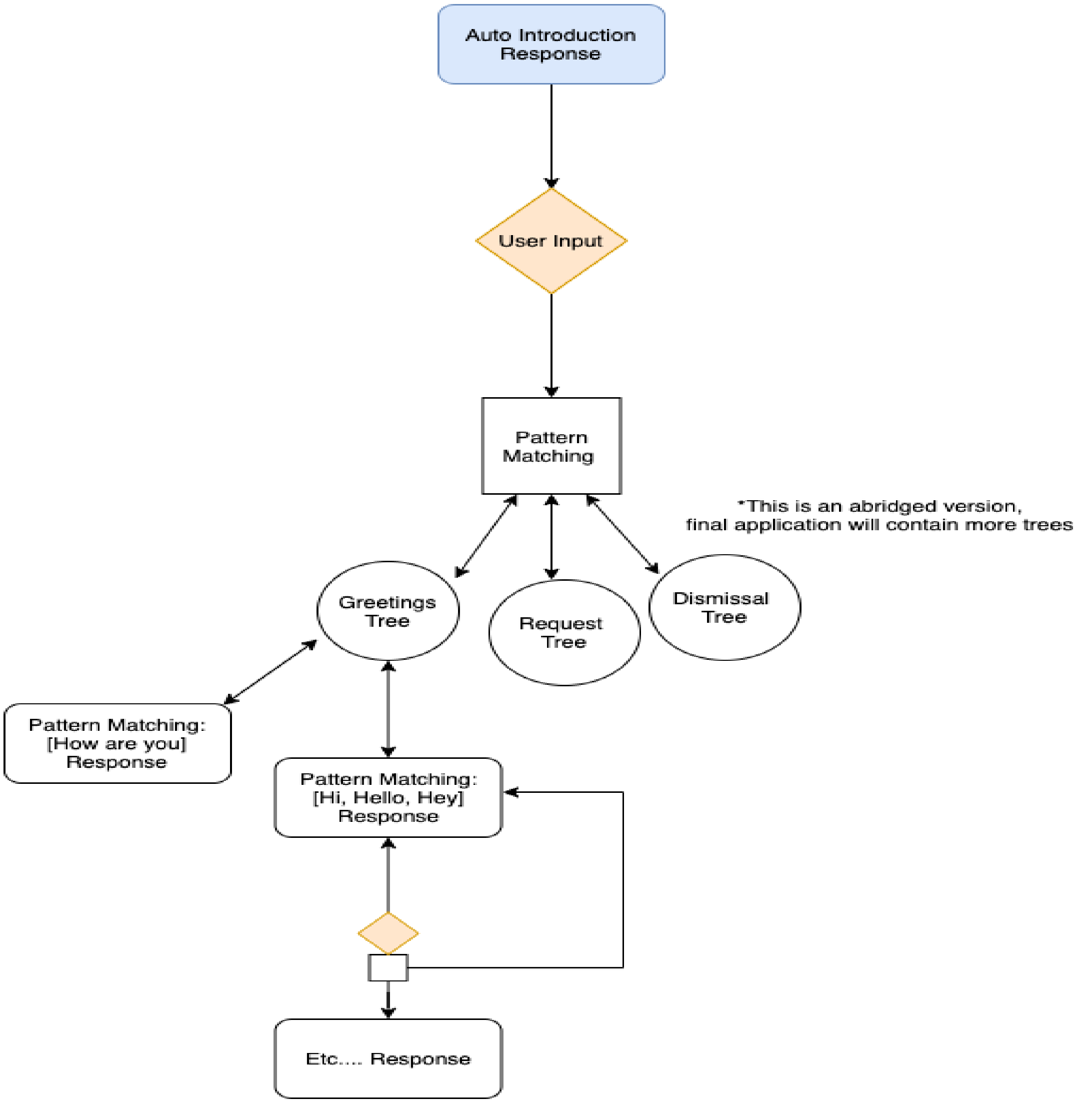
Alexander Rossetti & Damian Barrous
May 7, 2020

Project Description

Over the years AI and machine learning have been buzzwords that have been gaining a ton of traction. Alongside this traction buildup, "Intelligent" voice assistants and chatbots have found a place in almost everyone's smart devices. With highly sophisticated chatbots becoming more common by the day, it raises the question, would anyone want to engage with them in conversation with them on a regular basis? Even though these assistants and bots might seem intelligent, almost all of them are more of a nuisance to engage with in more than a brief interaction; they miss the best part of being human, sarcasm. We aimed to change that with the Sassbot 3000, a chatbot that aims to entertain, astonish, and wow it's users with its unique "sarcastic" personality. The chatbot interactions will feel like a comedy sketch and it's answers might not always (or ever) be straightforward or informational.

The language we used for implementation was Python (Interpreter : 3.7) and the algorithm was Breadth First Search to handle our quick responses and assign dialog trees. For the dialog trees however, we implemented the Depth First Search algorithm to travel along each tree or exit out of the tree if the response is not valid or incorrect. Only the libraries of re (Regular Expressions) and random were used in this project, everything else was written from scratch. All the data that was used was created from scratch by both of us. Whenever the user gives input to the chatbot, a breadth first search is performed in order to either assign a dialog tree or initiate a quick response. The pattern matching was done using regular expressions. If a user enters into a dialog tree then they keep traversing down the tree using depth first search, but if they enter an invalid response the traversing breaks and the user is back to the user input phase.

UML Diagram:


Auto Introduction Response
User Input
Pattern Matching
*This is an abridged version, final application will contain more trees
Greetings Tree
Request Tree
Dismissal Tree
Pattern Matching: [How are you] Response
Pattern Matching: [Hi, Hello, Hey] Response
Etc.... Response

Suggestions/Feedback from presentation

A GUI was recommended during our presentation, which we originally attempted to do with PyQT but because of lack of planning and time constraints, it would require a complete reworking of the script to make it work in that environment.

References

Python RE docs -
https://docs.python.org/3/library/re.html

Info on Rule Based Chatbots -
https://www.hubtype.com/blog/rule-based-chatbots-vs-ai-chatbots

Source Code

https://github.com/Alexzetti/Sassbot
Or
https://github.com/dbarrous/Sassbot