VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Natural language processing

Project report: Python based Chabot system for FFCS

M LALITH RAO 15MIS1022

SAI DEEPAK KUMAR 15MIS1123

Natural language processing

Project report: Python based Chabot system for FFCS

Abstract:

The project is based on building a Chabot system for VIT universities FAQ's. This concept of Chabot is used in various organisations for customer query satisfaction and helps people initiate conversation on a dynamic basis. The Chabot system is build using the chatterbot API and corpus that help it to reply to customer query.

With the help of corpus the catboat can train on them number of times to get specific reply for different query's in different categories like:

- Information on AI
- Computers
- Conversations
- Emotions
- Restraint service
- Food
- Greetings and salutations
- Business

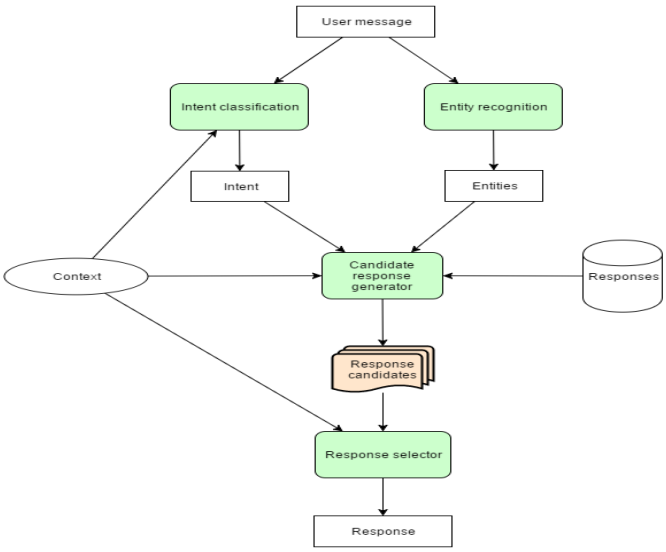
The corpus are stored as YML file from which query are answered are retrieved and replied.

Scope:

This project is based on answering query via the interface with the help of chatterbot API working in the background to supporting the UI.

The bot can be trained to answer on comments for reedit or twitter expressing human felling or can be asked a question as to infer knowledge.

The UI is based on a simple html script which is controlled by the python server to communicated for reply. The python server uses an instance for the UI to go live so that conversations can be started,



Application to run the Chabot:

1. **Application Programming Interface:**

Chatterbot API:

ChatterBot is a machine-learning based conversational dialog engine build in Python which makes it possible to generate responses based on collections of known conversations. The language independent design of ChatterBot allows it to be trained to speak any language.

Working:

An untrained instance of Chatterbot starts off with no knowledge of how to communicate. Each time a user enters a statement, the library saves the text that

they entered and the text that the statement was in response to. As Chatterbot receives more input the number of responses that it can reply and the accuracy of each response in relation to the input statement increase. The program selects the closest matching response by searching for the closest matching known statement that matches the input, it then returns the most likely response to that statement based on how frequently each response is issued by the people the bot communicates with.

simple-web socket-server

This is a python library for establishing TCP/IP connection between client and server that is the back end for the HTML interface of the system.

Using this we can establish a nondependent client server communication to the chat application.

2. **Python files:**

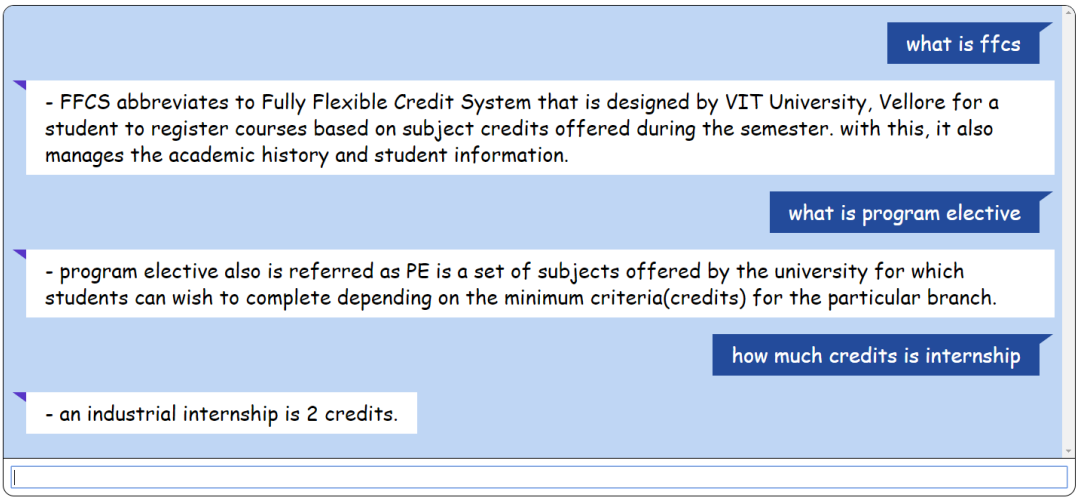
Namely there are three files for the application:

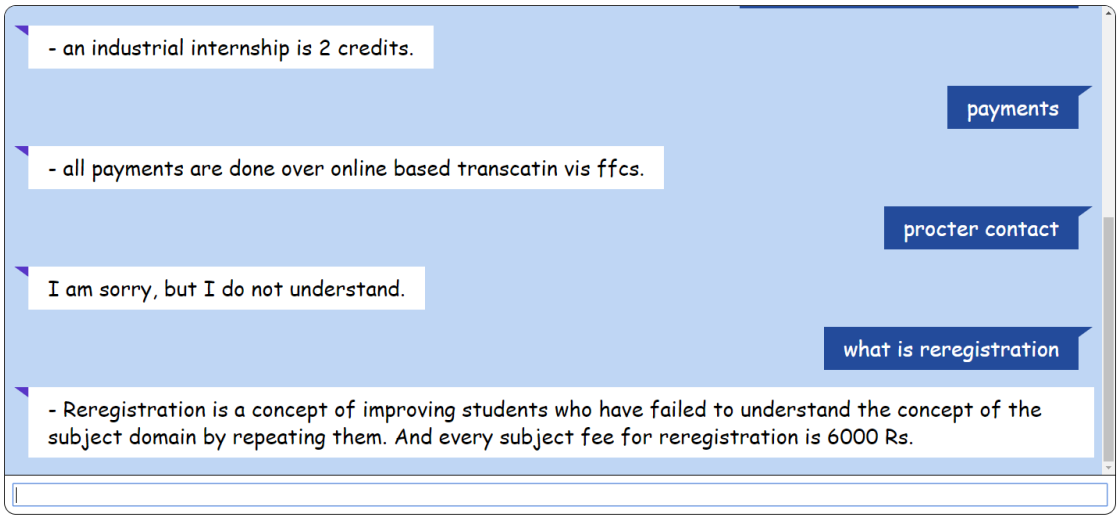
1. Chatbot_train.py that is used to train the Chabot with the YML files at the beginning of server start for all reply's to certain topics to question.
2. Chatbot.py is use as a internal service to reply from which the message interact to the UI.
3. And server.py is used for establishing a network GUI between the html file and Chabot system.

3 DB SQLite file :

This file stores the chats for previous question so that every time the servers go up it is not necessary to training the bot again and again but it can infer from the database.

Result





Other example:

conversations:

which is the first computer

- It's hard to say, but The ENIAC is regarded as the first 'real' computer. It was developed at University of Pennsylvania in 1946.