College Enquiry Chat Bot

(A computer program on web which converses with Humans)

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology.

in

Computer Science & Engineering

under the supervision of

Dr. Jagpreet Sidhu

By

Karanvir Singh Pathania 151248

To

Jaypee University of Information and Technology Waknaghat, Solan – 173234, Himachal Pradesh

Candidate’s Declaration

I hereby declare that the work presented in this report entitled **“College Enquiry Chatbot”** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an original record of my own work which has been carried out over a period from August 2018 May 2019 under the supervision of **Dr. Japreet Sidhu**(Assistant Professor Senior Grade,Computer Science & Engineering and Information Technology).

The subject matter included in this report is purely an original content.

Karanvir Singh Pathania(151248)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Jagpreet Sidhu

Assistant Professor (Senior Grade)

Computer Science & Engineering and Information Technology

Dated: 9-15-2019

Acknowledgment

Firstly, I would like to thank our supervisor Dr. Jagpreet Sidhu at the Department of Computer Science & Engineering and Information Technology at Jaypee University of Information Technology, who helped me with the project work. I would like to thank him for providing utmost guidance throughout the time of development of the project.

I have grown both personally and academically from this experience and I am also very grateful for having had the opportunity to conduct this study.

Dated: 09-05-2019 Karanvir Singh Pathania

Table of Contents

Abbreviations and Symbols

Abbreviations	**Full Forms**
API	Application Program Interface
NLP	Natural Language Processing
NLG	Natural Language Generator
NLU	Natural Language Understanding
AIML	Artificial Intelligence Markup Language
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
JSON	JavaScript Object Notation
UI	User Interface
GPU	Graphics Processing Unit
NLTK	Natural Language Tool Kit

List of Figures and Tables

Abstract

College Enquiry Chat bot is simple web application which aims to provide the information regarding college. The information can be in the form of teachers or student's GPA or the various activities in the college. It can be an upgraded form of our college's webkiosk. After some improvements and some additions this project can be fully embedded into the working site of the college.

The chatbot created here is a web based application which used Natural Language Processing Libraries and Artificial Intelligence Markup Language to have conversations with humans. "Eliza" and "Cleverbot" are some of the web applications which have been created in the past. Like "Eliza", the responses of this chatbot are programmed up to some extent. This is because of the fact that it is a simple bot which answers the queries regarding the college. Since the curriculum of the college keeps on changing, there has to be database which can be edited and upgraded from time to time.

So far, what I have achieved is a sample program which processes the response of the users by using simple parsing and substituting them into premade templates. It also uses hardcoded phrases so that the conversation is continued.

In the future, NLP can be implemented to understand what a user is saying and give the solutions to his problems. Natural Language Processing is a field of computer science sub- branch artificial intelligence which is concerned with interactions between computer and humans. Some the field inside NLP are Natural Language Understanding(NLU) and Natural Language Generation(NLG).

Introduction

College Enquiry ChatBot is a web application which uses artificial intelligence concepts to have conversations with humans. Some of the similar web applications built in the past are “Eliza”, “Cleverbot” etc.

This report will revolve around the concept of NLP and AIML along with the work committed to build Eliza. Further, we will also see the various problems and complications that arise while developing these applications and how these can be managed to make them better.

The sample application is developed using Python Kernel and XML’s Artificial Intelligence Markup Language(AIML) along with a database file which stores the name, e-mail, and password to tell the GPA of a student. It is accessed using MYSQL. The front end of the project is designed using HTML, CSS and Javascript.

The inspiration to build this project came from the working our college’s webkiosk. It is possible that the chatbot can be connected to the college database using the webkiosk’s API but it included the implementation of JSON.

The construction of the project is similar to Eliza. Since it was first of its kind and an open source, it provided an idea of how these programs work. It worked on an algorithm based on substitution. Another creation named Cleverbot was much efficient than Eliza, but since it is not an open source and its algorithm is also very complex, it is not much of an importance here. However, if its algorithm is studied, it would create an application which would be very complex and help to broaden the scope of ChatBot.

Literature Review

1.1 Components of ChatBot Application

The main components of the ChatBot are:

- UI
- Back-end

UI: The user interface is simple with not much colors. It is kept as simple as possible so as to make it look like a college chatbot. It consists of a text box at the bottom where the user may write the queries. A "send" button is placed in order to send the query to the bot. The UI is created using HTML, CSS and Javascript.

Background Working: There are three phases:

Parsing and Substitution

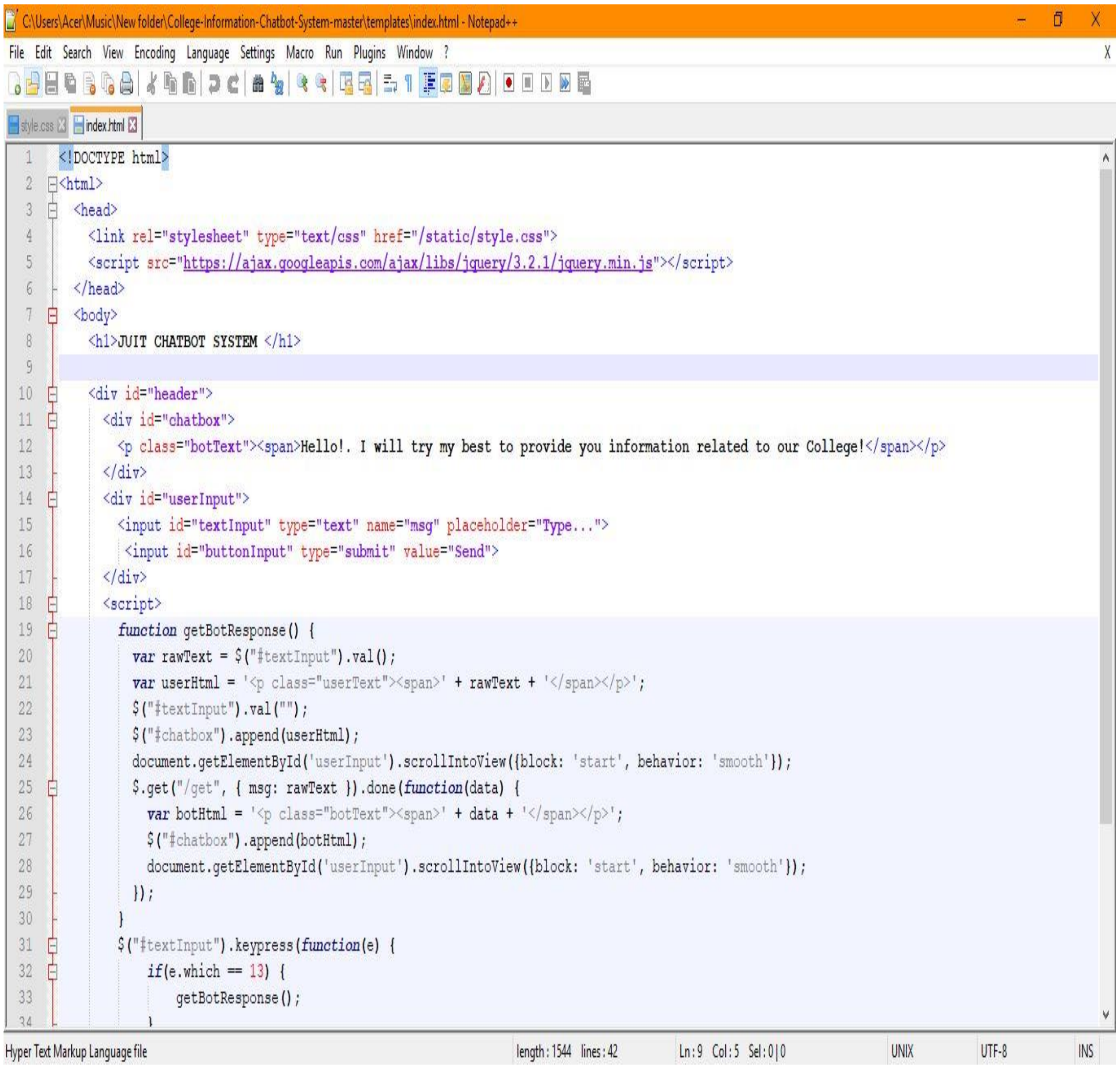
Natural Language Processing

Database

Parsing and Substitution: Whenever a user types a query, it is passes on to a class which parses the input and then substitutes words and phrases with other words and phrases so that a statement which is grammatically correct can be generated. It is carried out using XML and Python.

Natural Language Processing: NLP is required so that the data which is parsed can be "understood" by the application. For e.g. a user's humour, feelings, names and places mentioned in the input. NLP is not implemented in the project but can be implemented if needed in the future.

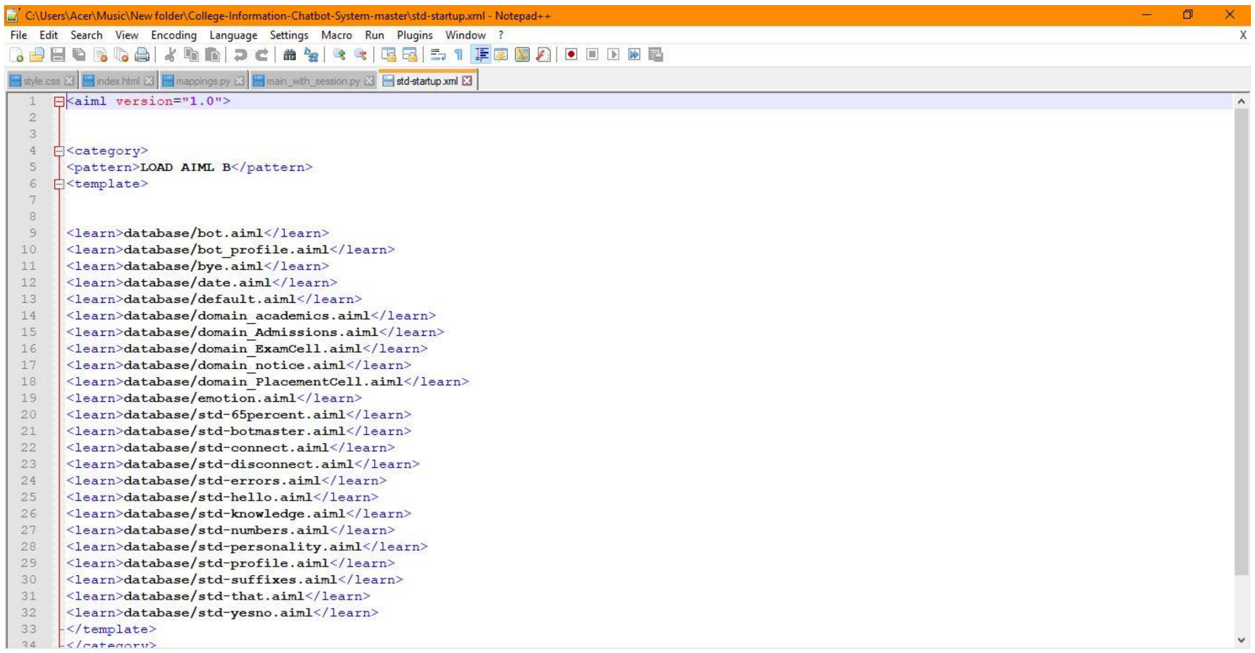
Database: There are various database files in the database folder with .aiml extension. These are the files which contain the various patterns of the conversations. Also a .db file is present which stores the name, e-mail and GPA of a student. It is stored offline because the college webkiosk's API was not available. Also it demanded the implementation of JSON.


C:\Users\Acer\Music\New folder\College-Information-Chatbot-System-master\templates\index.html - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
style.css
index.html
Hyper Text Markup Language file
length : 1544 lines : 42
Ln : 9 Col : 5 Sel : 0 | 0
UNIX
UTF-8
INS

```html
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="/static/style.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
  </head>
  <body>
    <h1>JUIT CHATBOT SYSTEM </h1>

    <div id="header">
      <div id="chatbox">
        <p class="botText"><span>Hello!. I will try my best to provide you information related to our College!</span></p>
      </div>
      <div id="userInput">
        <input id="textInput" type="text" name="msg" placeholder="Type...">
         <input id="buttonInput" type="submit" value="Send">
      </div>
      <script>
        function getBotResponse() {
          var rawText = $("#textInput").val();
          var userHtml = '<p class="userText"><span>' + rawText + '</span></p>';
          $("#textInput").val("");
          $("#chatbox").append(userHtml);
          document.getElementById('userInput').scrollIntoView({block: 'start', behavior: 'smooth'});
          $.get("/get", { msg: rawText }).done(function(data) {
            var botHtml = '<p class="botText"><span>' + data + '</span></p>';
            $("#chatbox").append(botHtml);
            document.getElementById('userInput').scrollIntoView({block: 'start', behavior: 'smooth'});
          });
        }
        $("#textInput").keypress(function(e) {
            if(e.which == 13) {
                getBotResponse();
```

Fig 1(a): The HTML file index.html

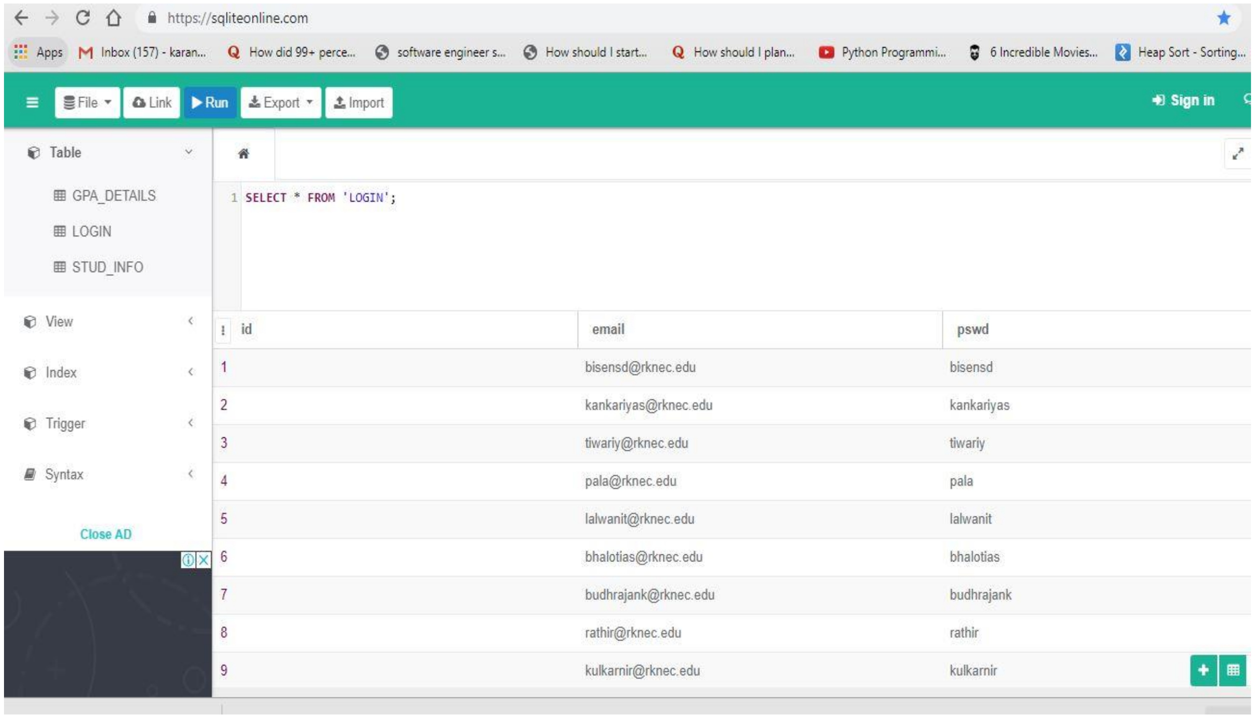

C:\Users\Acer\Music\New folder\College-Information-Chatbot-System-master\std-startup.xml - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
style.css
index.html
mappings.py
main_with_session.py
std-startup.xml

```xml
<aiml version="1.0">


<category>
<pattern>LOAD AIML B</pattern>
<template>


<learn>database/bot.aiml</learn>
<learn>database/bot_profile.aiml</learn>
<learn>database/bye.aiml</learn>
<learn>database/date.aiml</learn>
<learn>database/default.aiml</learn>
<learn>database/domain_academics.aiml</learn>
<learn>database/domain_Admissions.aiml</learn>
<learn>database/domain_ExamCell.aiml</learn>
<learn>database/domain_notice.aiml</learn>
<learn>database/domain_PlacementCell.aiml</learn>
<learn>database/emotion.aiml</learn>
<learn>database/std-65percent.aiml</learn>
<learn>database/std-botmaster.aiml</learn>
<learn>database/std-connect.aiml</learn>
<learn>database/std-disconnect.aiml</learn>
<learn>database/std-errors.aiml</learn>
<learn>database/std-hello.aiml</learn>
<learn>database/std-knowledge.aiml</learn>
<learn>database/std-numbers.aiml</learn>
<learn>database/std-personality.aiml</learn>
<learn>database/std-profile.aiml</learn>
<learn>database/std-suffixes.aiml</learn>
<learn>database/std-that.aiml</learn>
<learn>database/std-yesno.aiml</learn>
</template>
</category>
```

Fig 1(b): Startup file std-startup.xml



```sql
SELECT * FROM 'LOGIN';
```

id	email	pswd
1	bisensd@rknec.edu	bisensd
2	kankariyas@rknec.edu	kankariyas
3	tiwariy@rknec.edu	tiwariy
4	pala@rknec.edu	pala
5	lalwanit@rknec.edu	lalwanit
6	bhalotias@rknec.edu	bhalotias
7	budhrajank@rknec.edu	budhrajank
8	rathir@rknec.edu	rathir
9	kulkarnir@rknec.edu	kulkarnir

Fig 1(c): Database file contents opened in online browser

1.2 Eliza and the Doctor Script

Many programs and applications have been developed based on this concept. One of the first was Eliza. It is famous because it is an open source program.

ELIZA makes the use of primitive natural language processing. It operates on user's responses to scripts, in which the most famous was the DOCTOR, a simulation of psychotherapist. It provided a basic human-like interaction with almost no information about human thought or emotion. It was written at MIT between 1964 and 1966.

A simple conversation example:

User: My head hurts.

DOCTOR: Why do you say your head hurts?

User: My mother hates me.

DOCTOR: Who else in the family hates you?

ELIZA uses simple pattern matching techniques, but was in talks when the creator explained how it actually works. It was one of the first chatterbots in existence. It comprised of many scripts, each of which processed the responses of the users by simple parsing and by the substitution of them into premade templates and also there were hardcore phrases to as to continue conversations.

The creator of ELIZA never expected to create a fully working model consisting of NLP, it was rather a simple toy. What DOCTOR actually did was that it used a lot of data from the statements from the humans in order to compose the responses by using simple templates. It actually used an if-else pattern of using a certain template for a certain trigger. It was first implemented on SLIP language (an extension to FORTRAN).

ELIZA was important in revolutionizing the computing industry because such an idea of interaction was not a popular fancy.

A simple conversation transcript of ELIZA would look like this:

I wish to know about you.

Why do you say i might want to know me?

To write my report

What makes you believe to write your report?

I do not understand your question, please clarify

Earlier you said i would like to know me?

Yes

Can you elaborate on that?

I wish to know ELIZA

How does it feel to want?

Want what?

Maybe your plans have something to do with this.

Er, what plans?

Is it because of your life that you say

I would appreciate it if you would continue.

I will.

As it concluded here that ELIZA has a habit of parsing the responses of the user to frame the next question. Also after the conversation is continued these responses become predictable. Same is the case with the ChatBot created here. Since the data of templates is not much, the responses after some time become predictable.

1.3 Cleverbot

Cleverbot is also a web application which converses with humans and uses artificial intelligence algorithms. It was created be British scientist Rollo Carpenter. The difference between Eliza and Cleverbot is that its unique algorithm learns from humans and remembers words within its AI.

There is also a point to note that the bot's responses are programme. It learns from human input; The UI of my project is somewhat similar to Cleverbot where the human types into the box below the logo of Cleverbot and finds all the matching keywords or phrases matching the input. It responds to the input of the human by searching its saved conversations. It also responds to the particular input by finding how a human responded to that input when it was asked, but thtat happens in part or in full.

Because of its complex algorithm structure, it is constantly learning. It's data size is also increasing. Due to this it appears to display a degree of "intelligence". It's software updates are constantly checked and in 2014 it was upgraded to use GPU serving techniques.

Cleverbot also participated in a Turing Test in 2011 organised by IIT Guwahati. Cleverbot was judged to be 59.3% human. The software which participated in the test had to process 1 or 2 simultaneous requests, but it was noted that Cleverbot handled 80000 people at once.

1.4 Problems with Cleverbot

Despite the complex algorithm of CleverBot some problems were noted:

It doesn’t take anyone too many sentences for it to fail a Turing test, and figure out it was saying things that had been said to it many times.

It can't reference back than a single sentence.

It has no core identity.

It often answers a "why" question with a "where" answer.

The reason for this is that the CleverBot stores the responses that people give to it in return for that it says. It is also noted that it simply says the answers to someone else and records their answer.

The condition for a machine to hold a proper conversation with anyone is that it would need to design a system that "understands" what its hearing and saying properly so that it can hold a co-operative conversation. Making that happen is hard. The greatest obstacle to it is unraveling the jumbled up mess of complexity that words and ideas are made up of.

1.5 AIML

In Python, using AIML package, artificial intelligence ChatBots are easy to write. Its stands for Artificial Intelligence Markup Language. Basically it is just simple XML.

It was developed by Richard Wallace. He made a bot named ALICE. AIML is a form of XML that helps to define rules for patter matching and determining responses.

There are some phases to implement AIML which are used in this project:

# Create a standard Startup File

# Creating an AIML File

# Installing Python AIML Module

# Creating a Python Program

Some of the important tags used in AIML documents are:

# <aiml> - It defines the beginning and end of the AIML document.

# <category> -It defines the basic knowledge in Chatbot's knowledge base.

# <pattern> - It defines the pattern to match the user's input to the ChatBot.

# <template> - defines the response of the ChatBot to user's input

One of the very important questions that arises during the implemtation of AIML is that whether it counts as Artificial Intelligence or not.

A simple answer to this is that it more like an impression of intelligence rather than actual intelligence. It involves basic scripting which is done in XML and there are no learning models.

1.6 NLP

NLP is a part of artificial intelligence which deals with human languages. It has the following structure:

# Application

# NLP layer

# Knowledge Base

# Data Storage

NLP is divided into two very important components:

# Natural Language Understanding: It is mostly used to map inputs to useful representations. It is also helpful in analyzing different aspects of the language.

# Natural Language Generation: It is generally used text planning, sentence planning, and text realization.

NLP is implemented using a library in Python named NLTK.

There are some steps followed in NLP:

# Tokenization: It is the process to break a complex sentence into words. Also, the importance of each word is understood with respect to the sentence. It also helps to produce a structural description on an input sentence.

# Stemming: It is the process in which words are normalized into its base form or root form.

# Lemmatization: It is the process in which grouping of different inflected forms of a word is done. It also roots several words into one common root but the output of Lemmatization is a proper word.

# Stop Words: These are some of the words which are helpful to make a sentence meaningful but do not help in NLP.

# Parts of Speech: It is an inbuilt library containing the various parts of speech. For e.g.:

# CD- Cardinal Number

# NN- Noun Singular

# NNS- Noun Plural; etc.

# Named Entity Recognition: It helps to identify the particular entity name. For e.g. movie, monetary value, organization, location or quantities.

# Chunking: It is a process of picking up individual pieces of information and grouping them into bigger pieces.

Since NLP is a little hard topic to grasp, a smaller demo is implemented using the above processes. I have implemented NLP on Anaconda and the above processes have been implemented using various built-in functions.

However, all the above steps which make NLP complete are not implemented in this project structure. It might be a future task.

System Methodology

2.1 The general problem

# Making the program understand English language:

Making a computer understand the human language is the main problem with ChatBot. This is why as a conclusion it is too difficult to implement. This does not make it impossible. It can be done up to some extent. A solution to the problem is distinguishing the type of sentence.

Distinguishing the type of sentence (the user input) is a basic solution to this problem.

Here are four parts:

# a command (asking the bot to do something)

# a question (asking the bot to search for an answer but not performing)

# a statement (telling the bot to learn from the input)

# a simple response (like say hello, how are you doing)

Since it a very lengthy process, in order to minimize the scope, we will be using only 2 parts:

A Statement.

A Question.

Following is a representation of the division of the statement or a question entered by the user.

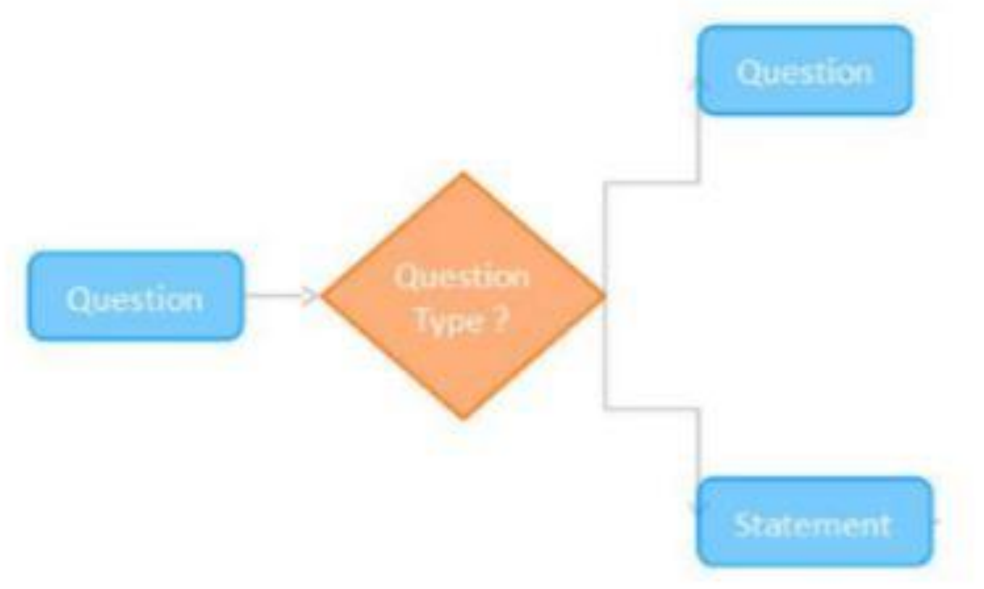


Fig 2(a): Query type

This was the main idea that went into the making of the ChatBot. A method od pattern recognition is used in the ChatBot.

The other problem that I faced is that if a user replies “karan’s panel name has Dr. Pradeep Kumar and Dr. Ekta Gandhotra and two others from HPU”. Here, the problem is that how will the program understand that karan is not an object. The solution was to find a keyword that represents humans from the input.

Some of the possibilities that arise from a single statement are:

“My friend Aditya has a PlayStation.”

Friend	Who is my friend?
Aditya	Who is Raj?/What does Aditya have?
PlayStation	Who has a PlayStation?

It is problem which goes out of the scope. Also we are trying to limit the use cases. Hence I have not covered the problem.

2.2 Understanding our scope of work

The main objective of our application “College Information ChatBot” is to provide information to the user. In order to do this, it has to be known who the user is, i.e. their name. Also there is a function inside the chatbot where the user can just enter the e-mail, password and get the GPA.

It is understood that this project is providing information only regarding the college, we need to limit the models made to understand the problem.

When this assumption is made, only one model comes to mind:

# Name-Relation-Relationship

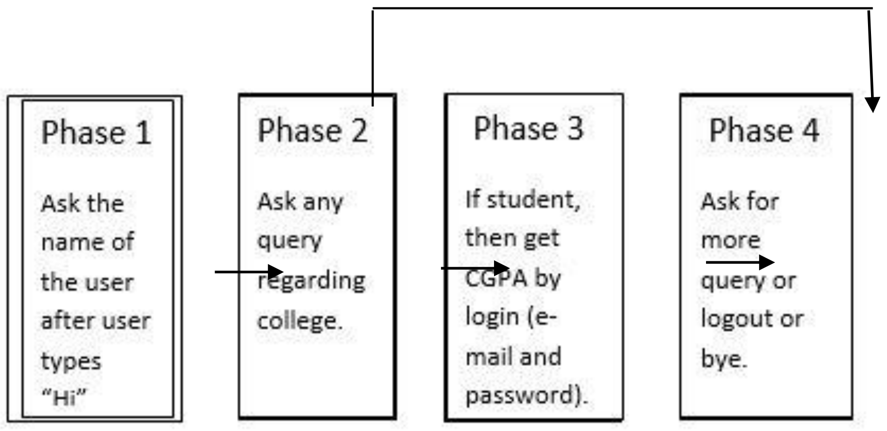
In this model the Name represents the name of the person which will be asked instantly after the user types “Hi”. Relation is the basic relation with the person, i.e. if he is a student or a faculty or anybody else.

One thing to note here is that it is an open application. Anyone can access information regarding college.

Anything which lies outside this scope will not be entertained and the user is required to keep the conversation within the scope of this application.

2.3 The Phases

The work flow of the application is roughly divided into 4 phases. It is also kept in mind the Chatbot is purely a college property so it does not include simple chatting. For e.g. answering the user's humor. The ultimate goal of the application is to provide information to the user. As a result, the bot will be formal. Due to this, the conversation is divided into 4 parts.



Phase 1
Ask the name of the user after user types "Hi"
Phase 2
Ask any query regarding college.
Phase 3
If student, then get CGPA by login (e-mail and password).
Phase 4
Ask for more query or logout or bye.

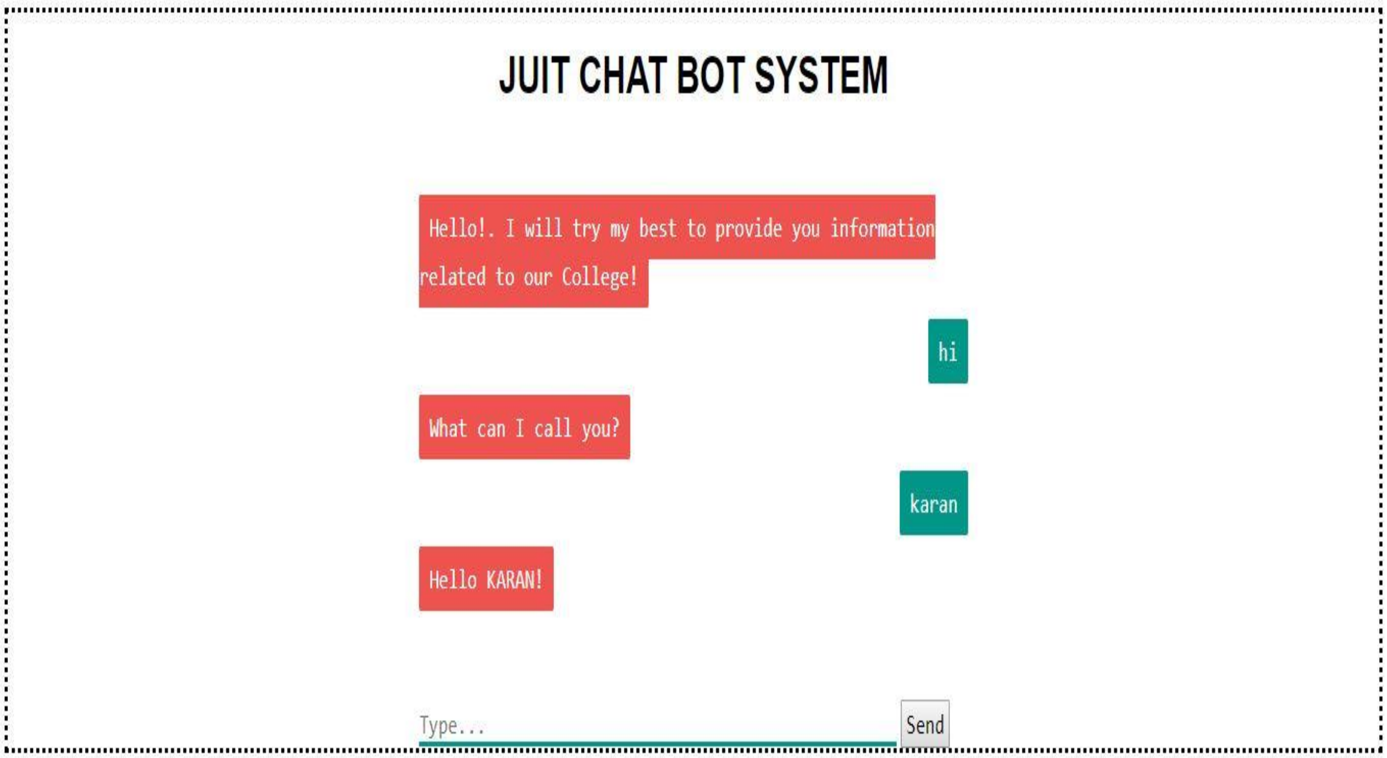
Fig 2(b): Different Phases

Phase 1:

This is the beginning phase in which the conversation starts. The bot will start answering after the user types something. Also, when the user opens the application the initial message displayed is

"Hello!. I will try my best to provide you information related to our College!"

When the user types "hi", the bot asks for the name of the user. For e.g. KARAN When typed, the reply is "Hello KARAN!".


JUIT CHAT BOT SYSTEM

Hello!. I will try my best to provide you information related to our College!

hi

What can I call you?

karan

Hello KARAN!

Fig 2(c): Chatbot Starting Context

Here the name is stored. The POS tagger of NLP is used to tag each word into different parts of the speech.

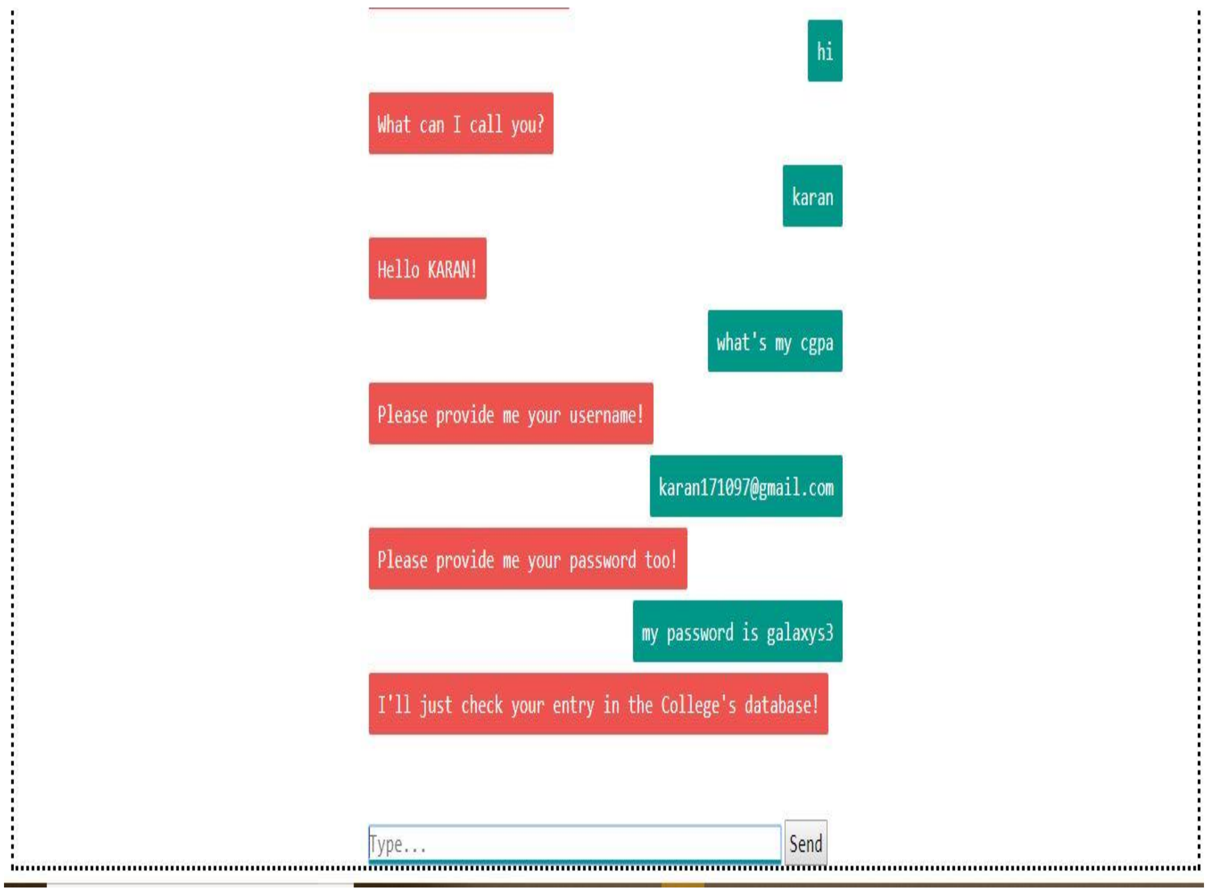
Phase 2:

This phase can be initiated in two ways. If the user is a simple simple person who wants the queries regarding the college irrespective of the affiliation with college or not, then the user can get the answer directly and can proceed to phase 4.

If the user is a student of the college and wants to know his/her CGPA, then the user proceeds to phase 3.

Phase 3:

If a particular student wants to know the CGPA then the bot asks for the username and password.



hi

What can I call you?

karan

Hello KARAN!

what's my cgpa

Please provide me your username!

karan171097@gmail.com

Please provide me your password too!

my password is galaxys3

I'll just check your entry in the College's database!

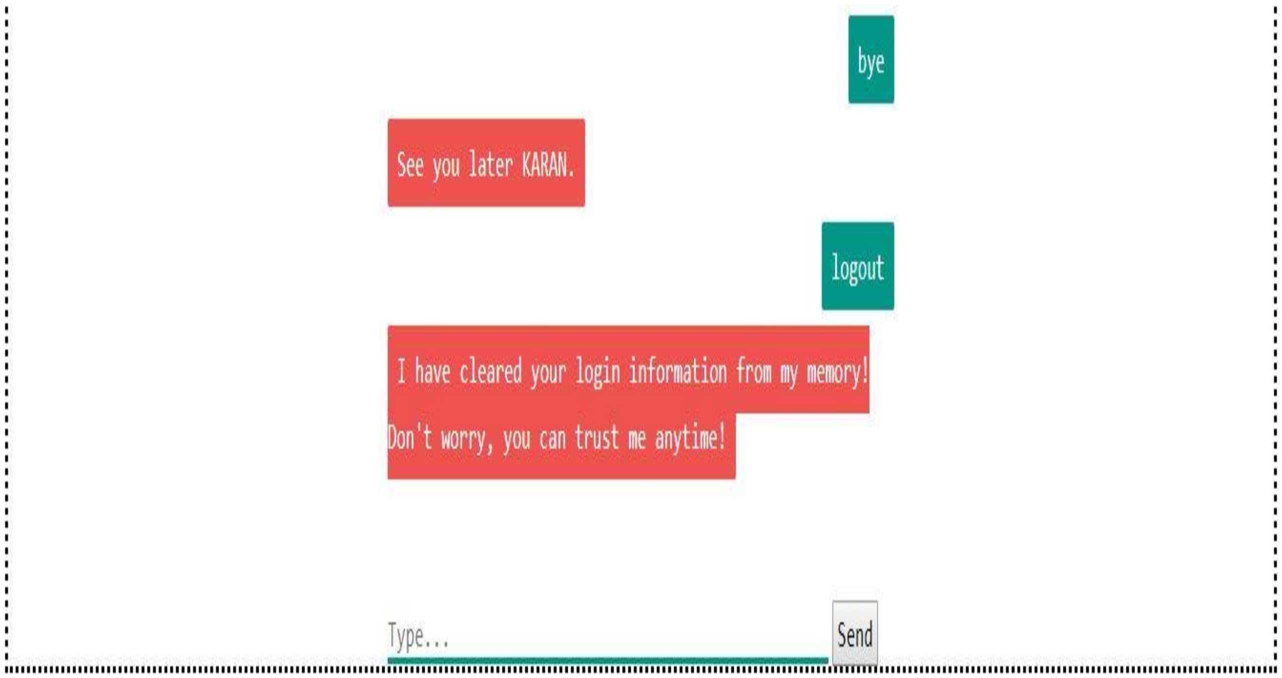
Fig 2(d): Bot answering the query

Here the bot matches the entered username and password which is present in the database file. There, various tables are made which contain information regarding the student.

Phase 4:

This is the last part of the conversation. Here the user gets the response from the user and after clearing the query the user can end the conversation by adding a statement “bye”. In case for a student, he/she can enter “logout” and the bot replies

“I’ve cleared your login information from my memory. Don’t worry, you can trust me anytime”



bye

See you later KARAN.

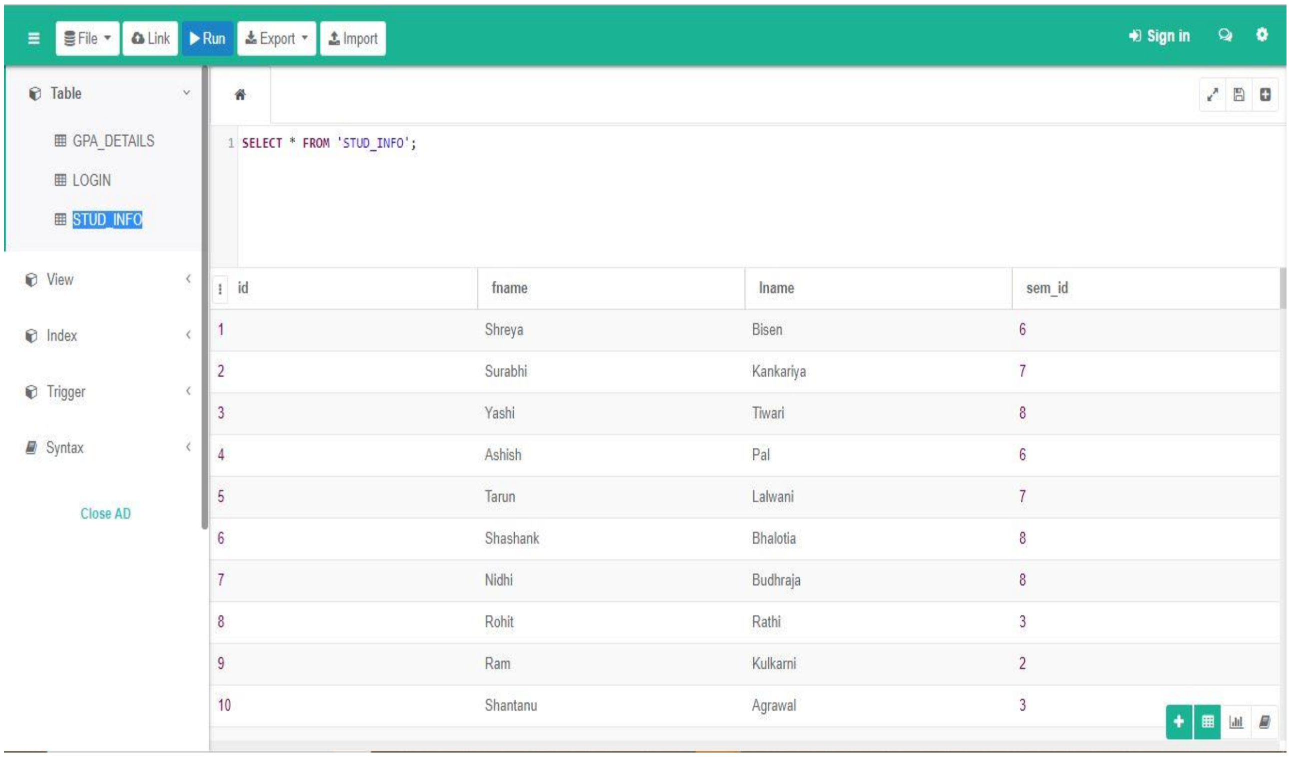
logout

I have cleared your login information from my memory!
Don't worry, you can trust me anytime!

Fig 2(e): Bot replying to user's end of conversation

2.4 The Storage

The data in this project is stored in two different formats. First is the files that have the extension of .aiml which contain the pattern of various questions and their supposed answers. The second file is the .db file which contains the pre-made table of the students, their cgpa, their name , passwords, sem id's etc.



```
SELECT * FROM 'STUD_INFO';
```

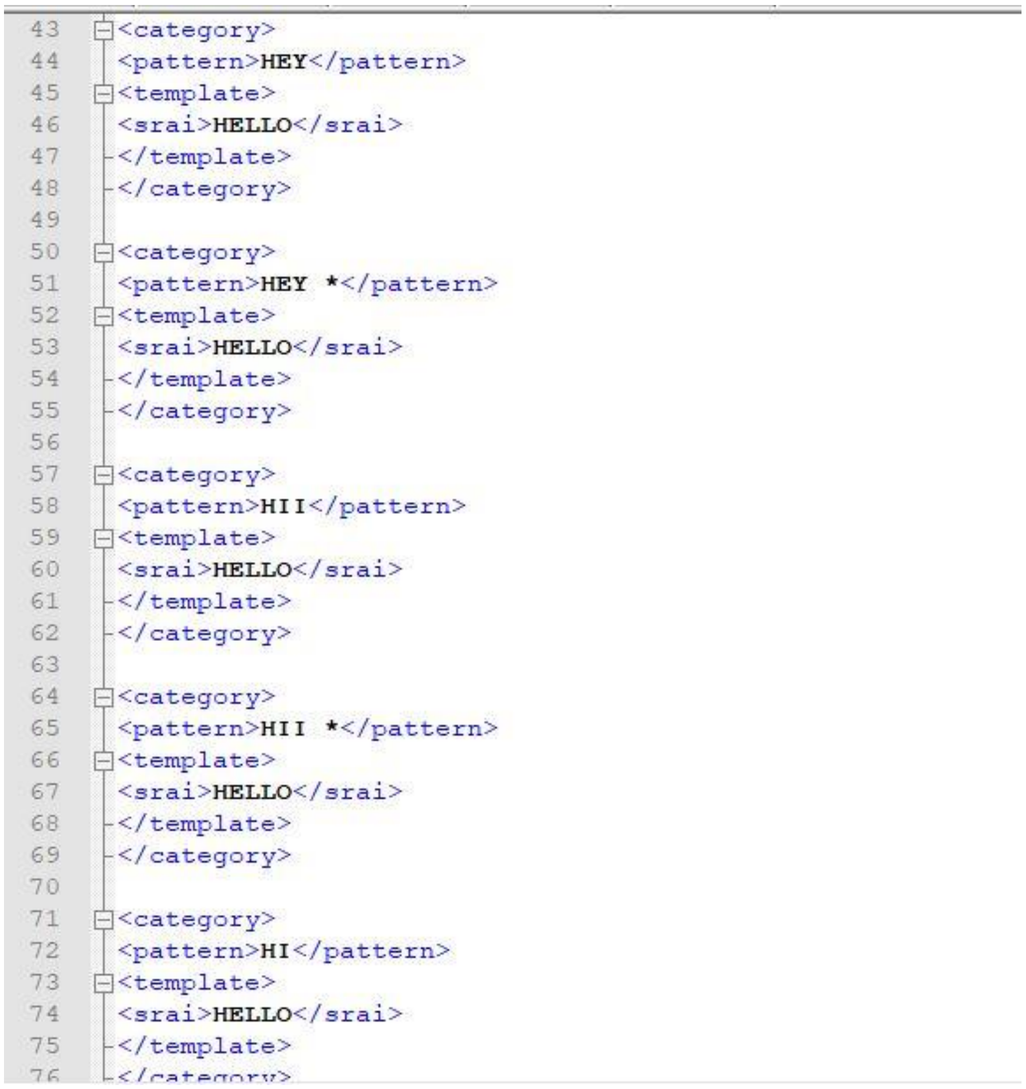
id	fname	lname	sem_id
1	Shreya	Bisen	6
2	Surabhi	Kankariya	7
3	Yashi	Tiwari	8
4	Ashish	Pal	6
5	Tarun	Lalwani	7
6	Shashank	Bhalotia	8
7	Nidhi	Budhraja	8
8	Rohit	Rathi	3
9	Ram	Kulkarni	2
10	Shantanu	Agrawal	3

Fig 2(f): Database containing various tables and values

Here the developer can edit the database.

There is also a "temporary memory". It stores the name of the user and the information related to the current conversation.

Also, a “sentence generation database” is also present which contains premeditated text and a substitute text. A premeditated is the one which conatins the question keywords. For e.g. “hi”, the answer can be “Hello there” or Hello again”.



```
<category>
<pattern>HEY</pattern>
<template>
<srai>HELLO</srai>
</template>
</category>

<category>
<pattern>HEY *</pattern>
<template>
<srai>HELLO</srai>
</template>
</category>

<category>
<pattern>HII</pattern>
<template>
<srai>HELLO</srai>
</template>
</category>

<category>
<pattern>HII *</pattern>
<template>
<srai>HELLO</srai>
</template>
</category>

<category>
<pattern>HI</pattern>
<template>
<srai>HELLO</srai>
</template>
</category>
```

Fig 2(g): The XML file containing templates and patterns

Here we see many alternate versions of replies of hello. The user can type hello in various ways.

3 Proposed System

The project consists of 5 basic parts:

A. Context Identification:

Here pre-processing has been applied to the input text so as to standardize the input. This is done keeping the system's requirements in mind. The recognition of the appropriate context is based on the keywords used in the text.

B. Personal Query Response System:

When the bot receives queries regarding CGPA, attendance etc. the validity of the user is checked by making them enter their e-mail and password. If found wrong, an appropriate answer is sent.

If the user is valid, the input text is processed so that the keywords can be extracted. Based on these keywords, the information is processed and provided to the user.

C. AIML Response System:

Artificial Intelligence Markup Language is used to map the input to an appropriate pattern. If the response is available inside the files, then it is sent to the user. If something does not match the database, then a random response is sent saying" Invalid Input".

D. Query Analysis and Response System:

This application provides information regarding the proceedings of the college.

It is known that the answer will be given to the user if the input matches a pattern in the AIML files. In a case where the input does not match with the data in AIML files, keywords are fetched.

An algorithm to check the sentence similarity(NLP) is applied to the modified input so that its similarity can be checked. If a sentence is retrieved with confidence>0.5, the query's answer is given.

If no questions map to the input, that particular data is stored in the log file for further improvement by the admin. Later that query can be included into the various AIML files. The bot, in this case gives the answer as “Answer not available”.

E. Context Reset

After the queries of the user are over and the he/she wants to exit the portal, then they have to just enter “logout” or “bye” or simple exit the portal.

When the user exits the application, all the input parameters are reset.

4 Design

Use case Diagram:

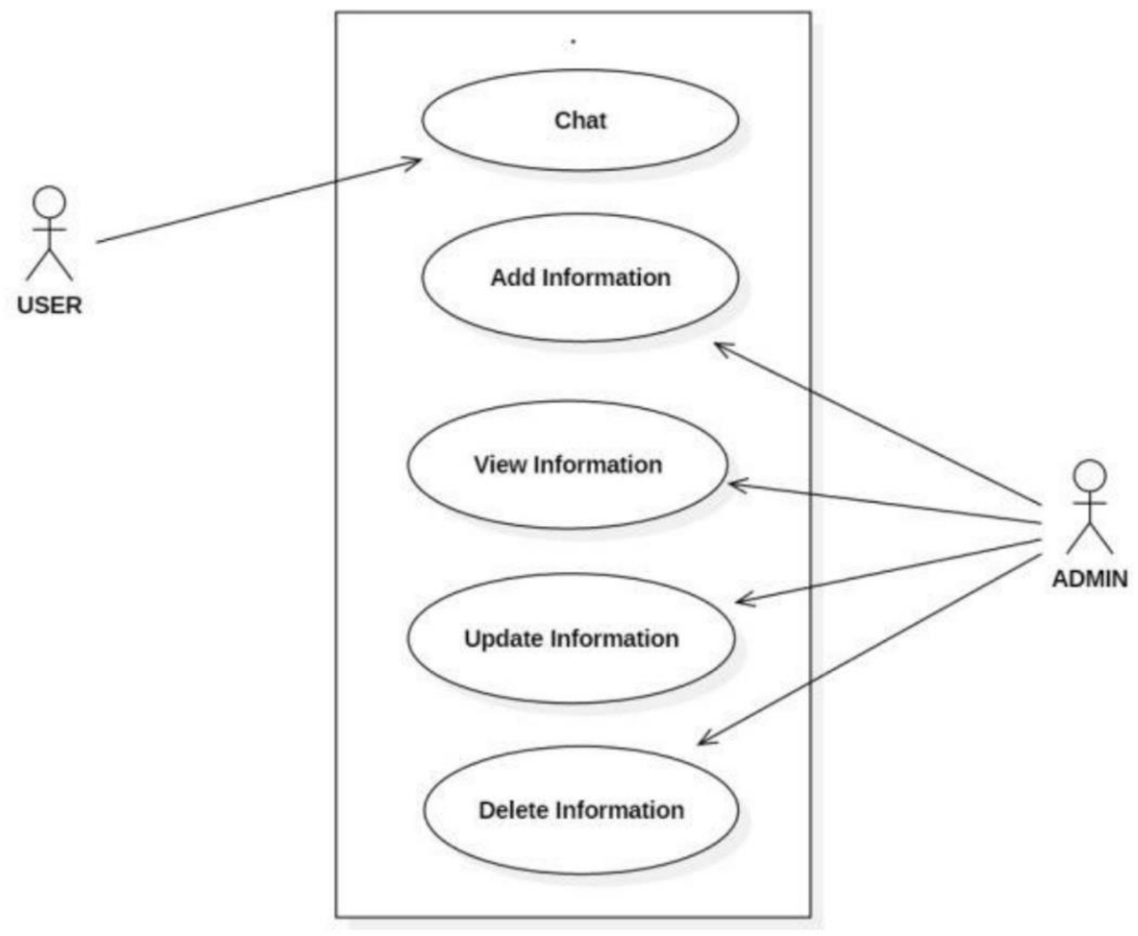


Fig 4(a): Use case diagram

Data Flow Diagrams:

A. Level 0:

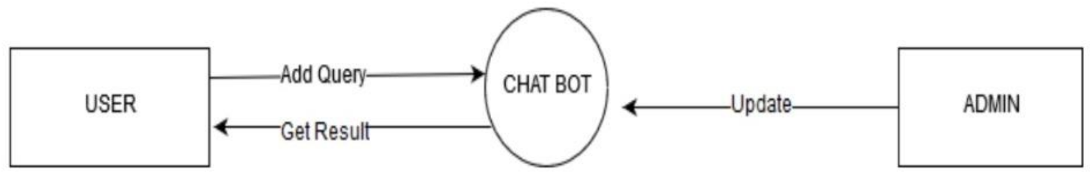


Fig 4(b): Level 0 DFD

B. Level 1:

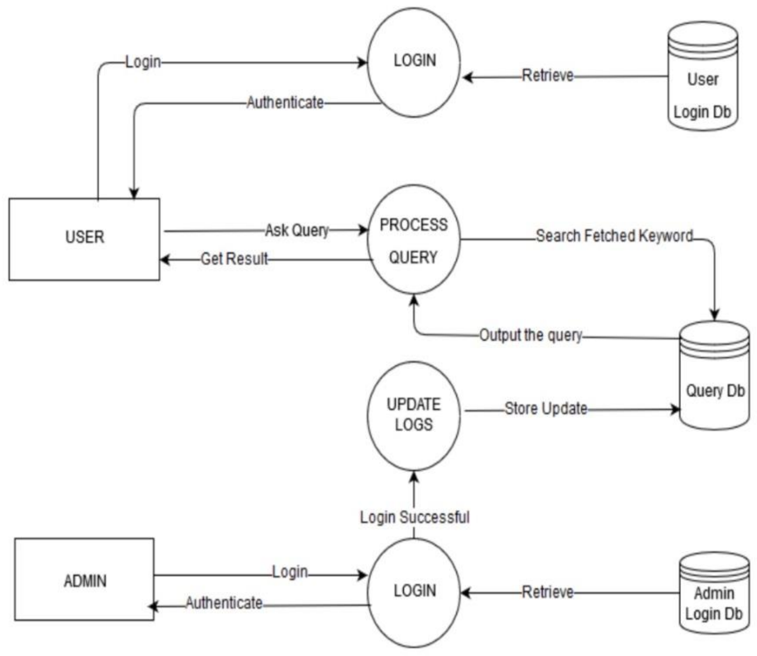


Fig 4(c): Level 1 DFD

Use case Diagram (Context Identification)

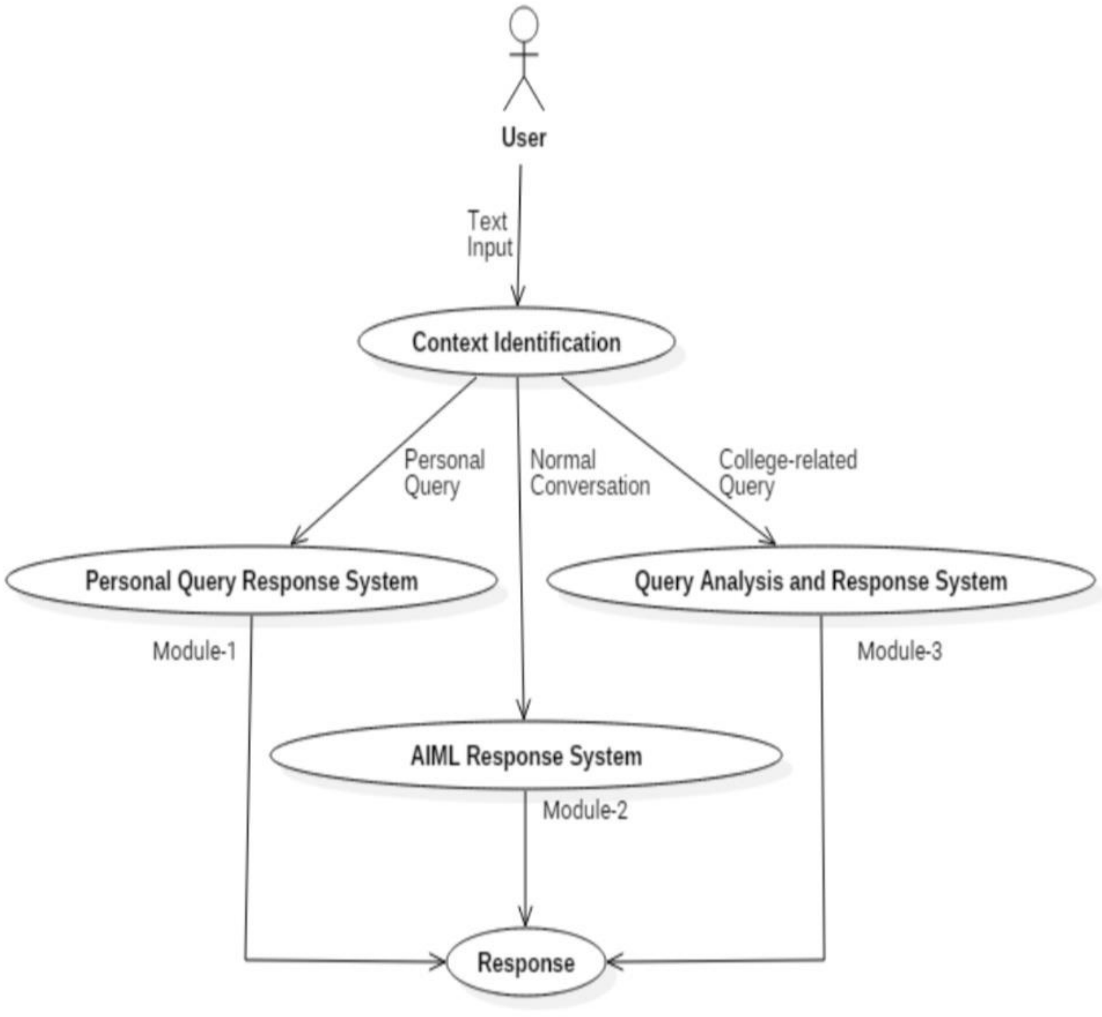


Fig 4(d): Use case Diagram(Context Identification)

Activity Diagram: Personal Query Response Activity:

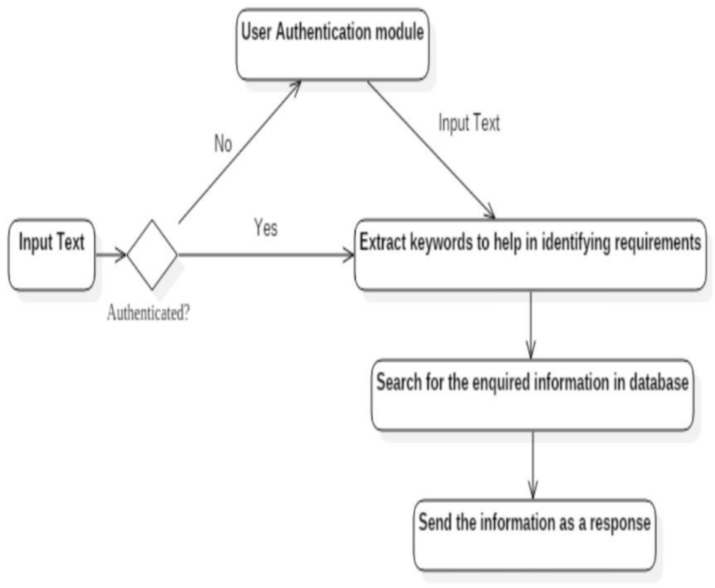

User Authentication module
No
Input Text
Input Text
Yes
Extract keywords to help in identifying requirements
Authenticated?
Search for the enquired information in database
Send the information as a response

Fig 4(e): Activity Diagram: Personal Query Response Activity

Normal Conversation Response Activity:

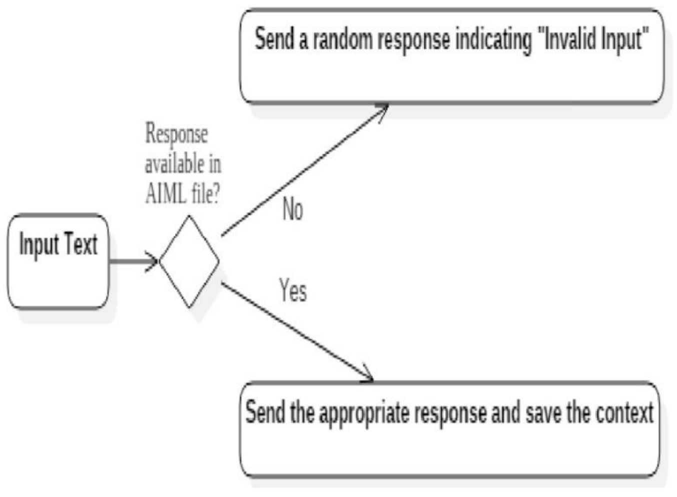

Send a random response indicating "Invalid Input"
Response available in AIML file?
No
Input Text
Yes
Send the appropriate response and save the context

Fig 4(f): Normal Conversation Response Activity

College Related Response Activity:

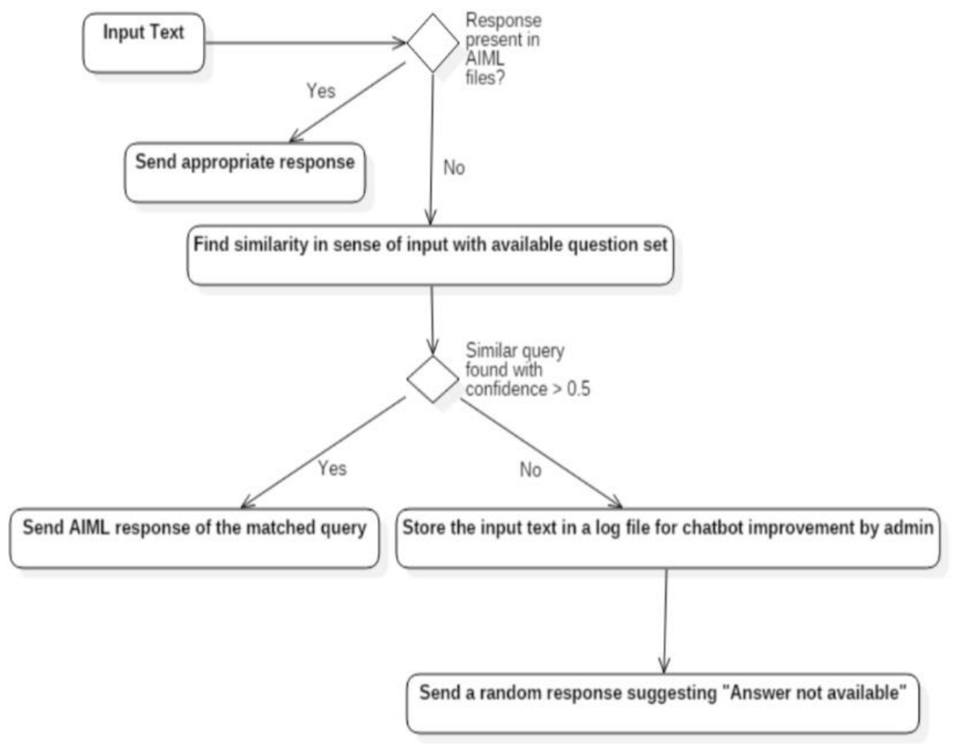

Input Text
Response present in AIML files?
Yes
Send appropriate response
No
Find similarity in sense of input with available question set
Similar query found with confidence > 0.5
Yes
No
Send AIML response of the matched query
Store the input text in a log file for chatbot improvement by admin
Send a random response suggesting "Answer not available"

Fig 4(g): College Related Response Activity

In this chapter i.e. Design every part of the proposed system is taken and use case diagrams, data flow diagrams and activity diagrams are made to make it easier for the reader to understand the work flow of the project.

5 Implementation

A. AIML:

The question and answer pairs are stored in the AIML files so that the base for the normal conversations can be created. The patterns in AIML are matched with the input from the user and the appropriate answer is given.

The sample AIML file structure is as follows:

```
<aiml version = “1.0.1” encoding = “UTF-8”?>

<category>

<pattern> HELLO USERNAME</pattern>

<template> Hello User! </template>

</category>

</aiml>
```

B. Lemmatization and POS Tagging Using WordNet:

Keywords were extracted from the input text which is known as information extraction. For e.g. “What is the distance from Shimla?” contain “distance”, “Shimla” as the keywords. Lemmatization and POS tagging were used to find appropriate Lemmas so that proper grouping can be done. Also, WordNet from Python 3’s NLTK package was used for mapping the words to their base from. For e.g. walking, walk, and walked should map to walk.

C. Semantic Sentence Similarity

There are various cases in which the user might write the same sentence in a different way. For e.g.,

Q1: Who is the hod of cse department?

Q2: Tell me about the hod of cse department.

[2]Here both the questions have the same meaning. There can be many other combinations for the same query. The problem here is that all those combinations cannot be added into the file. Hence, the performance of the system will get affected. To solve this problem, similarity is found between the input of the user and the data present in the question set. The input with the maximum score gets selected and then the proper answer is provided.[2]

The above mentioned score is calculated by averaging the similarity of the individual keywords of those sentences. The word with maximum similarity is matched between among the keywords of the sentences. Then the similarity score is averaged to make the sentence similarity.

Another approach is the implementation of word similarity using the Path Similarity and Wi-Palmer(WUP) Similarity is used.

[2]Like WordNet, a hierarchical structure is assumed where the path similarity compiles shortest number of edges from one word sense to another. It is to be noted that the longer word sense are less similar than those with a smaller path distance. E.g. car, bus and cycle(we expect that bus is similar to car and not a cycle). The Wu-Palmer metric calculates the distance using the weights in the edges.[2]

D. Log File:

A log file is also maintained which store those queries which the chatbot was not able to answer. It is the work of the admin to view the log file and the responses of the sentences. All this is done into the knowledge base. This is a very important method to improve the overall functioning of the chatbot.

Tools and Techniques

Tools used:

# A computer system running on Windows

# Notepad++ used for editing the python, XML or HTML code.

# NLTK, AIML, FLASK and SQLITE packages in python 3.

# OpenNLP

# Web Browser(Chrome)

# Sqlite Browser (online)

Techniques, Languages used:

# HTML, CSS, Javascript

#XML(AIML framework)

#SQLITE

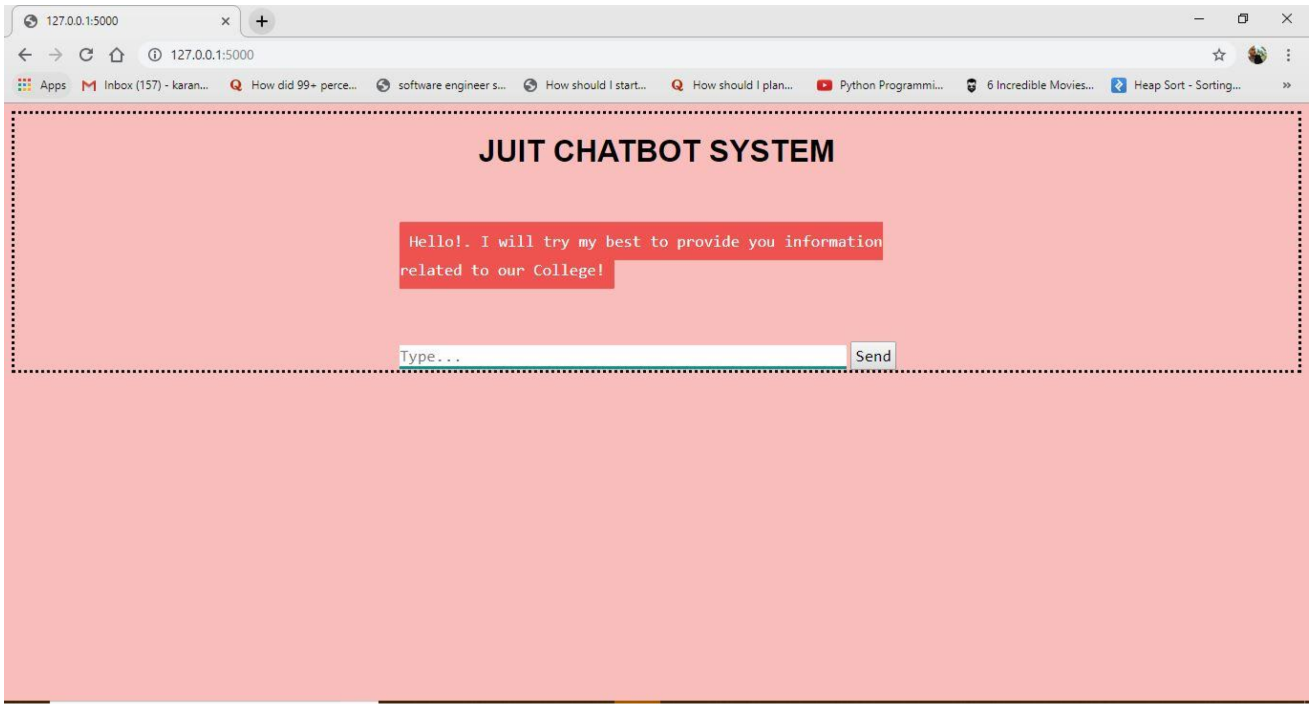
#Parsing and Substitution, the same which is used in Eliza.

#POSTagging NLP(Natural Language Processing)

#Database Management System

6 Results

The system after hosting:



JUIT CHATBOT SYSTEM

Hello!. I will try my best to provide you information related to our College!

Type...

Fig 6(a): The application College Enquiry Chatbot

The system works fine and the bot answers the queries of the users. Some humour has been added into the bot database so as to make it less boring to interact with. Also there seems to be some problem with the .db file as it is not correctly showing the CGPA. Everything else is working properly.

Conclusion and Future Work

What we can conclude from the above situation and usage is that greater the database and more the models and use cases, the better is the reaction produced for the client. In any case, the issues are many. For adjusting more uses, the scope changes from the investigation of AI to language examine.

We additionally need to recollect that we are taking a shot at a cell phone. The NLP is very broad and maybe utilizing them on a server and isolating this application into customer and server side application can fathom the speed issue as when we do that the speed of the program won't be restricted to the equipment.

We live in a time of intelligent technology. Our watches let us know the time, however they likewise remind us to work out. Our telephones prescribe the best places to eat, and our PCs foresee our inclinations, helping us to do our everyday work all the more productively.

All things considered, these advanced collaborators show just a modest bit of Artificial Intelligence (AI).

Google's Google Assistant or Apple's Siri is a new form of AI which is used by millions of people every day.The vast majority of the purchaser level computerized reasoning applications we're communicating with are customized to utilize gathered information put away in their databases to improve a response to inputs, which prompts a superior reaction inside foreordained parameters.

ELIZA, while itself a moderately straightforward framework, is significant from the perspective of understanding human insight and theoretical machine knowledge as it incites us to ask what knowledge really is and what can be viewed as authentic insight or "unique" deliberateness in a machine. It additionally influences us to consider the measure of what is seen as canny conduct of an operator lies.

In spite of the fact that the discussions are linguistically right, they don't pass on an important discussion as the program doesn't appear to have the capacity to comprehend whatever is being said by the client and henceforth the client never feels associated while chatting with Eliza or Cleverbot. There is by all accounts no "memory" and not in any case a feeling of character.

Such a machine is of no utilization to utilize.

Be that as it may in the event that we build up a program which can hold every one of the records and the "recollections" of the client, his sentiments the names and every one of the things which matter to him/her, we can truly approached assemble something which can assist a client when he is focused and act like an instructor.

This can be accomplished with the accompanying two segments which we plan on actualizing later on.

NLP is required with the goal that the parsed information can be "comprehended" by the application, things, for example, a client's temperament, his sentiments, if there is a funniness in the announcement, the names and places referenced in the info. Different things, and so forth.

Database: A Database is required with the goal that the parsed information can be put away by the application, things, for example, a client's mind-set, his sentiments, and his state of mind previously. What's more, on the off chance that he is feeling also worried, a pressure call or some other careful steps can be taken. Different things of significance to the client can likewise be put away.

For example, names and places. Different things, and so forth. A Database can be utilized with the goal that the client can feel that he is associated with somebody and not conversing with a machine which overlooks each time whatever the client says.

Future Applications

The 21st century has seen the advancement in technology and its impact on the people. In many ways, the technology innovation has helped to make the lives of the people easier.

It is widely known that information is carried out from one person to another. When someone does not know about something, he asks the other person. That way the information is carried on. Keeping this fact in mind when the computers came into existence, some of the scholars and scientists came up with the idea of making them answer the problems of the people. By creating this portal, the users could get their problems answered.

Google started with a simple web application which answered the queries of the people but that was something different from one on one talk. So it introduced the concept of Google Assistant through which every user could talk. Not only the user can talk but it is equipped with top notch features today.

You might have heard the statement “Something that works perfectly doesn’t mean it can’t be upgraded”. Keeping this thought in mind, this project can also be upgraded to a much higher extent according to the need.

One of the major additions that can be done is the introduction of Voice API in the near future. The users will give the queries through their voice and the system will give text and voice output both.

References

Published Papers:

1. Weizenbaum, Joseph. "ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine," Communications of the Association for Computing Machinery 9 (1966).

2. Suchman, Lucy A. Plans and Situated Actions: The problem of human-machine communication (Cambridge University Press, 1987).

3. Sagar Pawar, Omkar Rane, Ojas Wankhede and Pradnya Mehta. “A Web Based College Enquiry Chatbot with Results”, (IJIRSET, April 2018).

4. Pratike Salve, Vishruta Patil, Vyankatesh Gaikwad and Prof. Girish Wadhwa: “College Enquiry Chatbot”. (IJRTCC March 2017).

5. Tarun Lalwani, Shashank Bhalotia, Ashish Pal, Shreya Bisen, Prof. Vasundhra Rathod: “Implementation of a ChatBot system using AI and NLP”.

Outline Links:

1. https://www.tutorialspoint.com/aiml/aiml_introduction.htm

2.https://blog.recime.io/using-aiml-and-nlp-to-create-a-conversation-flow-for-your-chatbot-fea63d09b2e6

3. https://www.quora.com/How-would-you-compare-NLP-NLU-to-AIML

4. https://www.devdungeon.com/content/ai-chat-bot-python-aiml

5.https://www.quora.com/Does-AIML-count-as-artificial-intelligence

6.https://towardsdatascience.com/develop-a-nlp-model-in-python-deploy-it-with-flask-step-by-step-744f3bdd7776

7. https://sqliteonline.com/