BioGecko

Vol 12 Issue 03 2023

ISSN NO: 2230-5807

Chatbot for Educational Institution

Dr. D. Thenmozhi[1], Kalaivani A[2], Manickam L[3], Aadithya Hariharan A N[4]

[1]Associate Professor, Dept. of Computer Science and Engineering, SSN College of Engineering, Chennai.
[2]Research Scholar, Dept. of Information and Communication Engineering, Anna University, Chennai.
[3,4]Bach. Of Engineering, Dept. of Computer Science and Engineering, SSN College of Engineering, Chennai.

Abstract
Loads of information are stored in a university's webpage. Surfing through the website to get the information related university becomes time-consuming and exhausting as it involves webpages within webpages. But this data can be used to provide answers to university related queries using a chatbot. This project aims to implement online chatbot system to assist users who access college website and allows users to communicate with college chatbot to generate a response. The user query is tokenized, pre-processed to extract the keywords. The keywords are compared with the data extracted from web-scraping the official college website using similarity measures, thereby generating the response. It can also provide academic information of the students after validating the student credentials. It is capable of making conversations, answer the queries, display faculty information and provide the events information of college.

Keywords: Online Chatbot System, Text to Speech, Information Extraction.

1. INTRODUCTION

A chatbot, also called a chatterbot, is a piece of software that conducts online chat conversations instead of human live chat operators utilising text-based or text-to-speech communication [4]. Chatbots are computer programmes that can converse with users in everyday language, ascertain their goals, and react in accordance with data and rules that have been saved. The majority of internet chatbots are accessed through popup windows or virtual assistants on websites. A wide range of uses are possible for them, including business (including e-commerce via chat), education, entertainment, finance, health, news, and productivity. In dialogue systems, chatbots are employed for several purposes, such as data collection, request routing, and customer assistance [11]. We develop a chatbot that can respond to inquiries about the educational facility [16]. The goal of this chatbot is to make the access of college related information easier. The search feature present in the college website may or may not fetch the relevant information, hence the user has to navigate through multiple webpages. This becomes a tedious process. Our chatbot makes this job easier by fetching the exact information required.

1.1 BACKGROUND

Numerous chatbots have been created and put to use for a variety of purposes in the past. Some of those chatbots are:

1.1.1 ELIZA

ELIZA was the name of the very first chatbot ever created. It was developed in 1966 by Joseph Weizenbaum and imitates speech using pattern matching and replacement methods.
The programme was designed to look and sound like human speech. Users entered phrases into a computer to operate the chatbot ELIZA, which then compared them to a list of potential prepared

responses. It makes use of a made-up psychotherapist in the screenplay. The screenplay had a big impact on artificial intelligence and natural language processing, and universities all around the country started producing copies and revisions of it.

1.1.2. PARRY
In an attempt to simulate the condition, psychiatrist Kenneth Colby created PARRY in 1972, a programme that mimicked the behaviours of a schizophrenic patient. The way that people think is mimicked by the natural language system PARRY. It operates by the use of a complex system of assumptions, attributions, and "emotional responses" that are triggered by varying weights given to language inputs. A modified Turing test was used to verify the accuracy of the PARRY programme. Early in the 1970s, human interrogators using a remote keyboard to communicate with the programme were only slightly better than chance at distinguishing PARRY from a real individual displaying unusual behaviours.

1.1.3.Jabberwacky
Developer Rollo Carpenter made the chatbotJabberwacky in 1988 with the goal of providing a humorous simulation of casual human interaction. The creation of Jabberwacky led to more technological innovations, and its website is used for academic research. "Contextual pattern matching" is an AI technique used in the chatbot's construction.

1.1.4.Dr.Sbaitso
Dr. Sbaitso, developed by Creative Labs for MS-Dos in 1992, was one of the first attempts to incorporate A.I. into a chatbot. The chatbot earned prominence for being a fully voice-operated programme that conversed with users just like it were a psychotherapist, even though most of its responses were simple and repetitive, such as "Why do you feel that way?" rather than engaging in serious debates.

1.1.5.Smarter Child
In 2001, a chatbot named SmartChild was developed as a previous version to Siri. It was attainable on AOL IM and MSN Messenger, where users could engage in funny talks and quickly get information from various services. Later, Microsoft developed a version of SmartChild specifically for American adults aged 18 to 24. But at that time, the majority of users had already stopped using AIM. The account was set up specifically for conversation.

1.1.6.Siri
A variety of Apple operating systems, which includes iOS, iPadOS, watchOS, macOS, tvOS, and audioOS, come with \textbf{Siri}, an intelligent virtual assistant. It provides suggestions, replies to questions, and completes tasks through sending requests to a network of online services using voice commands, gesture recognition, and a user interface that can comprehend natural language.

1.1.7. Google Assistant
The majority of individuals use their mobile phones and home automation gadgets to access Google Assistant, an AI-based virtual assistant. It's a Google creation. It distinguishes itself from Google Now, the company's previous virtual assistant, by offering interactive two-way communication. Even though it also accepts keyboard input, Google Assistant primarily allows users to speak with it. This virtual assistant may do a variety of functions, including playing games, showcasing information from the user's Google account, responding to questions, setting up events and alarms, modifying hardware settings on the user's device, and more.

1.1.8. Cortana
At the Microsoft Build 2014 developer conference, Cortana first made an appearance. Later, it was added to Windows 10 computers and Windows phone gadgets. This virtual assistant utilises algorithms and voice recognition to hear and follow voice commands. Cortana is capable of doing a wide range of tasks, such as creating and organising lists, setting reminders based on time, place, or people, sending emails and SMS, playing games, holding informal chats, and searching for documents, locations, and other information.

1.1.9. A.L.I.C.E.
Richard Wallace developed the artificial linguistic internet computer entity (A.L.I.C.E.) in
1995. A.L.I.C.E., which stands for Artificial Linguistic Internet Computer Entity, is a chatbot that uses heuristic pattern matching to carry on discussions. It was initially called Alicebot for the computer it used. The program's operational language is artificial intelligence markup language (AIML), an XML structure that describes conversational rules. In 1998, Wallace rewrote the programme in Java, and in 2001, he published an AIML specification. Later, other programmers created open-source and free versions of A.L.I.C.E. in a number of different programming languages and other languages. A.L.I.C.E. pretends to be a young woman and has an online imitation chat with a real person. She divulges private information, including her age, her hobbies, and other intriguing details.

1.1.10. ALEXA
Designed by Amazon and released in 2014, Alexa is a smart personal assistant. It's incorporated into some devices, including the Amazon Echo, Echo Dot, Echo Show, and others. Independent companies also produce devices with Alexa integrated in, and an Alexa app is available. You may use voice commands to ask Alexa to perform a variety of tasks, such as play music or find an Italian restaurant. With Alexa, you can control smart home appliances, search the internet, make shopping or to-do lists, set alarms, stream podcasts or audiobooks, get news or weather updates, and do a lot more with only your voice.

1.1.11. ChatGPT
In 2021, the OpenAI team developed ChatGPT, a sizable language model. Its primary goal is to enable users to generate text from input that appears human-like. Language translation and dialogue creation are both possible uses for the method. Having access to a vast amount of data, ChatGPT may create content that is frequently hard to distinguish from writing by people. It has drawn accolades for its ability to generate language that sounds natural and for its adaptability to a variety of situations. It's essential that one remembers that ChatGPT developed the same abstract.

1.2. CATEGORIES OF CHATBOT

A chatbot can simultaneously be a member of numerous categories, each of which has been established using a simple criterion.

- Generic chatbots are able to provide information on any subject. Cross- or open-domain chatbots can operate across several domains, whereas domain-specific chatbots can only answer inquiries pertaining to that domain's body of knowledge.
- Contrarily, interpersonal chatbots are created to perform tasks like restaurant or airline reservations or FAQ searches without becoming a good buddy. Contrarily, intrapersonal chatbots

are close friends that live in the user's domain and are aware of their demands. Inter-Agent chatbots allow for conversation with other chatbots.

- Users interact with informative chatbots to receive precise information that is kept in a fixed source. Conversely, conversational chatbots interact with users in a manner similar to that of actual people. Task-based chatbotsspecialise in soliciting information from users and providing pertinent responses, handling specific tasks like accommodation reservations.
- Chatbots that generate responses can be categorised as Rule-based, Retrieval-based, or Generative-based chatbots.
- A chatbot that uses human processing in at least one aspect of its operation is referred to as human-mediated. The limits of fully autonomous chatbots might be overcome by adding human intelligence to them.
- Based on the permissions, chatbots are classified as either open-source or commercial.
- Based on the communication medium they employ—which may be text, speech, a picture, or a combination of these—chatbots can also be categorised. The ability of chatbots to reply to photographs has recently advanced, enabling them to not only identify things in the images but also to comment on them and convey emotions.

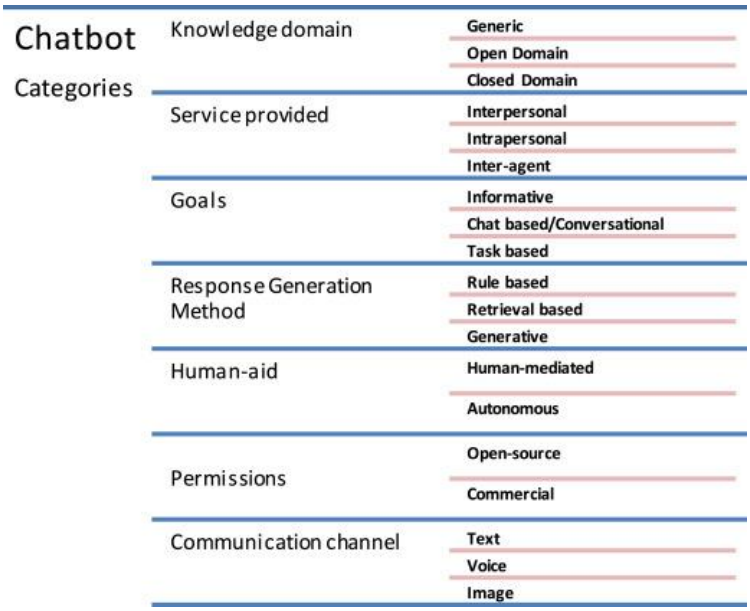


Figure 1: Chatbot Categories

1.3. LIMITATIONS OF CHATBOT

The creation and use of chatbots is a rapidly developing topic that is strongly related to artificial intelligence and machine learning. Although chatbots have many benefits, there are some restrictions on their functionality and application. These restrictions are, however, gradually being addressed and overcame as technology develops. A few of the most typical restrictions are:

- As a result of their limited and fixed input/output databases, Chatbots may fail when attempting to handle unsaved queries.
- Chatbot's language processing abilities are crucial to its efficacy, but they might be hampered by variances like accents as well as errors.

- Non-linear interactions comprising back-and-forth on a certain subject with a user are challenging for chatbots to deal with.
- A considerable quantity of conversational data needs to be collected for chatbot training. A sizable dataset of natural-language phrases is often needed for training generative models, which construct replies word-by-word based on user input using deep learning techniques.
- The prospects for interaction may be limited by chatbots' limited capacity to respond to several queries at once.
- Due to their poor comprehension, which draws attention to the fact that their enquiries are being handled by machines, some customers, especially those from older generations, may feel uneasy with chatbots, as is typical with technology-driven improvements in well-established services.

1.4. PROBLEM DEFINITION

To design a chatbot for an educational institution using Natural Language Processing (NLP). This chatbot is designed to provide answers for queries regarding the admissions, fees structure, examinations, placements, transport, hostel and much more information regarding SSN college.
The objective for this chatbot arises due to various reasons - many details are unavailable in college website, an outsider would not know where to search for a particular piece of information and the students themselves finding difficult to get the details. The search feature available in the college website doesn't yield fruitful results during most of the times and returns no results for user's queries. So, to solve all of this problems, we design a chatbot which would make the user to ask some question regarding the college and our product would deliver the answer. This paper is organized as follows. Related works are described in the section 2. The techniques of our models are presented in Section 3. Section 4 has the experimented setup. In section 5, we conducted research and its findings. Finally, we summarize our findings and suggest ways to improve our work in section 6.

2. RELATED WORK

Boudhir Anouar Abdelhakim et al. [5] developed a chatbot using natural language processing and machine learning for healthcare assistance. They used artificial intelligence concepts like natural language processing, natural language understanding, natural language generation, automated speech recognition for developing their chatbots. For answering the queries, it uses both manual and automated training. In manual training, a list of frequently asked questions is mapped with the answers and fed into the chatbot. Different types of documents like QandA document, policy documents are given to the chatbot and asked to train itself in automated training. TarunLalwani et al. [9] built a chatbot using AI and NLP. It aims to provide a chatbot system that will dispense all answers relating to domains such as admission, examination cell , notice board, attendance, placement cell and other miscellaneous domains. The college student and employees can freely upload their queries. This aims to act as a fast, standard and informative chatbot to bestow users with righteous information.

NiranjanDandekar et al. [1] developed a chatbot using techniques like pattern matching, parsing, A.I.M.L, Deep learning. This chatbot uses both methods of fetching data and artificial intelligence. Intelligence comes into play when the user typed query is not in the database. After fetching the answer, it is converted to a voice note.

Voice based chatbot is developed using natural language processing by Meet Popat et al. [4] The input is given to the chatbot via speech. After the user gives the input, it does speech recognition converting speech to text. And the normal process of fetching the answer is done.

Papiya Mahajan et al. [11] developed an healthcare Chatbot using natural language processing. It aims to identify the illness and supply basic information regarding the illness before consulting a doctor. It helps the patients apprehend additional details regarding their illness and improves their health. The system application uses question and answer protocol within the style of chatbot to answer user queries. This system is highly exhaustive and takes ample amount of time to provide a response for user's query.

An educational chatbot is built by Dr.M.Senthil Kumar et al. [8] to provide answers for the queries asked about their college. It aims to provide the user with information related to college and student details. This educational chatbot focuses on efficiently responding to the queries of the user. It aims to fetch pre-existing details present in the database and doesnt provide features to compute students' required internal marks or individual grades.

In this paper, Rishabh Shah et al. [2] developed an intelligent chatbot using Natural Language Processing. The query processing is done by N-gram division algorithm and the response generation is done by Long Short Term Memory(LSTM).

The paper by TraianRebedea et al. [14] describes an approach to the idea of identifying the most important facts in texts describing the life (including the personality) of an historical figure for building a conversational agent that could be used in middle-school CSCL scenarios.

College Information Chatbot system was developed by Amey Tiwari et al. [3] to provide details about their college. It consists of modules like Admin module, Text-to-speech, and the traditional chat system. The system uses Artificial Intelligence to generate the response for the queries.

HrushikeshKoundinya K et al. [15] developed a smart college chatbot which would answer for the queries about the college. This chatbot system is an internet application that gives an answer to the broken down queries of an user regarding their college. The user can question about the college related activities with the assistance of this web application.

College related activities, for example, admissions, academics, Intake, and other social activities. It will support the undergraduates/other user to be informed about the college activities. It uses various methodologies like tokenization, lemmatization, WordNet Algorithm, SHA-256 Encryption scheme.

A chatbot using Python was developed with Graphical User Interface by Susmitha Mary et al. [6] which would answer the queries about the college. The proposed system is a web application. The user is asked to login before he/she asks their query. After login, the user is given with the list of buttons with which he/she can ask their queries.

A chatbot for college information system was developed by Prof. ManishaSonawane et.al. [12] It is a chatbot that gets user input credentials and uses database to validate those credentials. Later on,it receives queries from user and analyses the query inorder to provide response to it. It uses Natural Language Processing technique to respond to the user by creating candidate response generator and response selector modules. Moreover,the college staff is relieved from answering the same queries again and again.

DarshilGada et al. [13] developed the college chatbot to remove the difficulty of access of data by providing a common and user friendly interface that uses natural language processing to interact and

solve queries of college students and teachers. It extracts structured data from MySQL and uses pre-processing methods like tokenization, lemmatization and then queries for the keywords generated from processed data to retreive or provide a response using a response generator module.

2.1 GAPS IDENTIFIED

- Lacking the usage of cosine similarity and uses exhaustive entity recognition techniques which is time consuming [8,10].
- Lacking the feature of logging.Logging refers to updation of queries into the database for which appropriate answers weren't found which will be looked upon by the admin. No feedback mechanism for answers that do not exist [9].
- Highly exhaustive and takes ample amount of time to provide a response for user's query. The proposed system Fails to provide complete free of access and it has paid communication with an actual person. This question's the feasibility of the consumer or user. This doesn't have feedback mechanism too [11].
- The accuracy of the dataset present in this chatbot is questionable as it has been trained over un-labelled dataset. Also lacking the feature of logging and hence the queries cannot be processed upon to the server end. Takes some time to provide a response to the user's query [12].
- Not a question-answer oriented chatbot as it offers multiple options for users and lets them pick the kind of query they have for the chatbot. Fails to provide answers for user generated queries most of the time. Time consuming entity recognition techniques instead of techniques like n-gram,cosinesimilarity,Wu-Palmer similarity etc [15].
- It doesn't have the features to extract placement details from the college website as it doesn't implement the concept of web scraping. Delayed response time for user's queries frequently [13].

3. PROPOSED METHODOLOGY

The actions and methods that will be employed to accomplish a specific research goal or target are described in the planned methodology. It primarily functions as a roadmap for carrying out research or finishing a project. A description of the research concept, data collection strategies, data analysis techniques, and other pertinent procedures are frequently included in the suggested methodology.

The quality and correctness of the research findings are mostly determined by the proposed technique, which is why it is so important to the research process. It aids in ensuring that the study is carried out methodically and rigorously and that the outcomes are trustworthy and valid. To ensure that the goals of the research are satisfied, the methodology must be carefully developed and carried out.

To give a thorough explanation of how the research will be carried out, the recommended methodology should be presented in the research proposal or report in a clear and simple manner. The methodologies and procedures employed should also be justified, stating why they are suitable for the current research challenge. A carefully thought out and implemented technique can increase the validity of the study findings and improve knowledge in the topic.

3.1.Detailed Design

The detailed design of the architecture is explained below.

Architecture for Student Module

- The input credentials is given by the user which is then validated using the MySQL database.
- The input query is now typed and the pre-processing (Removal of unwanted words, stopwordsetc) of the query happens.

- Now the query is broken into tokens with which we would compute the answer.
- For the calculation query(say calculation of gpa, cgpa, internals) etc, the application would fetch the data from the database, compute it and display.

Architecture for Commoner Module

- The SSN website and its corresponding URL's are web scraped using BeautifulSoup library, parsed and put into separate documents.
- The keywords of the documents are generated using the Yet Another Keyword Extraction method and is stored in another document.
- The query is given by the user and similar to the student module, the query is pre-processed and put down into a set of tokens.
- Now the query and the keywords are matched using the \textbf{Cosine Similarity} method and the document is retrieved which has the higher cosine value(Higher the value, similar the document).
- If there are more than one document relevant to the user query, the user prompts for which document to display.
- The user chooses what he/she wishes.
- The contents of the retrieved document is displayed.
- The application now prompts for the satisfication of the answer. If the user is satisfied, it prompts for the next query. If the user isn't, then the query gets saved in a log file which is managed by the admin.

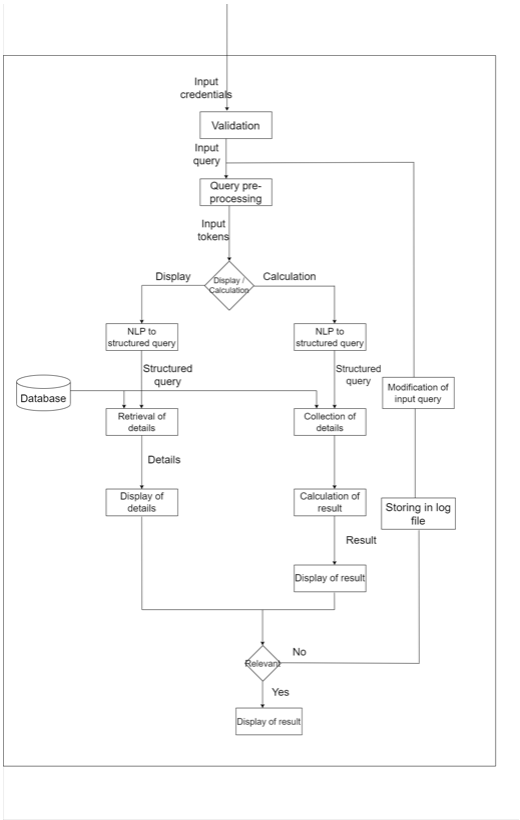


Figure 2:Student Architecture

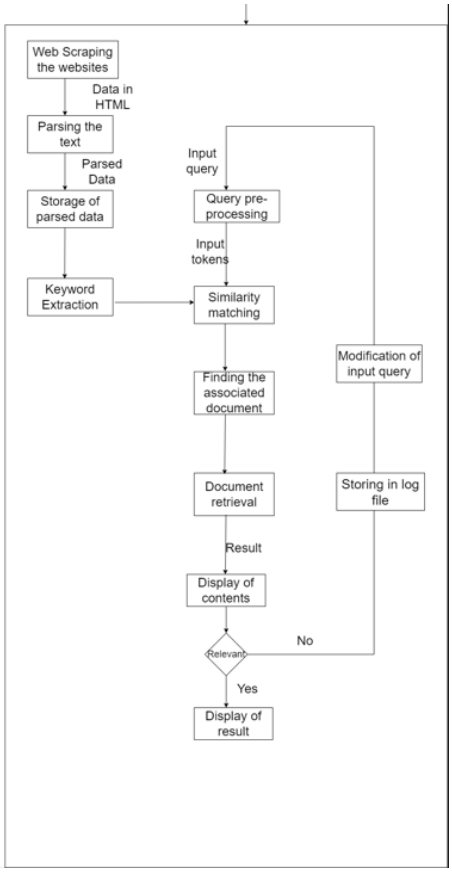


Figure 3: Architecture of Commoner Module

3.2. Explanation of Modules

3.2.1 Chatbot for College Student

User Validation

User (College Student) types in the user credentials. The credentials of all the students are stored in the MySQL database. The database connection is made through the mysql.connector module. The user typed credentials are matched with the credentials in the database which is fetched using the cursor, execute and fetchall methods. If there is a match, then the chatbot prompts for a query from the user. Else it returns Invalid and exits.

Query Pre-processing

User types the input query. The input query is pre-processed. i.e., it is converted into a form that makes the response generation even more accurate. It is done through the following steps.

a. Removal of stopwords – Stopwords are words which have less or no meaning so that removing the word won't affect the essence of the sentence. The query is tokenized using the word_tokenize from nltk.tokenize library. The stopwords are fetched using the stopwords from nltk.corpus library. They are removed.

b. Removal of unwanted details - The details other than the main ones like the semester, department are removed.

c. Conversion of word to numbers – The words are converted to numbers so that it would be easy to match with the data in the database.

NLP to Structured Query
The Natural Language Processing query is converted into the structured query that is SQL query so that it could fetch the particular data from the database. The keywords of the query are mapped to the format of SQL query \cite{7}.

Display Operation
Retrieval of details:
The details like the hostel mess menu, canteen menu etc are retrieved from the database using the cursor, execute and fetchall methods.

Display of details:
The retrieved details are displayed to the user.

Calculation Operation
Collection of details:
The details like the CAT marks of the student, internal marks, attendance etc are collected from the database using the cursor, execute and fetchall methods.

Calculation of result:
The result is calculated using the details collected from the database in the previous step. For eg, for the internal calculation, the CAT marks are collected along with the 10 mark internals and the 40 mark internals is calculated using these data.

Display of result
The calculated result is displayed to the user.

Storing in log file
If the answer provided is relevant to the user, the chatbot prompts for the next question. If not, the chatbot stores the query in a log file so that the admin can see and improve the chatbot.

Modification of input query
If the answer provided is not relevant, user modifies the same query and provides it to the chatbot.

3.2.2. Chatbot for Common People
Web scraping the websites:
The SSN website and the corresponding websites associated with the SSN website is scraped using the BeautifulSoup library. Using the get method in requests module, we open the website. Using the BeautifulSoup class, the website data is fetched and the particular data is fetched using the find and find_all method.

Parsing the text

The data in HTML format is converted into the list of sentences by looping through the links and fetching the text using the text attribute.

Storage of parsed data
The scraped text is written in a text document separately.
Keyword extraction
The keywords are extracted from the text documents which contain the scraped text. The algorithm that we use for the keyword extraction is Yake(Yet Another Keyword Extraction). The text document is given as a parameter to the extract_keywords method in yake module and the keywords are generated. These keywords are stored in another text document.

Query pre-processing
User types the input query. The input query is pre-processed. i.e., it is converted into a form that makes the response generation even more accurate. It is done through the following steps

a. Removal of stopwords – Stopwords are words which have less or no meaning so that removing the word won't affect the essence of the sentence. The query is tokenized using the word_tokenize from nltk.tokenize library. The stopwords are fetched using the stopwords from nltk.corpus library. They are removed.

b. Removal of unwanted details - The details other than the main ones like the semester, department are removed.

c. Conversion of word to numbers – The words are converted to numbers so that it would be easy to match with the data in the database.

Similarity matching
The input query is matched with the documents containing the keywords to the main document. The algorithm that we use for finding the similarity between the query and the keywords is Cosine Similarity. The query and the keywords are converted into the vector format and the comparison begins. The best match is found using the greatest cosine value. Lower the angle between two documents, greater the cosine value.

Finding the associated document
The most similar match is found in the previous step. The document associated with the document which has the matched keyword is fetched.

Document retrieval
The document matching with the doucment which has the matched keyword is retrieved.

Display of contents
The retrieved document contents are read using the file handling methods like open, read and close in Python. The read contents are displayed to the user.

Storing in log file
If the answer provided is relevant to the user, the chatbot prompts for the next question. If not, the chatbot stores the query in a log file so that the admin can see and improve the chatbot.

Modification of input query
If the answer provided is not relevant, user modifies the same query and provides it to the chatbot.

3.3. Implementation
Our chatbot is used by both SSN students and common people. Common people clarifies their queries about the college and SSN students get to know their academic details through our chatbot. Natural Language ToolKit(NLTK) module for tokenizing the input query, removing the stop words and spelling mistakes.

Web Scraping is used for extracting the data from the website. We load the SSN Website URL and the corresponding URL's into the Web scraping and get the response as an XML or HTML file. Now we use BeautifulSoup module to parse the XML file and convert to a text document. Likewise we maintain a series of documents for different websites. The keywords corresponding to a website are extracted using the Yake module. We receive the query from the user and perform text pre-processing steps to process the query. Similarity matching is done between the query and the keyword. After finding the most similar keywrod, the document corresponding to the matched keyword is extracted. We display the contents of the webpage which got associated with the matched keyword.

SSN Student types the login credentials. The credentials are checked using the MySQL database. We receive the query from the user and perform text pre-processing steps to process the query. If the query is display query, then the details are fetched from the MySQL server and displayed. If it is based on calculation, the corresponding details are fetched, calculated and then displayed to the user.

3.4. Data Collection
Web scraping:
The research makes use of a method known as web scraping to collect unstructured data from the college's website. Writing code to crawl through the webpage's HTML and retrieve the necessary information is required for this. The project saves the unstructured data in a text file after it has been scraped. Given that programming languages make it simple to read and write to, this method is frequently used to store massive amounts of text data.
BeautifulSoup Library is used for web-scraping the data.

BeautifulSoup
A well-known Python package for data parsing and web scraping is called Beautiful Soup. It is intended primarily for extracting data from files like HTML and XML, two of the most widely used file types for web sites.
The versatility of Beautiful Soup's parser compatibility is one of its main advantages. A parser is a programme that examines an HTML or XML file's structure and transforms it into a parse tree, a data structure that is simple to edit and search.

JSON file:
The project employs text files as well as JSON files to hold key-value objects in addition to text files. JSON (JavaScript Object Notation) is a simple data exchange format that is simple for both humans and machines to read, write, parse, and produce.

MySQL:

Working with structured data, specifically student information, is another aspect of the project. The project uses a MySQL database to hold this data. Popular relational database management system MySQL makes it possible to store and retrieve structured data quickly.

In conclusion, the project extracts, stores, and retrieves numerous types of data from diverse sources using a range of data storage techniques and technologies.

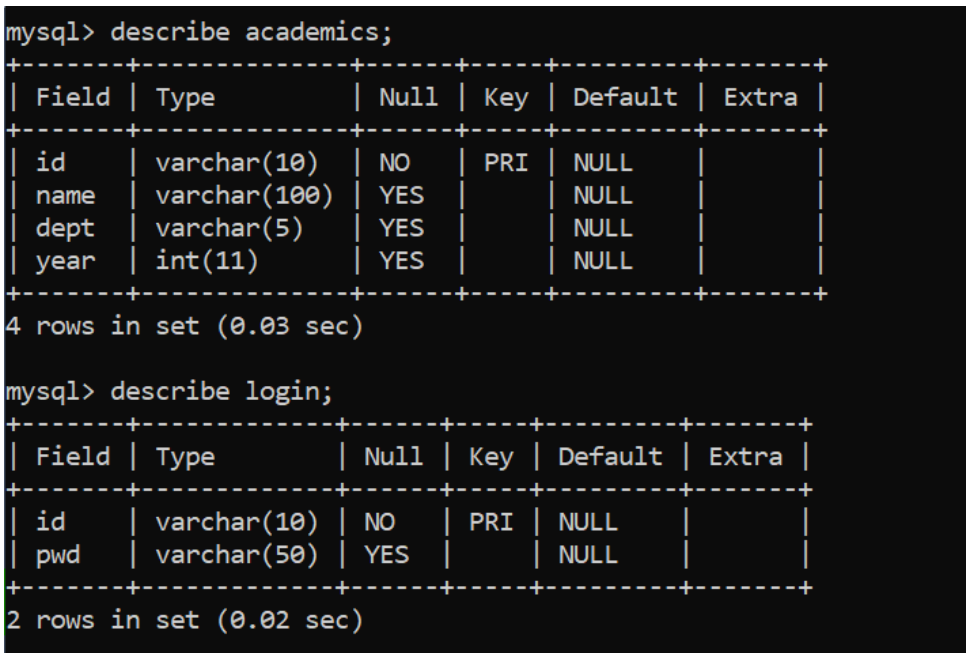
4. EXPERIMENTAL RESULTS

4.1. Dataset Description

Database
We have tables for

- Login Details
- Academic Details
- End Semester Marks
- Cat Marks
- Internals

Login details table and the academic details table are depicted in figure 4. End semester marks table is shown in figure 5. The table refers to the semester 1 of all the departments. The subject names only differ for different semesters and departments. Cat marks table is shown in figure 6. The table refers to the cat marks of CSE department of semester 2. The subject names only differ for different semesters and departments. The internals marks table is shown in figure 7. The table refers to the internal marks of CSE department of semester 2. The subject names only differ for different semesters and departments.



```
mysql> describe academics;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | varchar(10) | NO   | PRI | NULL    |       |
| name  | varchar(100)| YES  |     | NULL    |       |
| dept  | varchar(5)  | YES  |     | NULL    |       |
| year  | int(11)     | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
4 rows in set (0.03 sec)

mysql> describe login;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | varchar(10) | NO   | PRI | NULL    |       |
| pwd   | varchar(50) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.02 sec)
```

Figure 4: Login and Academics Table

```
mysql> describe sem1;
+---------+------------+------+-----+---------+-------+
| Field   | Type       | Null | Key | Default | Extra |
+---------+------------+------+-----+---------+-------+
| id      | varchar(10) | NO   | PRI | NULL    |       |
| uen1176 | varchar(2)  | YES  |     | NULL    |       |
| uma1176 | varchar(2)  | YES  |     | NULL    |       |
| uph1176 | varchar(2)  | YES  |     | NULL    |       |
| ucy1176 | varchar(2)  | YES  |     | NULL    |       |
| uge1176 | varchar(2)  | YES  |     | NULL    |       |
| uge1177 | varchar(2)  | YES  |     | NULL    |       |
| uge1197 | varchar(2)  | YES  |     | NULL    |       |
| ugs1197 | varchar(2)  | YES  |     | NULL    |       |
+---------+------------+------+-----+---------+-------+
9 rows in set (0.01 sec)
```

Figure 5: End Semester marks Table(Sem1)

```
mysql> describe sem2_cse_cat1;
+---------+------------+------+-----+---------+-------+
| Field   | Type       | Null | Key | Default | Extra |
+---------+------------+------+-----+---------+-------+
| id      | varchar(10) | YES  | MUL | NULL    |       |
| uen1276 | int(11)     | YES  |     | NULL    |       |
| uma1276 | int(11)     | YES  |     | NULL    |       |
| uph1276 | int(11)     | YES  |     | NULL    |       |
| ucy1276 | int(11)     | YES  |     | NULL    |       |
| uee1276 | int(11)     | YES  |     | NULL    |       |
| ucs1201 | int(11)     | YES  |     | NULL    |       |
+---------+------------+------+-----+---------+-------+
7 rows in set (0.01 sec)
```

Figure 6: Cat Marks Table(Sem2, Cse)

```
mysql> describe sem2_cse_internals;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| id      | varchar(10) | YES  | MUL | NULL    |       |
| uen1276 | int(11)     | YES  |     | NULL    |       |
| uma1276 | int(11)     | YES  |     | NULL    |       |
| uph1276 | int(11)     | YES  |     | NULL    |       |
| ucy1276 | int(11)     | YES  |     | NULL    |       |
| uee1276 | int(11)     | YES  |     | NULL    |       |
| ucs1201 | int(11)     | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
7 rows in set (0.01 sec)
```

Figure 7: Internals (Sem2, Cse)

Web Scraping
Totally 263 websites are web-scraped using the BeautifulSoup library.
Figure 8 is the list of documents scraped.

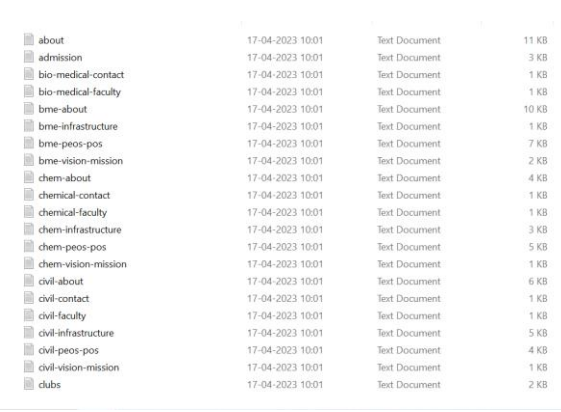


Figure 8: Scraped List

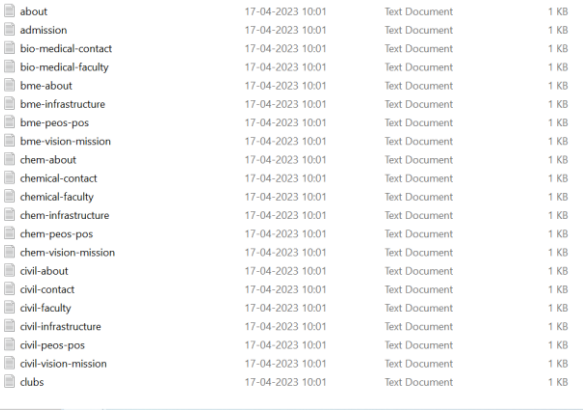


Figure 9: Keywords List

The list of keywords is presented in figure 9.

4.2. Ecosystem (Hardware and Software Specifications)

There are specified hardware and software requirements that must be satisfied in order to execute a given programme or piece of software on your computer smoothly and error-free. To avoid problems like sluggish performance or programme crashes, it is imperative to adhere to these specifications, which the software supplier lists.

The processor speed, memory, and storage capabilities of your computer are often factors in the hardware requirements for the software or programme you want to utilise. For instance, a CPU with a higher clock speed, such as an Intel Core i3 or higher, is required if you wish to run a programme that requires a lot of computing power, such gaming or video editing applications. Your computer should

also have 4GB or more of RAM, which is helpful for multitasking and running many programmes at once. Additionally, the software you intend to utilise might need a particular amount of hard disc space, typically 10–25GB.

The operating system and any additional software or tools required to run the programme are typically tied to the software requirements for the programme or software you want to utilise. You'll need to install the Python programming language on your computer, for instance, if the programme demands its use. It can also be necessary to use Jupyter Notebook, an Integrated Development Environment (IDE) for Python programming. Finally, you might need to install MySQL on your computer if the software needs a database to store data.

Before installing or using any programme, it is crucial to confirm that your computer satisfies the hardware and software requirements. By doing this, you can make sure that the software operates faultlessly and effectively.

4.3. Experiments Conducted

Figures 10, 11 and 12 are the college related queries.

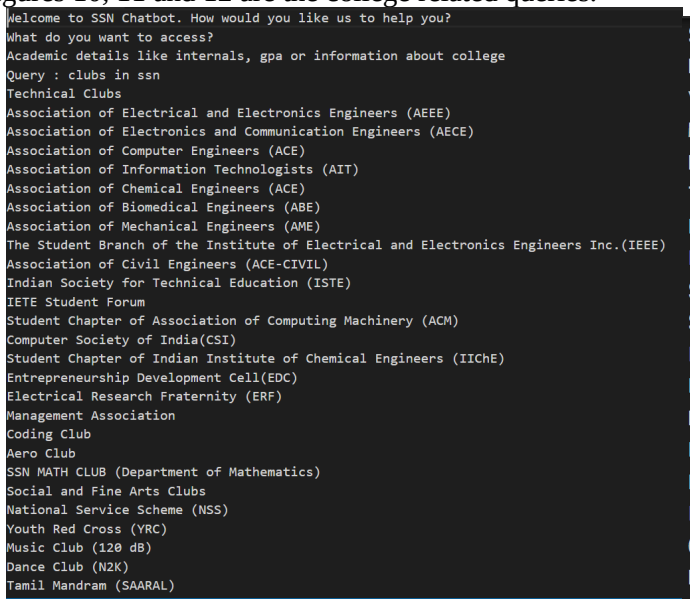


Figure 10: College Query1-Clubs

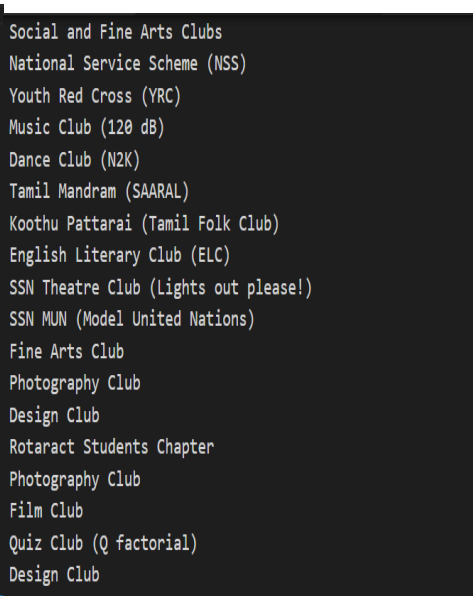


Figure 11: College Query2-Club Continued

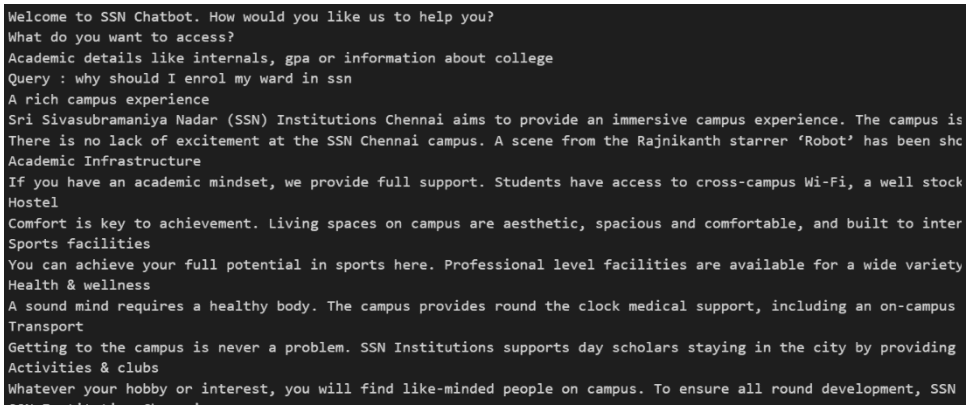


Figure 12: College Query - Why join SSN?

Figure 13 is the unsatisfied query. When an user is not satisfied with the answer the query gets stored in a log file. Figure \ref{fig:logFile} is the log file which is managed by the admin.

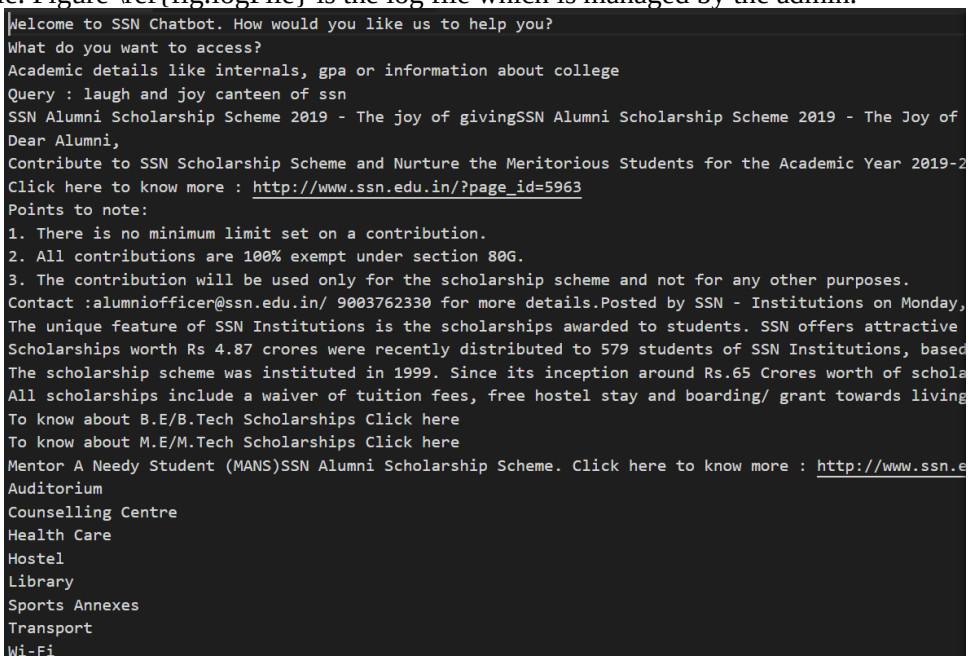


Figure 13: Unsatisfied answer - Laugh and joy canteen

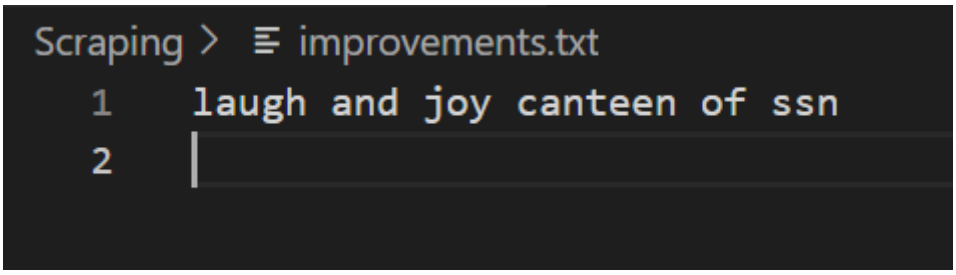


Figure 14: Log File

Figure 15 is the result where an user gets his/her GPA. Figure 16 is wrong credentials possibility. Figure 17 is the result where an user gets his/her internals. Figure 18 is the result where an user gets his/her estimated CAT-3.

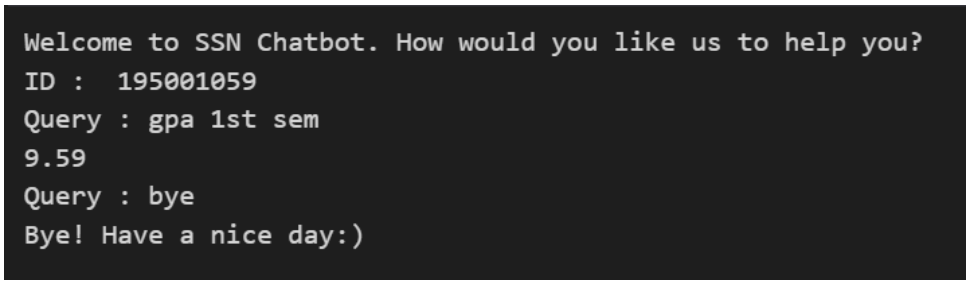


Figure 15: GPA

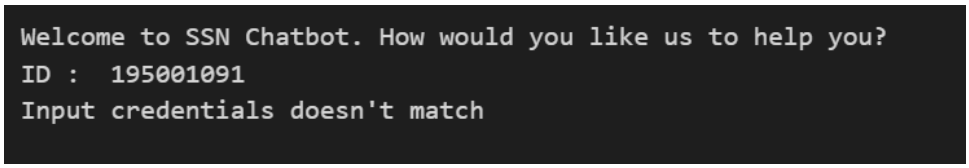


Figure 16: Wrong Credentials

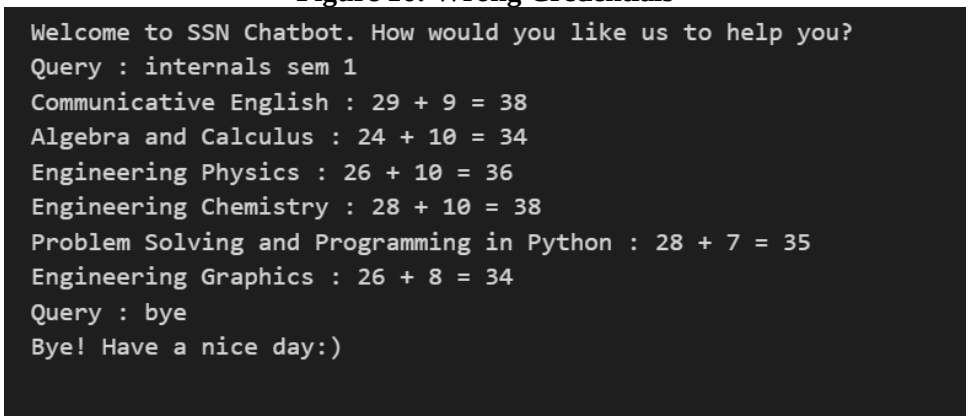


Figure 17: Internals

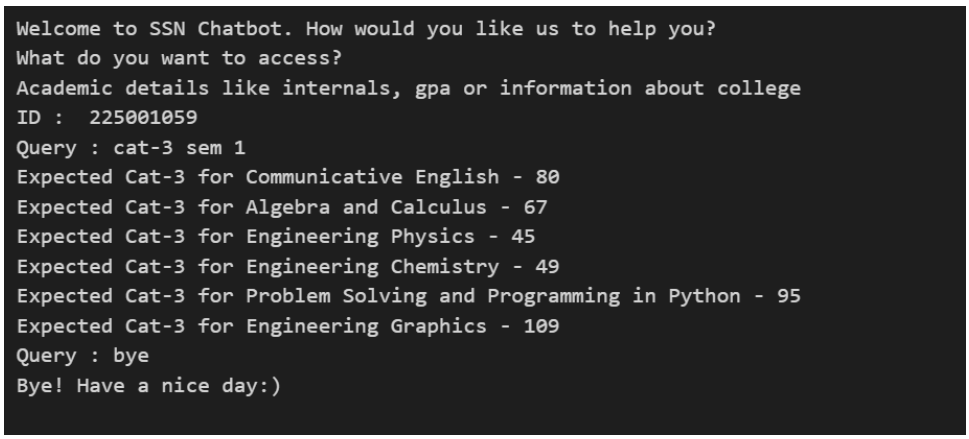


Figure 18: CAT-3

Performance Analysis

The effectiveness of a chatbot application is assessed by contrasting the responses to a random sample of user queries with previously provided responses to the same queries. To assess the performance of the chatbot, no particular performance indicator, such as the F1 score, Area Under Curve (AUC), or Confusion Matrix, is utilised.

In other words, the evaluation is carried out by contrasting the chatbot's answers with the predicted ones for a collection of questions. When a chatbot responds, it is either accurate if it is similar to the anticipated response or inaccurate if it does not. The proportion of the total number of questions for which the chatbot produced accurate answers serves as a measure of its accuracy.

The chatbotprogramme has responded correctly to 27 of 35 randomly selected questions and incorrectly to 8 of 35 questions based on the information provided.

Following formula can be used to determine the chatbot's accuracy:

Accuracy = (Number of accurate responses / Total queries) x 100%

In this situation, the chatbot's correctness can be determined as follows:

(27 / 35) x 100% = 77.14% accuracy

Therefore, for the supplied set of 35 random queries, the chatbot application's accuracy is 77.14%.

Accuracy	Percentage
27/35	0.77

Table 1: Accuracy

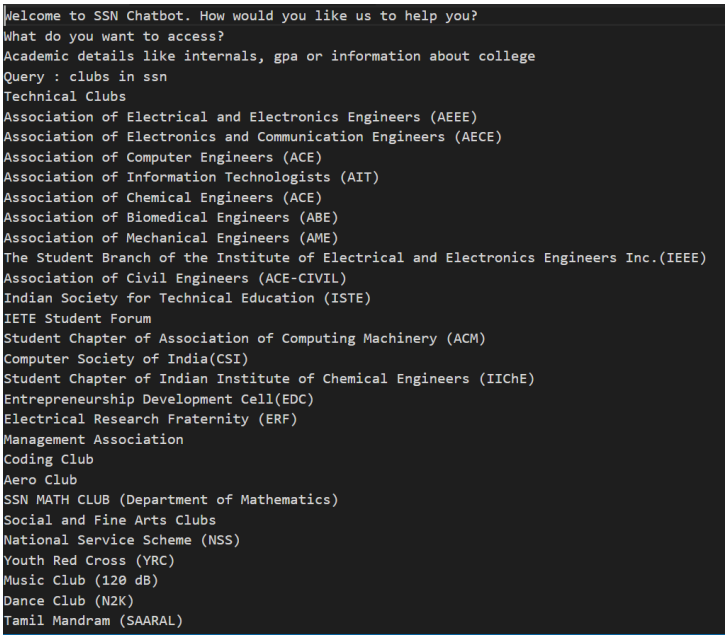


Figure 19: College Query – Clubs

Our chatbot yields the desired result.

Social Impact and Sustainability

The Chatbot can be used by anyone who needs information about educational institution to get responses to inquiries about admission, placement, fee structures, and other related topics. It is presumable that the chatbot is built to deliver trustworthy and correct information about the college.

It is crucial to remember that this project does not take health, safety, environmental, or cultural factors into account.

As the information is publicly available in the college website, it is legally sustainable. The Chatbot has the potential to be developed as an application for Android and iOS operating systems and can be incorporated along with the college's web page.

5. CONCLUSIONS AND FUTURE WORK

The detailed architecture and associated algorithms for an educational chatbot system that responds to user inquiries about educational institution were included in the study. The report most likely contains details on the operation of the chatbot, its design and implementation, and the algorithms used to analyse user requests and produce pertinent responses. For those looking for information on educational institution, the study probably offers a thorough review of the educational chatbot system and its capabilities.

Hence, an educational chatbot is developed that obtains an input query from user. The query is transformed into keywords using NLP techniques. The keywords of the query are matched with the content in the website that are extracted through web-scraping using similarity measures. The document having the higher similarity measure is retrieved and displayed to the user as the response.

Certain potential improvements can be made in this educational chatbot.

As part of our future work, we wish to integrate the application in the ssn.edu.in website so that it would be easy for the people to get their queries solved. And we would like to get more information about the college, so that the user is left with very minimum or no unsatisfied answers. After it's integrated as an application, we plan on making our chatbot as a hybrid application so that it can also be deployed as an Android and IoS application. We plan on including Voice-based queries as well, where users provide input through their voice and the system generates output in text format.

6. REFERENCES

1. Niranjan Dandekar, SuyogGhodey (2017),“Implementation of a Chatbotusing Natural Language Processing”, 9th International Conference on recent developments in Engineering Science, Humanities and Management, ISBN:978-93-86171-88-7, pp. 1076-1082.
2. Prof. Lavanya. K, Rishabh Shah, SiddhantLahoti (2017), “An Intelligent Chat-bot using Natural Language Processing”, International Journal of Engineering Research, Vol.6, No.5, pp. 281-286.
3. Amey Tiwari, Prof.S.M.Patil, Rahul Talekar (2017), “College Information Chat Bot System”, International Journal of Engineering Research and General Science, Vol. 5, No. 2, pp. 131-137.
4. AayushDoshi, Deep Vakharia, GrinalTuscano, Meet Popat, YashrajRai (2022), “Alexis : A VoicebasedChatbot using Natural Language Processing”, International Journal of Engineering Research and Technology, Vol. 11, No.04, ISSN: 2278-0181, pp. 290-294.

5. BoudhirAnouarAbdelhakim, Mohammed Benhmed, SoufyaneAyanouz (2020), "A Smart Chatbot Architecture based NLP and Machine Learning for Health Care Assistance", NISS2020: Proceedings of the 3rd International Conference on Networking, Information Systems and Security, pp. 1-2.
6. Susmitha Mary, SweetySahani (2022),"Chatbot Using Python", International Journal for Research in Applied Science and Engineering Technology (IJRASET), Vol.10, No.V, ISSN: 2321-9653, pp. 3565-3568.
7. Bharathi B, Bhuvana J, Sneha G, Sneha V, Uma M (2019),"Formation of SQL from Natural Language Query using NLP", International Conference on Computational Intelligence in Data Science (ICCIDS), pp. 1-5.
8. Ms. J. Dharshani, Ms. K. Divyabharathi, Dr.M.Senthil Kumar, Ms. S. Sneha (2022), "An Automated Chatbot for an Educational Institution using Natural Language Processing", International Journal of Creative Research Thoughts - IJCRT, Vol.10, No.5, ISSN: 2320-2882, pp. 142-147.
9. Ashish Pal, ShashankBhalotia, Shreya Bisen, TarunLalwani, VasundharaRathod (2018),"Implementation of a Chatbot System using AI and NLP", International Journal of Innovative Research in Computer Science and Technology (IJIRCST) , Vol.6, No.3, ISSN: 2347-5552, pp. 26-30.
10. Prof.K.Bala, Mukesh Kumar, SahilPandita, SayaliHulawale (2017), "Chat-Bot For College Management System Using A.I", International Research Journal of Engineering and Technology (IRJET), Vol.4, No.11, pp. 2030-2033.
11. AishwaryaBhoge, AnupJawade, Papiya Mahajan, PragatiDange, RinkuWankhade (2020), "Healthcare Chatbot using Natural Language Processing", International Research Journal of Engineering and Technology (IRJET) , Vol.7, No.11, pp. 1715-1720.
12. Prof. ManishaSonawane, ShubhamKasar, SiddheshPednekar, Siddhi Suvare (2021), "Chatbot for College Information System", Journal of Emerging Technologies and Innovative Research (JETIR) , Vol.8, No.5, pp. 898-905.
13. Aayush Shah, DarshilGada, Mohan Sharma, Riddhish Shah, Yash Mehta (2017), "The College Chatbot", International Journal of Computer Applications, Vol.173 ,No.7, pp. 0975-8887.
14. Emanuela Haller, TraianRebedea (2013), "Designing a Chat-bot that Simulates an Historical Figure", 19th International Conference on Control Systems and Computer Science, pp. 582-589.
15. Ajay Krishna Palakurthi , Ashok Kumar K, HrushikeshKoundinya K, VaishnaviPutnala (2020),"Smart college Chatbot using ML and Python", IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), ISBN 978-1-7281-6202-7, pp. 1-5.
16. Aafiya Shaikh, Dipti More, RuchikaPuttoo, SayliShrivastav, Swati Shinde (2019), "A Survey Paper on Chatbots", International Research Journal of Engineering and Technology (IRJET), Vol.6, No.4, pp. 1786-1789.