A MINI PROJECT REPORT ON

AUTOMATED CHATBOT FOR COLLEGE APPLICATION USING AI & ML

in the partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

Submitted by

MOHAMMED NADEEM ISRAR	**19B81A3324**
NAHEEDA AFREEN	**19B81A3325**
BONTHALA VENKATESH	**19B81A3356**

Under the guidance of

Mrs. A. Chitty

Assistant professor - CSIT

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
CVR COLLEGE OF ENGINEERING
(*An Autonomous institution, NBA, NAAC Accredited and Affiliated to JNTUH, Hyderabad*)
Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),
Rangareddy (D), Telangana- 501 510

DECEMBER 2022

CERTIFICATE

This is to certify that the minor project report entitled “**AUTOMATED CHATBOT FOR COLLEGE APPLICATION USING AI & ML**” bonafide record of work carried out by **MOHAMMED NADEEM ISRAR 19B81A3324, NAHEEDA AFREEN 19B81A3325** and **BONTHALA VENKATESH 19B81A3356** submitted to **Mrs. A. Chitty** for the requirement of the award of **Bachelor of Technology** in **Computer Science and Information Technology** to the CVR College of Engineering , affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad during the year 2022-2023.

Project Guide	**Head of the Department**
Mrs. A. Chitty	**Dr. Lakshmi H N**
Assistant professor	Professor & HOD ET
Department of CSIT	
Project Coordinator	**External Examiner**

DECLARATION

We hereby declare that the project report entitled **“AUTOMATED CHATBOT FOR COLLEGE APPLICATION USING AI & ML”** is an original work done and submitted to CSIT Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad in partial fulfilment for the requirement of the award of Bachelor of Technology in Computer Science and Information Technology and it is a record of bonafide project work carried out by us under the guidance of **Mrs. A. Chitty**, Assistant professor, Department of Computer Science and Information Technology.

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma in this Institute or any other Institute or University.

Signature of the Student
MOHAMMED NADEEM ISRAR

Signature of the Student
NAHEEDA AFREEN

Signature of the Student
BONTHALA VENKATESH

ACKNOWLEDGEMENT

The success and outcome of this project require a lot of guidance and assistance from many people, and we are extremely privileged to have got this all along with the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank our internal guide **Mrs. A. Chitty**, for providing us an opportunity to do this project work in CVR College of Engineering and giving us all the support and guidance, which made us complete the project duly. We are extremely thankful to her for providing such a nice support and guidance, although she had a busy schedule managing the official affairs.

We would like to thank the Head of the Department, Professor **Dr. Lakshmi H N** for her meticulous care and cooperation throughout the project work. We thank **Dr R. Raja**, Project Coordinator for providing us an opportunity to do this project and extending good support and guidance.

We are thankful for and fortunate enough to get constant encouragement, support, and guidance from all **Teaching staff of CSIT Department** which helped us in successfully completing our project work. Also, we would like to extend our sincere esteem to all staff in the laboratory for their timely support.

ABSTRACT

The Web Application "**AUTOMATED CHATBOT FOR COLLEGE APPLICATION USING AI & ML"** is designed by python using JSON Database structure to retrieve the data. The project is implemented mainly to solve the doubts and queries of both students and parents. A chatbot suits the user needs and requirements. A chatbot is a virtual person who can effectively speak to any human using interactive text skills. These are programs that work on Artificial Intelligence (AI) & Machine Learning (ML) Platform, NLP plays an important role in training the chatbot. The project provides user to have basic interaction with them just like a human. And provides a text-based user interface, allowing user to type commands as well as receive the texts. It can answer the questions asked by the user. The questions could be about college details.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ML:	Machine Learning
NLP:	Natural Language Processing
NLTK:	Natural Language Tool Kit
AIML:	Artificial Intelligence Mark-up Language
XML:	Extensible Markup Language
SQL:	Structured Query Language
IDE:	Integrated Development Environment
GUI:	Graphical User Interface
API:	Application Programming Interface
GPS:	Global Positioning System
JSON:	JavaScript Object Notation
SGD:	Stochastic gradient descent
UML:	Unified Modeling Language

CHAPTER 1
INTRODUCTION

Chatbots are software applications used to conduct online chat conversations via text or text-to-speech rather than direct contact with a real human agent. A chatbot must convincingly simulate how a person behaves as a conversational partner. You can build bots using languages such as AIML (Artificial Intelligence Mark-up Language), an XML-based language that allows developers to write rules that the bot need to be followed[13]. There are two categories of chatbots. One Category is command-based chatbots where the chatbot relies on a database of answers to generate a response. The user has to be very specific when asking questions for the bot to answer. Therefore, these bots can only answer a limited number of questions and cannot perform any functions outside the code. Another category is chatbots powered by AI or machine learning Algorithms allow these bots to answer obscure questions that users shouldn't have to Be specific when asking questions. These bots therefore create responses to user requests Using natural language processing (NLP)[1].

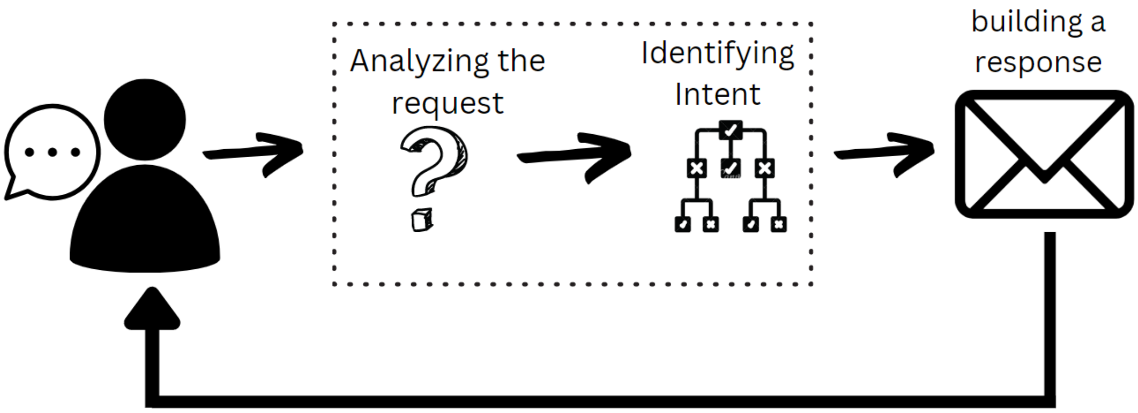


Figure 1: How a chatbot works

Figure 1 shows how a chatbot works. Every time the user asks a question, the bot then First analyzes the request, then identify the intent and entity, builds a response, and send the response to the user. Here intent means the intention of the query and entity means details of the query. For example, if a student wants to know the office hours of the college, the intent in this case is office hours and the entity is the college.

1.1 MOTIVATION

AI-powered chatbots are motivated by the need for traditional websites to provide a Chat features that require a bot to chat with users and resolve queries. When Live agents can only perform two-three operations at a time, but chatbots work without an upper limit which really increases the operations. Also, if there is a school or business If you have a lot of requests, a chatbot on your website will ease the burden of support team. Chatbots significantly improve response rates compared to human support team. We also found chatbots because millennials prefer live chat over phone calls.

It offers a highly engaging interactive marketing platform. again, Chatbots can automate repetitive tasks. company or the school receives the same requests many times a day. The support team should respond to Each query repeatedly. After all, the most important advantage of chatbot is They are Available 24/7. At any time, the user can resolve the request. all these Advantages of chatbots are motivations for implementing college Enquiry chatbot. Before implementing the college Enquiry chatbot, there were various existing chatbots Verified as Amazon Alexa, Google Assistant, Hey Siri and Bixby. To understand the requirement of a chatbot, A sample Amazon shopping app. In this app, customers purchase items There is no information on how to return the product. To get this information, Customers must call and wait long hours to speak to a customer representative. But This complete process is cumbersome for the customer[17]. Therefore, Amazon created a chatbot to answer simple customer request.

Similarly, the College Enquiry Chatbot is designed to help students solve their questions with the click of a button. The Questions are answered at the touch of a button. The main drawbacks we found when using the existing chatbot is a lack of personality and conversational flow. Another downside we found when researching about chatbots was that the bots Designed to follow a specific route and most likely will not satisfy Anything other than a previously defined script. This means that if they are not part of Predefined script, quite a few bots cannot even understand it even if It's the most basic type of query and makes a repetitive and terrible experience[6]. To resolve this issue, we implemented this approach of identifying the intent of the query before responding to the query to give the best possible answer of the request.

1.2 PROBLEM STATEMENT

In this modern age of technology, people do not want to go to college and waste their time asking for informational tasks. Traditional methods are generally slow. Universities have different departments for that, but they still need chatbots for student information. To get the right answer, you must follow some guidelines and go through each process. It saves a lot of time if the user does not have to manually enter data for the information. It is efficient and timesaving if students can retrieve data with one click.

- College students face several problems with college information.
- Burden on students to get information.
- Lack of information about recently held programs.
- Wasting students' time to find out complete information.
- Need to visit person to person for required information.

1.3 PROJECT OBJECTIVES

- To minimize the time required to solve the queries.
- To simplify communication between user and machine.
- Provide detailed information about college.
- Easy access to information.
- To give response to the user based on queries.

1.3 PROJECT REPORT ORGANISATION

a. Literature Survey: This is the first section of the project where we have discussed the similar existing projects and literature that we have surveyed for the project. We have given examples of the different methods and technologies that have been used for Development of chatbot. The next part of this section states and explains the limitations in these existing systems.

b. Software and Hardware Requirements: This section discusses the different kinds of software used in our project and it contains information about the different functional and non-functional requirements of our project. Furthermore, we have discussed the minimum hardware specifications that are required for the chatbot to work efficiently.

c. Proposed System Design: This section contains a detailed review of the proposed system design of our project. We have included various UML diagrams that include use case diagram, class diagram, sequence diagram and activity diagram along with full-fledged description for every diagram. This section will provide a clear about the architecture, flow, and functionality of our project.

d. Implementation and Testing: This section gives details about the implementation of our project and testing result. Every component of the project is explained clearly. The application is then tested against all the functional requirements. Detailed explanation and screenshots are provided for the same.

e. Conclusion and Future Scope: This section states and explains the final results that we have obtained from our project. We also discuss few additional features that can be added to our project to make it more efficient accurate and have a wider scope.

CHAPTER 2
LITERATURE SURVEY

A chatbot (or chatterbot) is software that interacts with users (humans). A virtual assistant that can answer a series of user questions and provide the best possible answer [7]. In recent years, the use of chatbots has expanded rapidly in various fields such as Healthcare, marketing, education, support systems, major etc.

Companies are developing several chatbots for both industrial solutions and research. The best known are Apple Siri, Microsoft Cortana, Facebook M etc. These are just a few of the most popular systems.

Chatbots were originally designed to entertain and mimic human conversations. This is still the reason for the popularity of chatbot development, but since the popularity with the technology has gone up, so has the different uses. The chatbot technology is used for a variety of purposes, including getting information, answering questions, helping with fact-based decision-making, shopping assistants, museum guides, language partners, and education. Especially in a world where tech-savvy students rely heavily on social media and instant messaging platforms like Slack and Facebook Messenger. Chatbots have the potential to provide students with standardized information on the fly. Using chatbots is possible to adapt the speed at which a student can learn without being too pushy.

2.1 EXISTING SYSTEMS

1. Harshala Gawade, Prachi Vishe, Vedika Patil, Sonali Kolpe[2] a chatbot is designed by them using knowledge in database. The proposed system features an online inquiry and an online chatbot system. Development is done using various programming languages by creating user-friendly graphical interfaces for sending and receiving responses. It makes use of SQL (Structured Query Language) for pattern matching.
2. Ms.Ch. Lavanya Susanna, R. Pratyusha, P. Swathi, P. Rishi Krishna, V. Sai Pradeep[3] created a rule based chatbot in which the user will be provided with a set of categories or questions to be asked and the answers are provided to those questions only.
3. Hrushikesh Koundinya K, Ajay Krishna Palakurthi, Vaishnavi Putnala, Dr. Ashok Kumar K [4] a chatbot is designed by them using ML and Python. Which is also a rule

based chatbot if the query is matching with the database, then the response will be provided to the user otherwise some predefined response will be provided.

4. Gandhar Khandagale, Meghana Wagh, Pranali Patil, Prof. Satish Kuchiwale [5] created a chatbot which displays a list of options to the user and the user need to input the option number which needs to be answered, the chatbot will provide a link of the college institution on the user’s request.

2.2 LIMITATIONS OF EXISTING WORK

The limitations of existing work are

- The existing chatbots are rule based which provides a set of options and can answer to those only.
- As the institute’s website contains enormous data still some of the bots provide the link of the website which is time consuming to read the entire content to get the basic answer.

CHAPTER 3
SOFTWARE & HARDWARE SPECIFICATIONS

3.1 SOFTWARE REQUIREMENTS

Operating system: windows 10

VISUAL STUDIO: Visual Studio is an **Integrated Development Environment (IDE)** developed by Microsoft to develop GUI (Graphical User Interface), console, Web applications, web apps, mobile apps, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc. It is not a language-specific IDE as you can use this to write code in C#, C++, VB (Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages. It is available for Windows as well as for macOS[8].

PYTHON: Python is an interpreted high-level general-purpose programming language. Its design Philosophy emphasizes code readability with its use of significant indentation[9]. Its Language constructs as well as its object-oriented approach aim to help programmers Write clear, logical code for small and large- scale projects.

Python is dynamically typed, and garbage collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Characteristics of PYTHON:

- Easy Language. Python is an easy language.
- Readable. The Python language is designed to make developers life easy.
- Interpreted Language. Python is an interpreted language.
- Dynamically Typed Language.
- Object-Oriented.
- Popular and Large Community Support
- Open-Source.
- Large Standard Library.

3.2 HARDWARE REQUIREMENTS

Processor:	Intel Core i5.
RAM:	8GB or more.
SSD:	256GB or more.

CHAPTER 4

PROPOSED SYSTEM DESIGN

NATURAL LANGUAGE PROCESSING (NLP):

Natural language processing strives to build machines that understand and respond to text or voice data—and respond with text or speech of their own—in much the same way humans do[10].

What is natural language processing?

Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of AI concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.NLP combines computational linguistics—rule-based modelling of human language—with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to 'understand' its full meaning, complete with the speaker or writer's intent and sentiment. NLP drives computer programs that translate text from one language to another, respond to spoken commands, and summarize large volumes of text rapidly—even in real time. There is a good chance you have interacted with NLP in the form of voice- operated GPS systems, digital assistants, speech-to-text dictation software, customer service chatbots, and other consumer conveniences. But NLP also plays a growing role in enterprise solutions that help streamline business operations, increase employee productivity, and simplify mission-critical business processes.

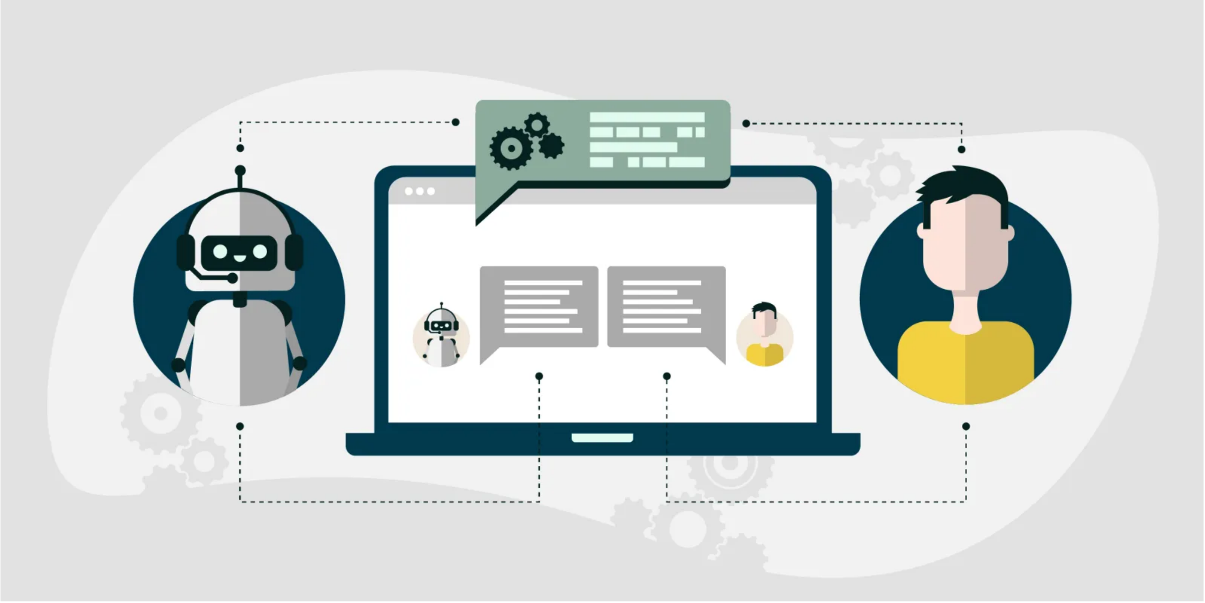


Figure 4.1.1: Natural Language Processing

NLP tasks:

Human language is filled with ambiguities that make it incredibly difficult to write software that accurately figures out the intended meaning of text or voice data. Homonyms, homophones, sarcasm, idioms, metaphors, grammar and usage exceptions, variations in sentence structure—these just a few of the irregularities of human language that take humans years to learn, but that programmers must teach natural language-driven applications to recognize and understand accurately from the start, if those applications are going to be useful. Several NLP tasks break down human text and voice data in ways that help the computer make sense of what it is ingesting. Some of these tasks include the following:

- **Speech recognition**, also called speech-to-text, is the task of reliably converting voice data into text data. Speech recognition is required for any application that follows voice commands or answers spoken questions. What makes speech recognition especially challenging is the way people talk—quickly, slurring words together, with varying emphasis and intonation, in different accents, and often using incorrect grammar.
- **Part of speech tagging**, also called grammatical tagging, is the process of figuring out the part of speech of a particular word or piece of text based on its use and context. Part of speech finds ‘make’ as a verb in ‘I can make a paper plane,’ and as a noun in ‘What make of car do you own?’
- **Word sense disambiguation** is the selection of the meaning of a word with multiple meanings through a process of semantic analysis that decide the word that makes the most sense in the given context. For example, word sense disambiguation helps distinguish the meaning of the verb 'make' in ‘make the grade’ (achieve) vs. ‘make a bet’ (place).
- **Sentiment analysis** attempts to extract subjective qualities—attitudes, emotions, sarcasm, confusion, suspicion—from text.
- **Natural language generation** is sometimes described as the opposite of speech recognition or speech-to-text; it is the task of putting structured information into human language.

NLP tools and approaches

Python and the Natural Language Toolkit (NLTK):

The Python programming language supplies a wide range of tools and libraries for attacking specific NLP tasks. Many of these are found in the Natural Language Toolkit, or NLTK, an open- source collection of libraries, programs, and education resources for building NLP programs.

The NLTK includes libraries for many of the NLP tasks listed above, plus libraries for subtasks, such as sentence parsing, word segmentation, stemming and lemmatization (methods of trimming words down to their roots), and tokenization (for breaking phrases, sentences, paragraphs, and passages into tokens that help the computer better understand the text). It also includes libraries for implementing capabilities such as semantic reasoning, the ability to reach logical conclusions based on facts extracted from text.

NLP use cases

Natural language processing is the driving force behind machine intelligence in many modern real-world applications. Here are a few examples:

- **Spam detection**: You may not think of spam detection as an NLP solution, but the best spam detection technologies use NLP's text classification capabilities to scan emails for language that often writes down spam or phishing. These indicators can include overuse of financial terms, characteristic bad grammar, threatening language, inappropriate urgency, misspelled company names, and more. Spam detection is one of a handful of NLP problems that experts consider 'mostly solved' (although you may argue that this does not match your email experience).
- **Social media sentiment analysis:** NLP has become an essential business tool for uncovering hidden data insights from social media channels. Sentiment analysis can analyze language used in social media posts, responses, reviews, and more to extract attitudes and emotions in response to products, promotions, and events information companies can use in product designs, advertising campaigns, and more.
- **Text summarization:** Text summarization uses NLP techniques to digest huge volumes of digital text and create summaries and synopses for indexes, research databases, or busy readers who do not have time to read full text. The best text summarization applications use semantic reasoning and natural language generation (NLG) to add useful context and conclusions to summaries.

Bots offer a new way to communicate with your customers. With chatbots, we can capture customer’s attention at just the right moment. Chatbots help businesses better understand consumer issues and take action to address those issues[11]. One operator can serve one customer at a time. Chatbots, on the other hand, can answer thousands of requests. Chatbots operate within a pre-defined framework and rely on a single authoritative source within a catalog of commands to answer questions, reducing the risk of confusion or inconsistency in responses[12].

Before going deeper into the methodology, we need to know the following:

- Neural Network
- Bag-of-Words Model
- Lemmatization

NEURAL NETWORK: This is a deep learning algorithm that resembles the way neurons in the brain process information (hence the name). It is often used to achieve patterns between input features in a dataset and corresponding outputs.

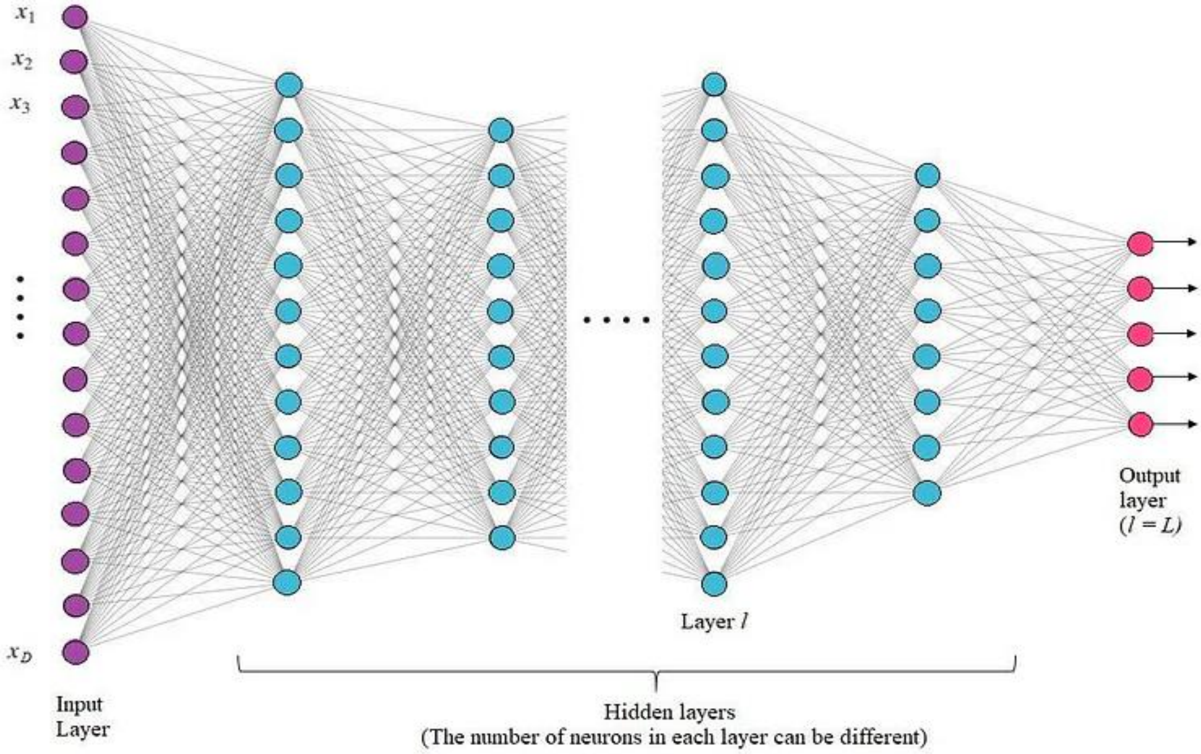


Figure 4.1.2: neural network architecture

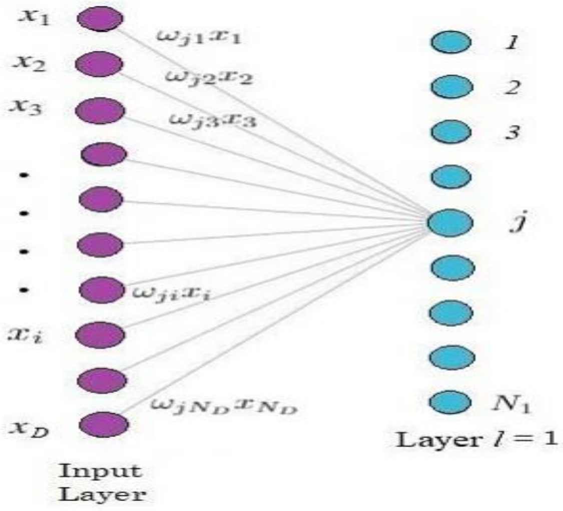
In the above figure purple circles represent the input vector xi, where i = 1, 2, ….. ., D, and are just features of the data set. Blue circles are hidden layer neurons. These are the layers that learn the mathematics required to relate inputs to outputs. Finally, we have the pink circles that make up the output layer. The dimensionality of the output layer depends on the number of different classes used. For example, say you have a 5x4 dataset with 5 input vectors, each with values for of 4 features (A, B, C, D). Suppose you want to classify each row as good or bad and use the number 0 to represents good and 1 represents bad. The neural network then has 4 neurons in the input layer and 2 neurons in the output layer.

Neural Network algorithm involves two steps:

1. Forward Pass through a Feed-Forward Neural Network
2. Backpropagation of Error to train Neural Network

1. Forward Pass through a Feed-Forward Neural Network

This step connects the input layer to the output layer through a series of hidden layers. The first layer of neurons (l=1) receives the weighted sum of the elements of the input vector (xi) along with the bias term b. Each neuron then transforms the weighted sum received on input, a ,using a differentiable nonlinear activation function h(•) to produce output z.



$$\begin{aligned} z_j^{(1)} &= h(a_j^{(1)}) \\ &= h(\sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + b_j^{(1)}) \end{aligned}$$

Figure 4.1.3: Hidden layers of neural network architecture

For subsequent layer neurons, the weighted sum of the outputs of all previous layer neurons is passed as input along with the bias term. The layers of subsequent layers transform the input they receive using activation function.

This process continues until the outputs of the neurons in the last layer (l = L) are evaluated. These neurons in the output layer are responsible for identifying the class to which the input vector belongs. Input vectors are tagged with the class whose corresponding neuron has the highest output value.

Activation function may differ from layer to layer. The two most commonly used activation functions for our Chatbots are the Rectified Linear Unit (ReLu) function and the SoftMax function. The former is used for the hidden layer and the latter for the output layer. A SoftMax function is usually used in the output as it gives a stochastic output.

Rectified Linear Unit (ReLU) function:

The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero[14].The ReLU function is defined as:

$$\boldsymbol{f(x)} = \begin{cases} \mathbf{0}, & \boldsymbol{x} < \mathbf{0} \\ \boldsymbol{x}, & \boldsymbol{x} \geq \mathbf{0} \end{cases}$$

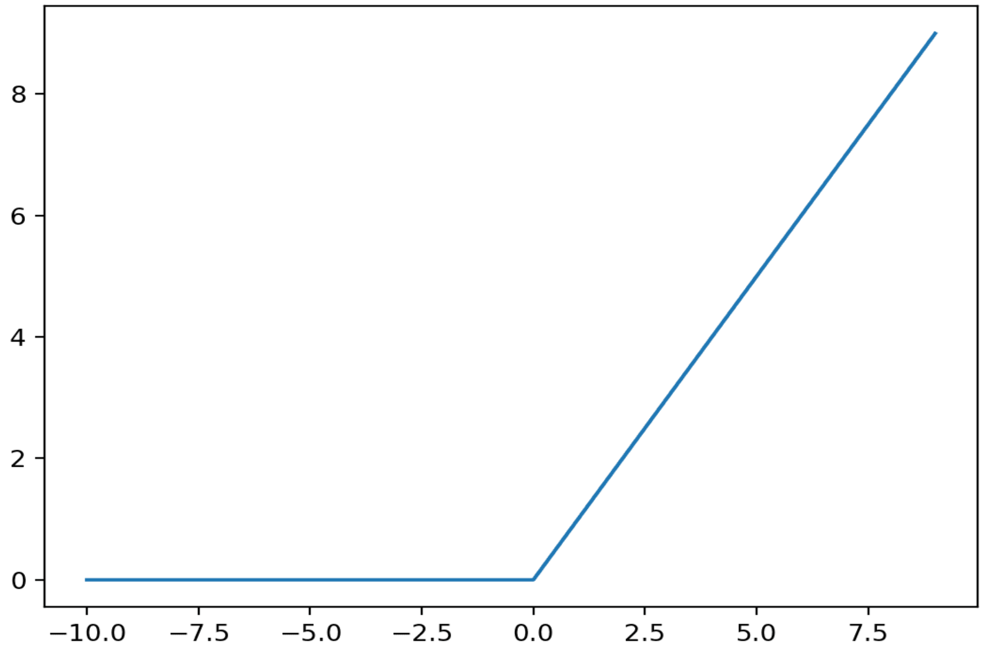


Figure 4.1.4: ReLU Activation Function

Softmax Activation Function:

The softmax function is a function that turns a vector of K real values into a vector of K real values that sum to 1. The input values can be positive, negative, zero, or greater than one, but the softmax transforms them into values between 0 and 1, so that they can be interpreted as probabilities[15]. If one of the inputs is small or negative, the softmax turns it into a small probability, and if an input is large, then it turns it into a large probability, but it will always remain between 0 and 1.

The softmax function is used as the activation function in the output layer of neural network models that predict a multinomial probability distribution. That is, softmax is used as the activation function for multi-class classification problems where class membership is required on more than two class labels.

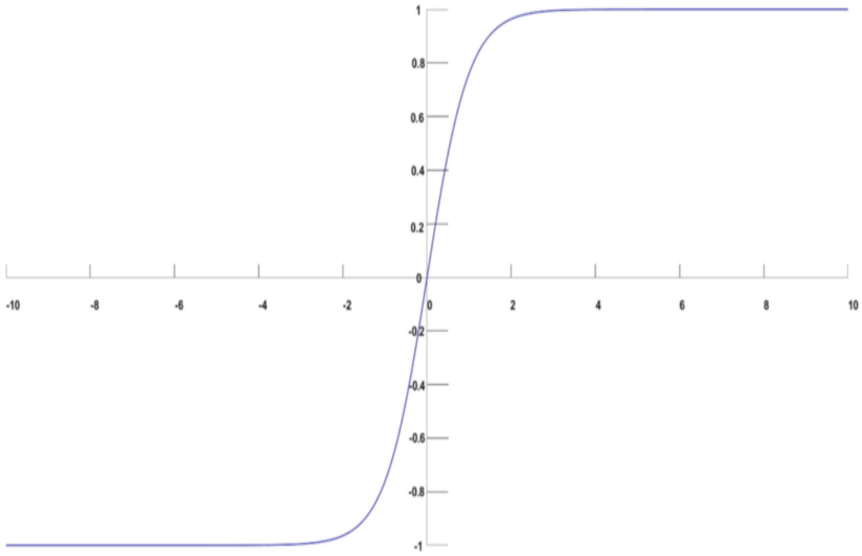


Figure 4.1.5:Softmax Activation Function

2. Backpropagation of Error to train Neural Network

This step is the most important. In this the job of a neural network algorithm is to find the correct set of weights for all layers that give the correct output, and all this step is to find the correct weights and biases. Imagine an input vector passed to the network and know that it belongs to class A. Suppose the output layer gives the highest value of class B. Therefore, our prediction is wrong. Now that we can compute only the error at the output, we need to propagate that error backwards to learn the correct set of weights and biases.

BAG-OF-WORDS (BOW) MODEL: As the computer can understand only numbers, if we want to apply any neural network algorithm on the text first we need to convert the text into numbers. And one way to achieve this is using the Bag-of-words (BoW) model. It is one of the most common models used to represent text through numbers so that machine learning algorithms can be applied on it.

To understand this, consider 5 sentences.
“Hi! I am Rakesh.”

“Hello! Kiran this side.”

“Hi! My name is Kriti.”

“Hey! I am Leena.”

“Hello everyone! Myself Srishti.”

Now, these sentences constitute our input dataset. For the BoW Model, we first need to create vocabulary for our dataset, that is we need to find the unique words from the sentences.in this case, the vocabulary would look like:
hi, I, am, Rakesh, hello, Kiran, this, side, my, name, is, Kriti, hey, Leena, everyone, myself, Srishti.
After this, we need to represent the sentences of our dataset using vocabulary and its size.in our example, we have 17 words, so we represent each sentence using 17 numbers. We will mark “1” if the word is present in the vocabulary otherwise, we will ark as 0 to represent that the word is absent.
“Hi! I am Rakesh.”: 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0

“Hello! Kiran this side.”: 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0

“Hi! My name is Kriti.” 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0

“Hey! I am Leena.” 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0

“Hello everyone! Myself Srishti.” 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1

We will not consider punctuations while converting our text into numbers. This is actually because they are not of that much significance when considering large dataset. Therefore, we need to preprocess the text before using the bag-of-words model. Basic steps include converting all text to lowercase, removing punctuation, correcting misspelled words, and removing helping verbs.

LEMMATIZATION: Most of us, when looking up a word in a dictionary, have noticed that it contains the root form of the word. For example, if you search for the word "hibernating",the dictionary shows it as "hibernate". Similarly, to save computational power, the words in the text must be reduced to their root form, a process called lemmatization.

For example

Partying-party

Eats-eat

Studying-study

4.1 PROPOSED METHODS

[16]The Problem can be classified into the below steps:

i. **IMPORT AND LOAD THE DATA FILE:**In the first step we will import the relevant libraries, and load the data file. The data file is in JSON format which consists of an object called intents. Each intent having its own tag, patterns, and responses. patterns are similar to the interactive statements that we expect from our users whereas responses are the replies to those statements. And, a tag is a one-word summary of the user’s query.to load this json file into python We will use json.loads() and store the content of the file into a variable.

ii. **PREPROCESS DATA:** When working with text data, it is necessary to perform various preprocessing operations on the data before building a machine learning or deep learning model. Based on the requirements, you need to apply different operations to preprocess the data. Tokenization is the most basic and the first thing you can do with text data. Tokenization breaks all text into smaller word-like pieces.Here we iterate over the pattern, tokenize the sentence using the nltk.word_tokenize() function, and add each word to the word list. Also create a list of classes for the tag. Then lemmatize each word and remove duplicate words from the list. Lemmatization is the process of converting a word into its lemma form and then creating a pickle file to store the Python objects used in making predictions.

iii. **MAKING THE DATA MACHINE-FRIENDLY:** In this step, we will convert our text into numbers using the bag-of-words (bow) model. The two lists words and classes act as a vocabulary for patterns and tags respectively. We'll use them to create an array of numbers of size the same as the length of vocabulary lists. The array will have values 1 if the word is present in the pattern/tag being read and 0 if it is absent. The data has thus been converted into numbers and stored in two arrays.

iv. **BUILDING THE NEURAL NETWORK MODEL:** Now we create a neural network using Keras Sequential model. The input to this network will be the array created in the previous step. These would then traverse through the model of 3 different layers with the first having 128 neurons, the second having 64 neurons, and the last layer having the same number of neurons as the length of the classes array. Next, to reach the correct weights, we have chosen the SGD optimizer and defined our error function using the categorical cross-entropy function. And, the metric we have chosen is accuracy. We'll train the python chatbot model about 200 times so that it reaches the desired accuracy.we have also used a Dropout layer which helps in preventing overfitting during training.

v. **PRE-PROCESSING THE USER'S INPUT AND GETTING THE RESPONSE:** In this step we will create a few functions that will convert the user's input query to arrays and predict the relevant tag for it. Our code will then allow the machine to pick one of the responses corresponding to that tag and submit it as output.

The functions are:

a. clean_up_sentence(sentence): This function receives text (string) as an input and then tokenizes it using the nltk.word_tokenize(). Each token is then converted into its root form using a lemmatizer. The output is basically a list of words in their root form.

b. bag_of_words(sentence): This function calls the above function, converts the text into an array using the bag-of-words model using the input vocabulary, and then returns the same array.

c. predict_class(sentence): This function takes sentence as input and returns a list that contains a tag corresponding to the highest probability.

d. get_response (intents_list, intents_json): This function takes in the tag returned by the previous function and uses it to randomly choose a response corresponding to the same tag in intent_json. And, if the intents_list is empty, that is when the probability does not cross the threshold, we will pass the string “Sorry! I don’t understand” as Chatbot’s response.

vi. **DEVELOPMENT OF GRAPHICAL USER INTERFACE:**Here we make use of html to create GUI. We will take the input message from the user and then use the helper functions we have created to get the response from the bot and display it on the GUI.

4.2 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

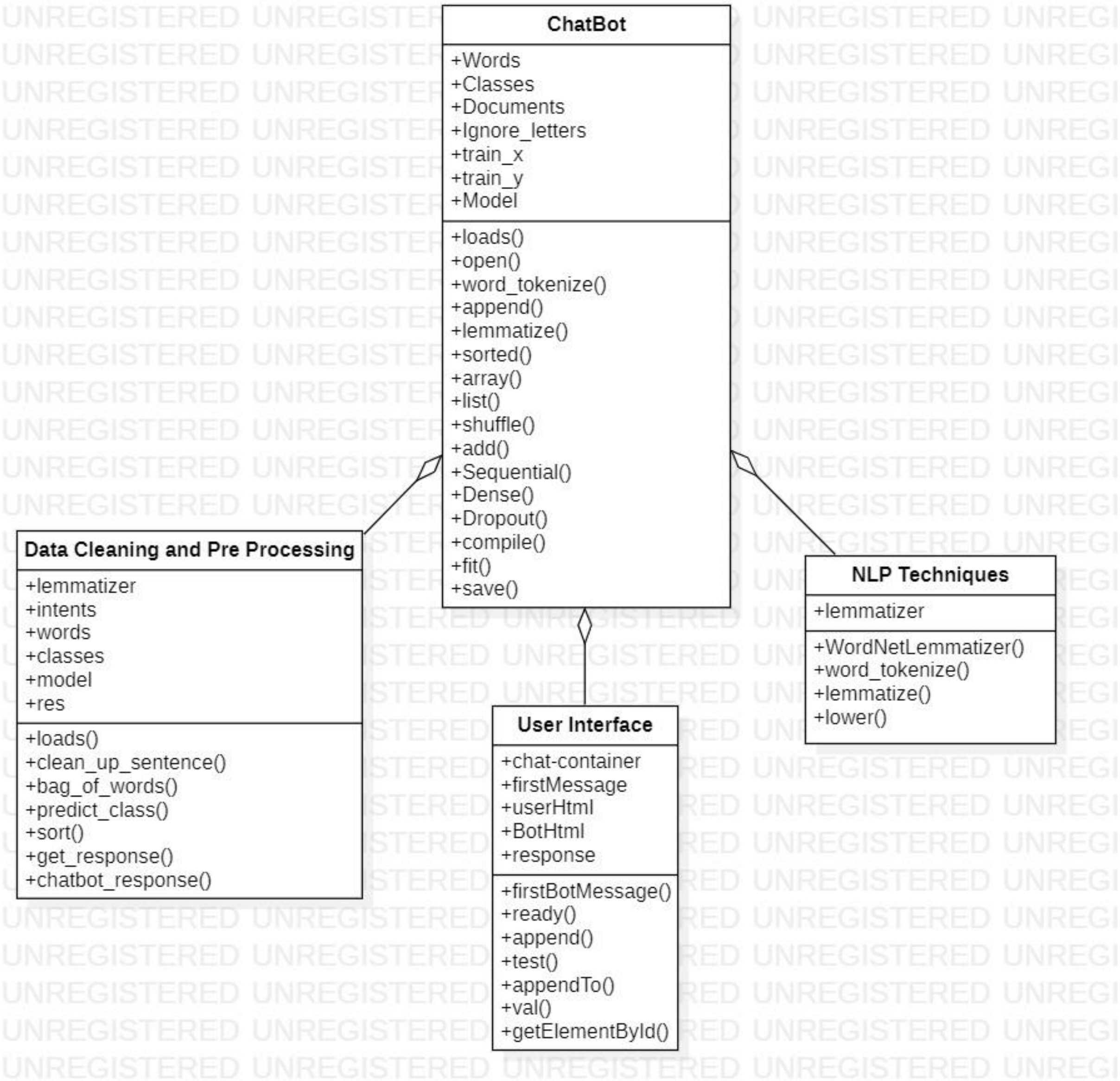


Figure 4.2.1 Class Diagram

Figure 4.2.1 represents the class diagram. There are 4 classes in the system ChatBot , NLP Techniques, Data Cleaning and Preprocessing, User Interface. The Chatbot is responsible for loading the data from intents file and preprocessing it and training the model. Data Cleaning and Preprocessing manages to preprocess the input text and predict the response. NLP Techniques controls the lemmatization of text and finally the user interface is responsible for extracting the input data and displaying the response to the user.

4.3 USECASE DIAGRAM

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

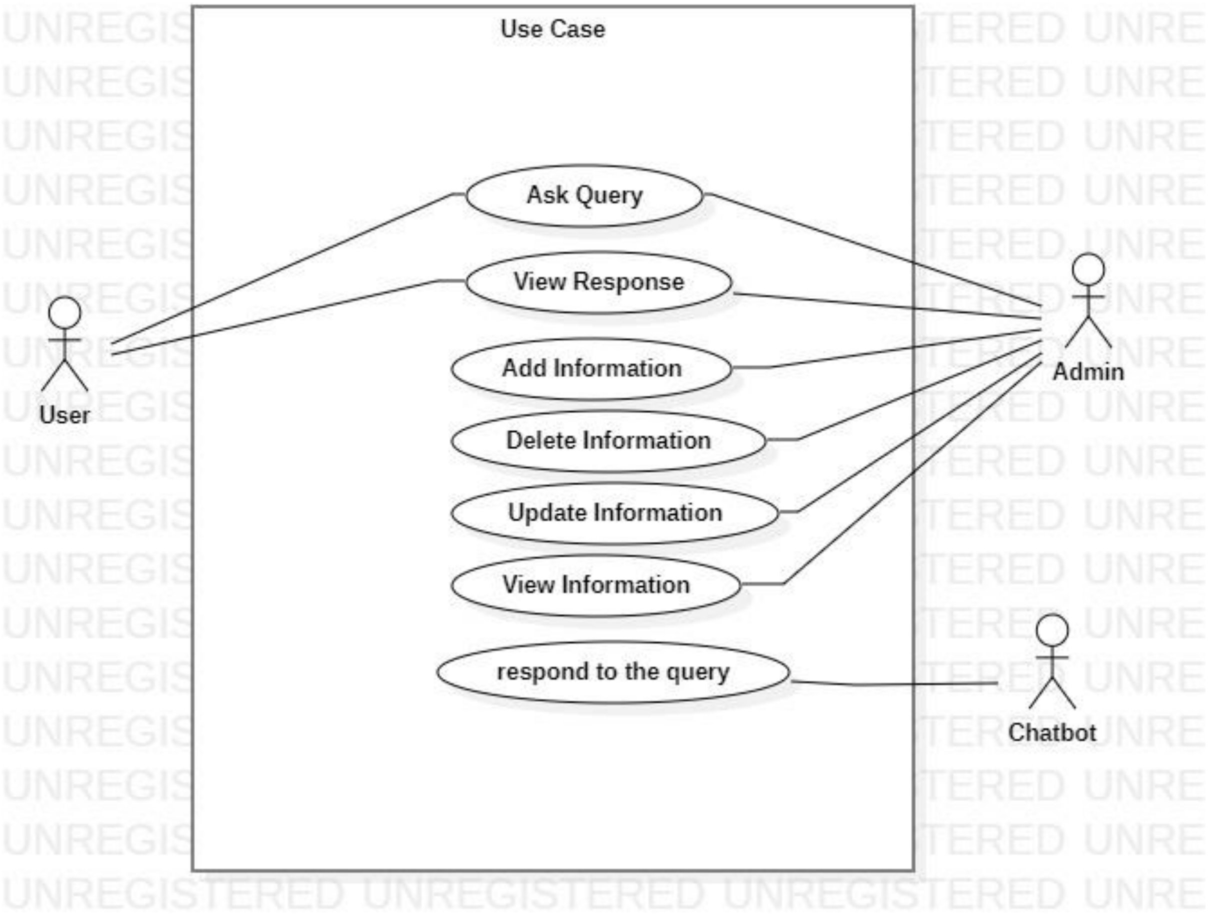


Figure 4.3.1 Use Case Diagram

The above Diagram Represents that there are 3 Actors in the proposed System User, Admin, Chatbot. The User is responsible for Asking the query and view the response and Admin is responsible for Asking the query, viewing response, Add, delete, Update And View Information in the database. Lastly, the chatbot is responsible for Processing the query and responding with the most suitable answer.

4.4 ACTIVITY DIAGRAM

We use Activity Diagrams to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

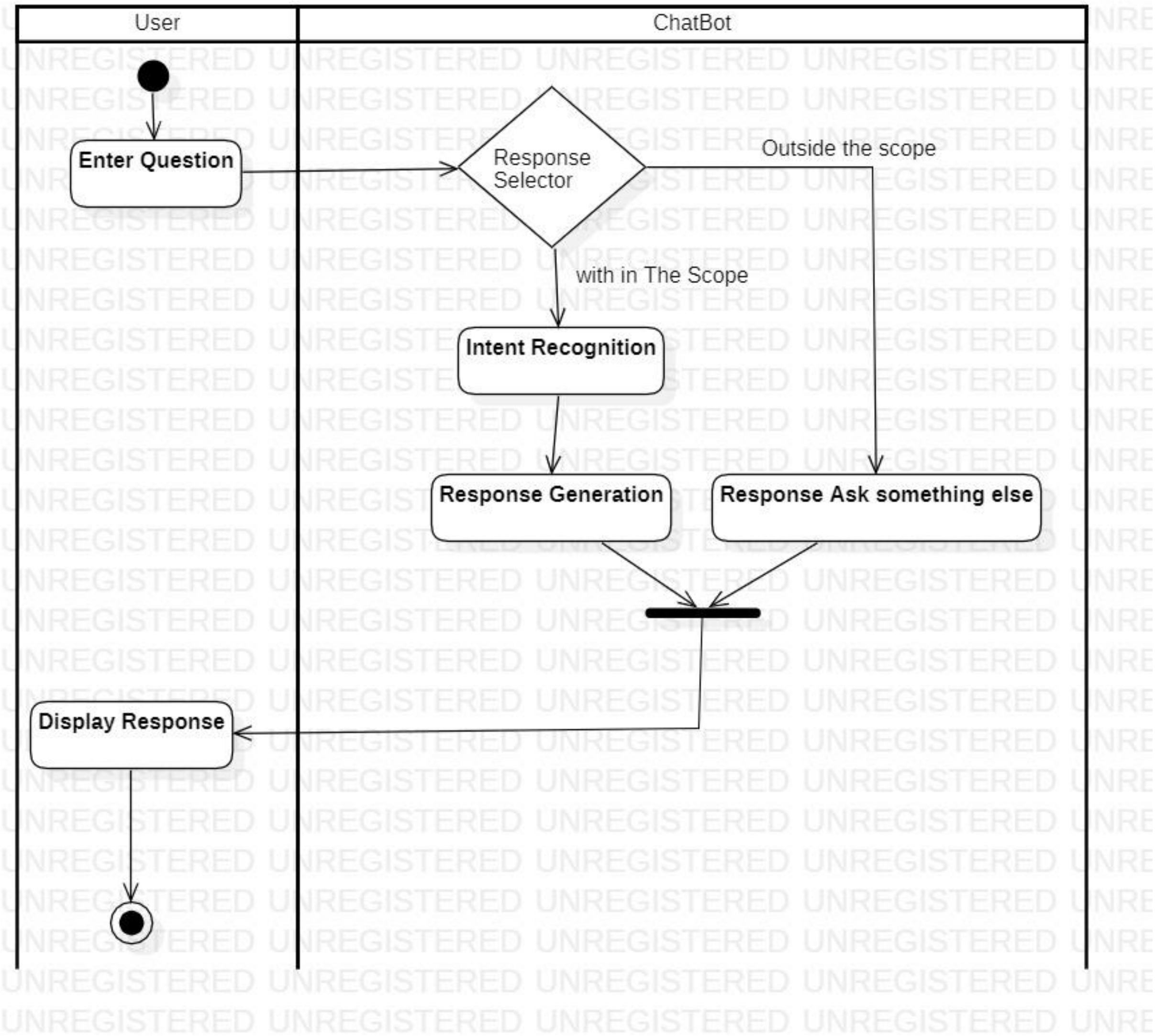


Figure 4.4.1: Activity Diagram

The Above diagram represents activity diagram of the system. There are 2 organizations in the system user and the Chatbot. The user is responsible for the activities enter question and view the response The chatbot manages to Generate the response to the user query.

4.5 SEQUENCE DIAGRAM

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching. A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

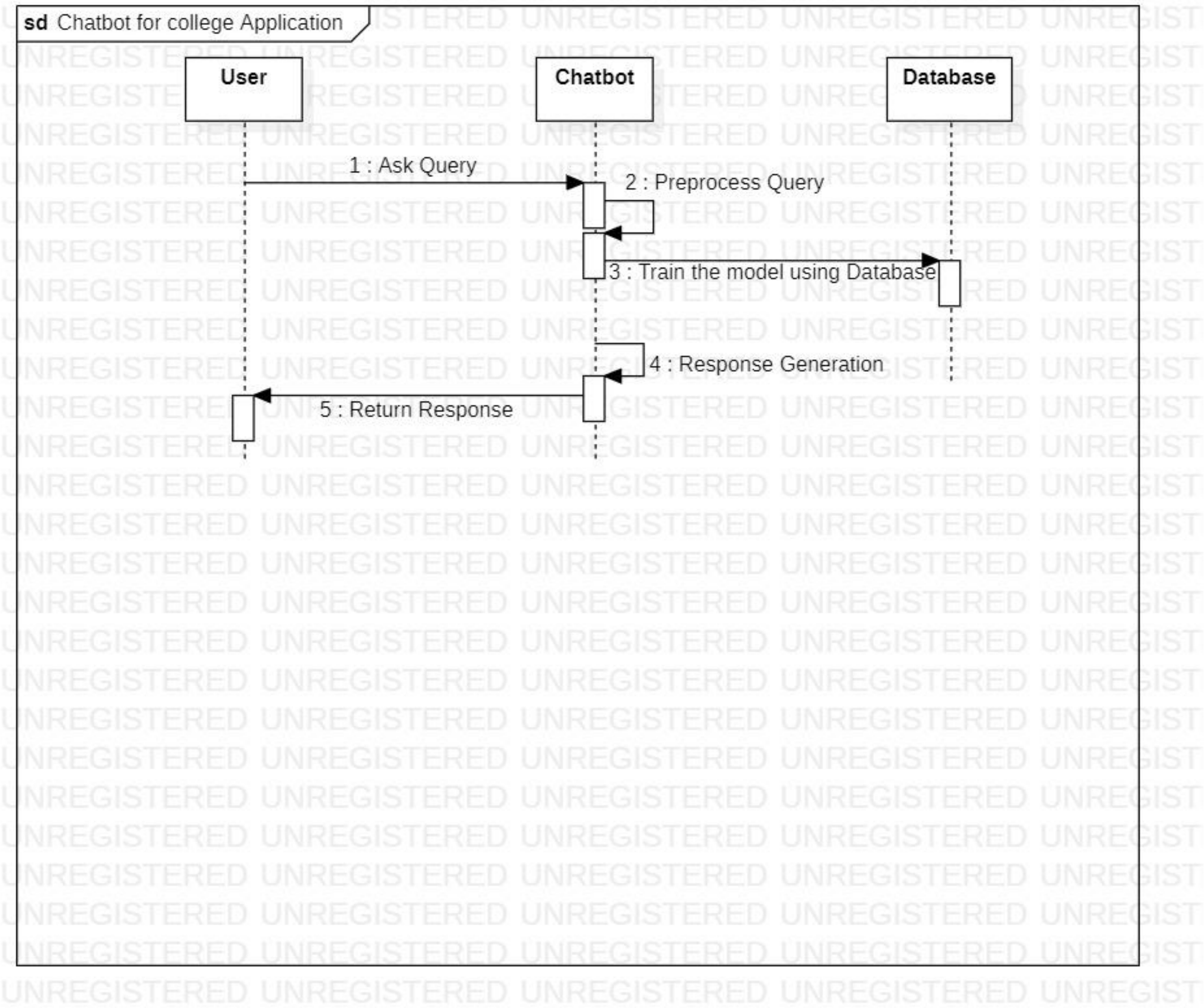


Figure 4.5.1: Sequence Diagram

Figure 4.5.1 Represents the sequence diagram where we have 3 LifeLines User, Chatbot, Database. The user sends the query which is received by the chatbot. Initially, the chatbot pre-processes the data and then trains the model with the help of database. The chatbot then Preprocesses the query and Search for the response in the data base. The chatbot interns Returns the response to the user.

4.6 SYSTEM ARCHITECTURE

This section describes the overall architecture of the proposed system. The main purpose of this chatbot is to respond to user queries without manpower. Users can use chatbots in any Web browser. Whenever the user requests, the chatbot receives the request and analyses it, and respond to users in return. This analysis makes use of various machine learning algorithms. The queries are defined with the Certain tags for each set. This tag is Nothing but keywords to help the chatbot analyze User request. After analysis, the chatbot replies to the user with a required response. When the users request is unclear For chatbot, responses are standard messages defined by Developer. Almost all user questions are clearly answered. Only rare cases are exceptional[1].

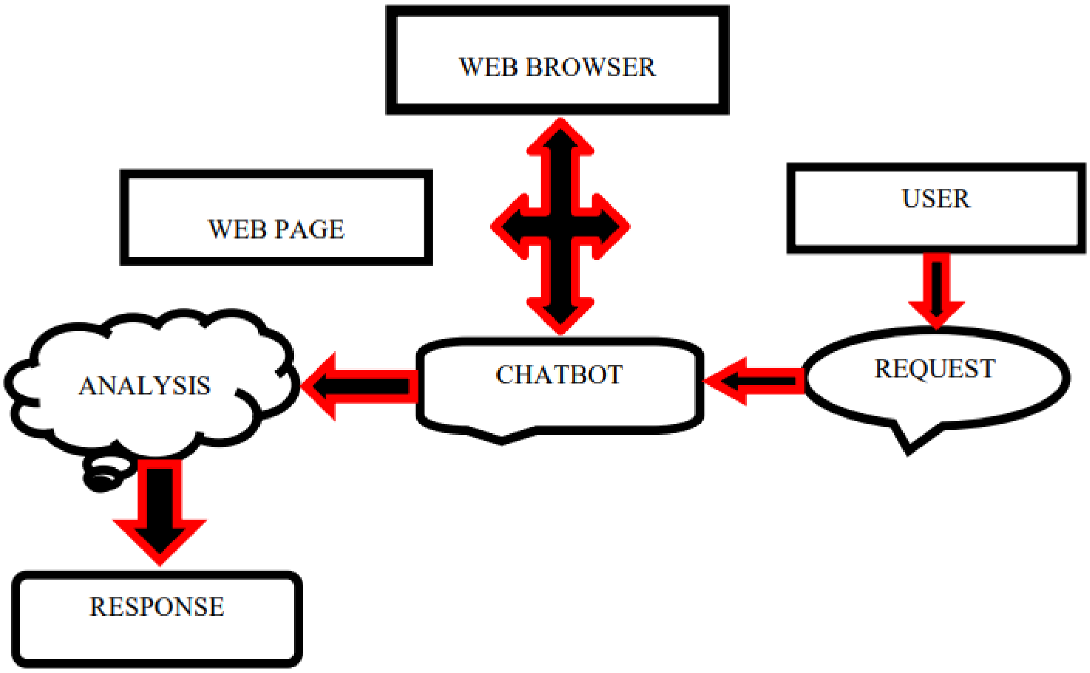


Figure 4.6.1: System Architecture

4.7 TECHNOLOGY DESCRIPTION

1. FRONT END

HTML & CSS

HTML stands for Hyper Text Markup Language, the web's most popular language for developing web pages. In order to provide the user with an easy and responsive User Interface, we have created the web page using HTML to place various elements such as buttons and text fields.

CSS is the language we use to style an HTML document.CSS describes how HTML element Should be displayed. Use CSS to control text color, font style, spacing between paragraphs, column size and layout, background images or colors used, layout design, display variations on different devices and screen sizes, and other effects Host of.

Various HTML widgets we have used in the project are:

- The **<input>** element is the most important form element. It is the tag which specifies an input field where the user can enter query.
- The **<button>** tag defines a clickable button. After entering the query in the input field if the user clicks on the button the query will be sent to the server code to be processed.
- The **<div>** tag is used to group the large section of HTML elements together. Here we place the enter chat container in the div tag. By wrapping a set of elements in a div tag, you can take advantage of CSS styles to apply font styles to all paragraphs at once, rather than coding the same style for each paragraph element.
- The p tag is used to define paragraphs in web pages.

We make use of Various attributes such as id, class to access various elements of the web page.

JAVASCRIPT

jQuery is the most popular JavaScript library used for HTML DOM Manipulation, Event Handling, Animations, and Ajax.A lot of tasks that need to write in many lines of JavaScript code can be called with a single line of jQuery code. That is because jQuery wraps those common tasks into methods. Until the document is "ready", the page cannot be safely manipulated. jQuery detects this ready state for us. Code included inside $(document). ready () will only run once the page Document Object Model (DOM) is ready for JavaScript code to execute. Whenever the button is clicked after inputting the query a POST request is sent to the server along with the data which includes the question that the user asked.On successful request the result is fetched in a variable and displayed in the browser.

2. BACK END

FLASK

Flask is used to develop web applications using Python implemented in Werkzeug and Jinja2. The advantages of using the Flask framework are:

- There is a built-in development server and a fast debugger provided.
- Lightweight
- Secure cookies are supported.
- Request dispatching using REST.
- Support for unit testing is built in.

Steps:

1. We started by importing the Flask class.
2. We then make an instance of this class. It is passed an argument "__name__" which is the name of the application's module or package. Flask needs this to know where to look for resources like templates and static files.
3. The route () decorator is then used to inform Flask which URL should activate our method.
4. This method returns the message that should be shown in the user’s browser.

PYTHON

Python is a famous programming language. It is used for web development (server-side), software program development, mathematics, System scripting, and more.

Features and specifications:

- Easy to learn and use.
- Expressive language
- Interpreted language
- Cross platform
- Free and Open source
- Object-Oriented Language

- Extensible
- Large Standard library
- GUI Programming Support
- Integrated

Python libraries used:

- random
- json
- numpy
- pickle
- nltk
- tensorflow.keras

random: random is a python inbuilt module which is used to generate random numbers. These are pseudo random numbers which are not completely random. It is used to perform random actions such as generating random numbers, printing random numbers etc. In our project we make use of method shuffle from random module which is random.shuffle() to generate a random response from a list of responses after classifying the intent.

json: JSON stands for JavaScript Object Notation. It is a syntax for Storing and exchanging data. From this module we make use of the method loads which is json.loads() to load the data from the text file. This method converts the JSON data into python dictionary.

numpy: numpy stands for numerical python. numpy is a python library to work with arrays. Basically, python have lists which serves the purpose of arrays, but they are slow in processing. NumPy aims to provide array objects that are up to fifty times faster than traditional Python lists. From this module we make use of the method array which is numpy.array() will convert any python array like object into ndarray.

pickle: The Python pickle module is used to serialize and deserialize Python object structures. You can insert any Python object so that it can be saved to disk. Pickle first "serializes" the object before writing it to the file. Pickling is a way to convert a Python object (list, dict, etc.) to a character stream. The idea is that this string contains all the information needed to reconstruct the object in another Python script. from this module we use two methods pickle.dump() and pickle.load(). the pickle.dump() is used to store the object data to the file. To retrieve pickled data, we have to use pickle.load().

nltk: NLTK is a standard Python library with pre-built functions and utilities for ease of use and implementation. It is one of the most widely used libraries in natural language processing and computational linguistics. From this library we import WordNetLemmatizer from nltk.stem for lemmatization of the data. Lemmatization is the process of reducing inflection from words. it reduces words to their origins which have actual meaning.

Tensorflow.keras: Keras is a compact, easy-to-learn, high-level Python library that runs on top of the TensorFlow framework. It focuses on understanding deep learning techniques such as: Create layers of neural networks that hold the concepts and mathematical details of shapes. For generating the neural network model we make use of sequential API that is sequential() from tensorflow.keras.models. which organizes the group of layers in the form of a stack. After creating the instance of a neural network, we need to add layers to it. for that we make use of two methods dense () and dropout (). By using dense () from tensorflow.keras.layers we add neurons to the neural network instance. Similarly using dropout () Is used to remove the neurons from the neural network layer.

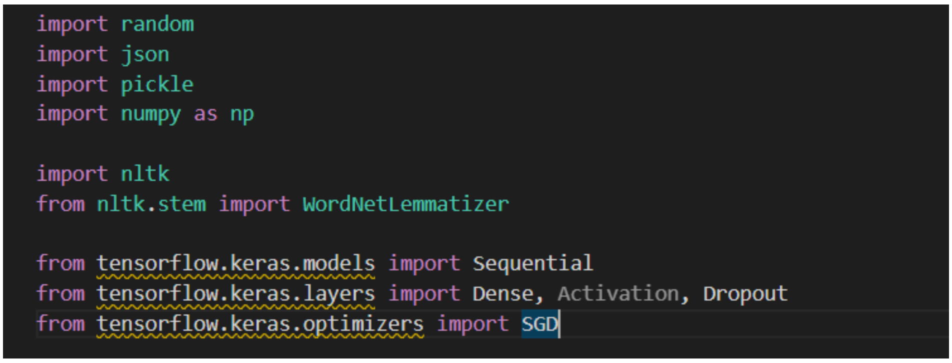
After compiling and fitting the model the model is stored in a .h5 file. Which stores the data in the Hierarchical Data Format 5.

CHAPTER 5

IMPLEMENTATION AND TESTING

This section describes the working of the system on an overall basis and further with specific focus on the software part of the chatterbot and the predefined query data set. An algorithm of the process, proceeded by a design motive of the system is also included.

The coding part is worked with python, HTML, CSS and JavaScript. This includes many library functions like NLTK, TensorFlow, NumPy and few other. These library functions help the chatbot to analyze the user request and decide the response to be given. Python itself has a package for chatbots, which is mainly required for the development of a user-friendly chatbot[1].



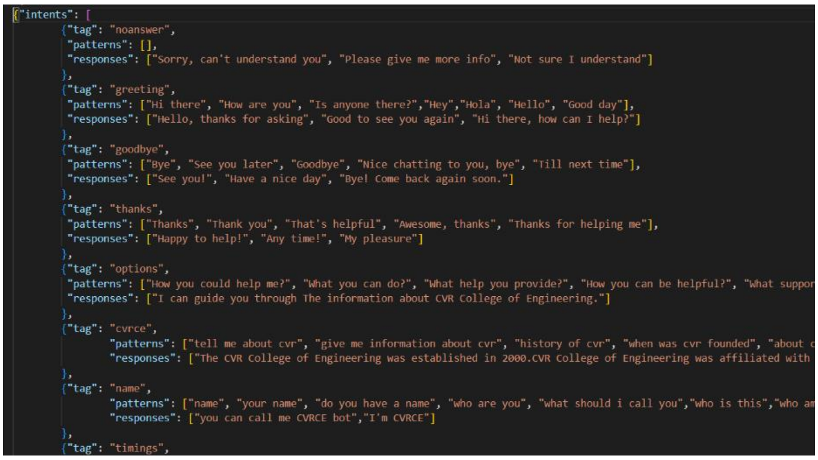
```python
import random
import json
import pickle
import numpy as np

import nltk
from nltk.stem import WordNetLemmatizer

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation, Dropout
from tensorflow.keras.optimizers import SGD
```

Figure 5.1 : Importing Libraries.

Json is a python package that helps the query data set to get parsed by the Python code. This json file has the query with different tags. Each tag has a set of patterns and responses. This tag has default tag with it.



```json
{"intents": [
        {"tag": "noanswer",
         "patterns": [],
         "responses": ["Sorry, can't understand you", "Please give me more info", "Not sure I understand"]
        },
        {"tag": "greeting",
         "patterns": ["Hi there", "How are you", "Is anyone there?","Hey","Hola", "Hello", "Good day"],
         "responses": ["Hello, thanks for asking", "Good to see you again", "Hi there, how can I help?"]
        },
        {"tag": "goodbye",
         "patterns": ["Bye", "See you later", "Goodbye", "Nice chatting to you, bye", "Till next time"],
         "responses": ["See you!", "Have a nice day", "Bye! Come back again soon."]
        },
        {"tag": "thanks",
         "patterns": ["Thanks", "Thank you", "That's helpful", "Awesome, thanks", "Thanks for helping me"],
         "responses": ["Happy to help!", "Any time!", "My pleasure"]
        },
        {"tag": "options",
         "patterns": ["How you could help me?", "What you can do?", "What help you provide?", "How you can be helpful?", "What suppor
         "responses": ["I can guide you through The information about CVR College of Engineering."]
        },
        {"tag": "cvrce",
                "patterns": ["tell me about cvr", "give me information about cvr", "history of cvr", "when was cvr founded", "about c
                "responses": ["The CVR College of Engineering was established in 2000.CVR College of Engineering was affiliated with
        },
        {"tag": "name",
                "patterns": ["name", "your name", "do you have a name", "who are you", "what should i call you","who is this","who am
                "responses": ["you can call me CVRCE bot","I'm CVRCE"]
        },
        {"tag": "timings",
```

Figure 5.2 : Data Set.

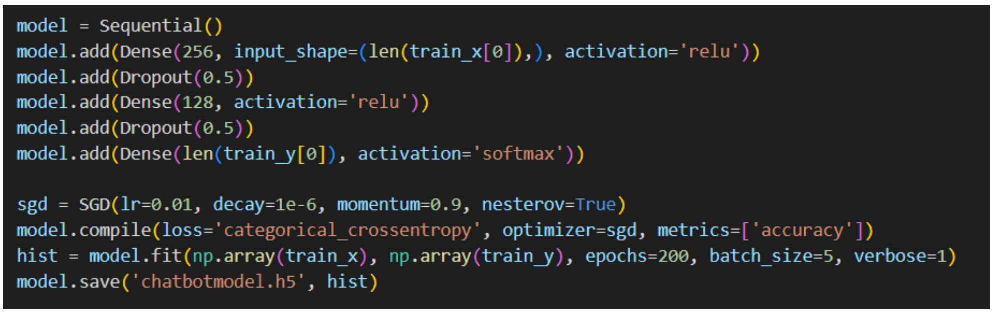
we will extract words from patterns and the corresponding tag to them. This has been achieved by iterating over each pattern using a nested for loop and tokenizing it using nltk.word_tokenize. The words have been stored in “words” and the corresponding tag to it has been stored in “documents”.

For the list words, the punctuations have not been added by using a simple conditional statement and the words have been converted into their root words using NLTK's WordNetLemmatizer(). This is an important step when writing a chatbot in Python as it will save us a lot of time when we will feed these words to our deep learning model. At last, both the lists have been sorted and these functions have been used to remove any duplicates.

Next, we will convert our text into numbers using the bag-of-words (bow) model.

The two lists words and classes act as a vocabulary for patterns and tags respectively. We’ll use them to create an array of numbers of size the same as the length of vocabulary lists. The array will have values 1 if the word is present in the pattern/tag being read and 0 if it is absent. The data has thus been converted into numbers and stored in two arrays: train_x and train_y where the former represents features and the latter represents target variables.

Next, we will create a neural network using Keras Sequential model. The input to this network will be the array train_x created in the previous step. These would then traverse through the model of 3 different layers with the first having 256 neurons, the second having 128 neurons, and the last layer having the same number of neurons as the length of one element of train_y. Next, to reach the correct weights, we have chosen the SGD(Stochastic Gradient Descent) and defined our error function using the categorical cross-entropy function. And, the metric we have chosen is accuracy. We’ll train the python chatbot model about 200 times so that it reaches the desired accuracy.



```
model = Sequential()
model.add(Dense(256, input_shape=(len(train_x[0]),), activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(len(train_y[0]), activation='softmax'))

sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
hist = model.fit(np.array(train_x), np.array(train_y), epochs=200, batch_size=5, verbose=1)
model.save('chatbotmodel.h5', hist)
```

Figure 5.3: Keras Sequential Model

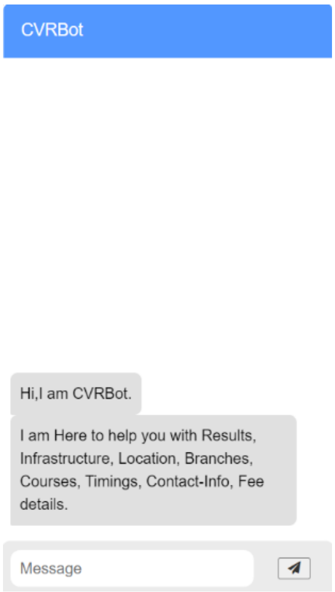
We have used a Dropout layer which helps in preventing overfitting during training. we will create a few easy functions that will convert the user’s input query to arrays and predict the relevant tag for it. Our code will then allow the machine to pick one of the responses corresponding to that tag and submit it as output.

The functions are:

1. clean_up_sentence(sentence): This function receives text (string) as an input and then tokenizes it using the nltk.word_tokenize(). Each token is then converted into its root form using a lemmatizer. The output is basically a list of words in their root form.
2. bag_of_words(sentence): This function calls the above function, converts the text into an array using the bag-of-words model using the input vocabulary, and then returns the same array.
3. predict_class(sentence): This function takes sentence as input and returns a list that contains a tag corresponding to the highest probability.
4. get_response (intents_list, intents_json): This function takes in the tag returned by the previous function and uses it to randomly choose a response corresponding to the same tag in intent_json. And, if the intents_list is empty, that is when the probability does not cross the threshold, we will pass the string “Sorry! I don’t understand” as Chatbot’s response.

We now just take the input from the user and call the previously defined functions.

FRONT PAGE SCREENSHOT:

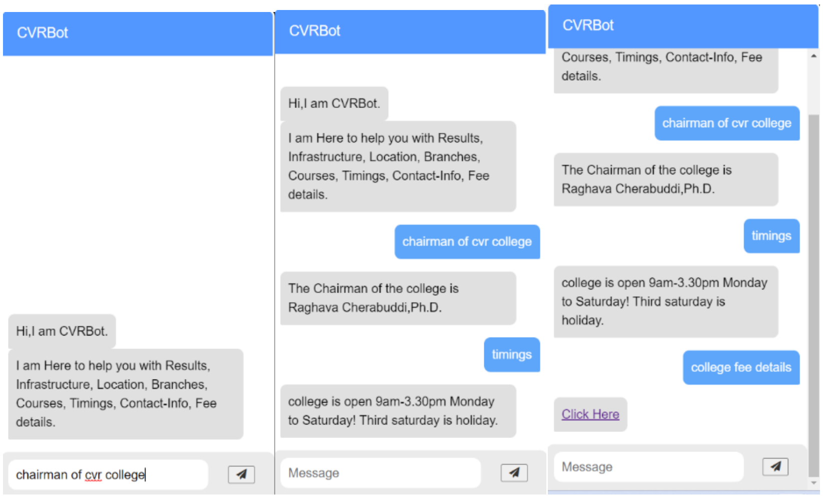


Hi,I am CVRBot.

I am Here to help you with Results, Infrastructure, Location, Branches, Courses, Timings, Contact-Info, Fee details.

Figure 5.4: Front page Screen shot

TESTING TEXT-BASED RESPONSES:



CVRBot

Hi,I am CVRBot.

I am Here to help you with Results, Infrastructure, Location, Branches, Courses, Timings, Contact-Info, Fee details.

chairman of cvr college

CVRBot

Hi,I am CVRBot.

I am Here to help you with Results, Infrastructure, Location, Branches, Courses, Timings, Contact-Info, Fee details.

chairman of cvr college

The Chairman of the college is Raghava Cherabuddi,Ph.D.

timings

college is open 9am-3.30pm Monday to Saturday! Third saturday is holiday.

Message

CVRBot

Courses, Timings, Contact-Info, Fee details.

chairman of cvr college

The Chairman of the college is Raghava Cherabuddi,Ph.D.

timings

college is open 9am-3.30pm Monday to Saturday! Third saturday is holiday.

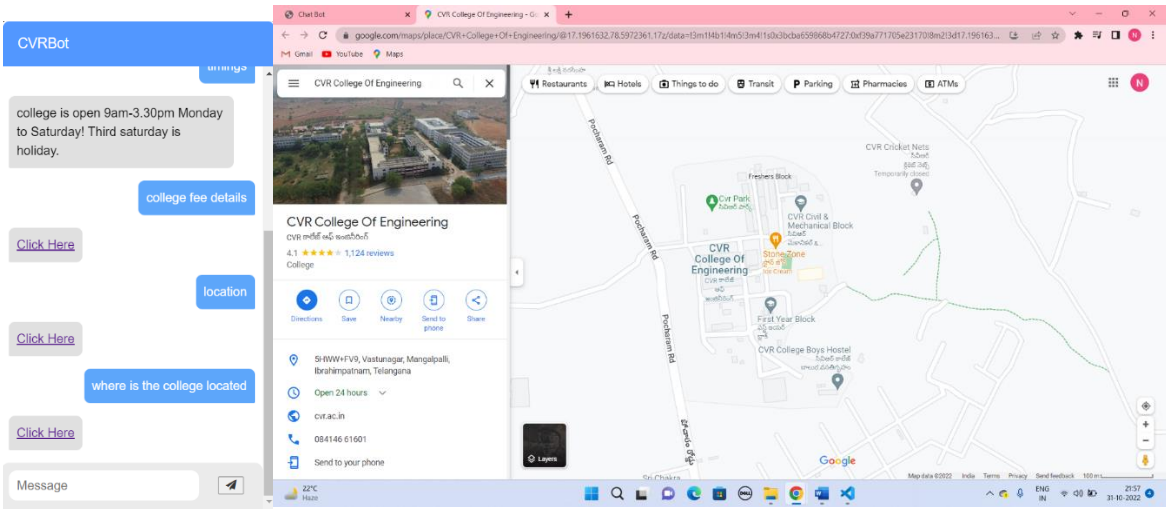
college fee details

Click Here

Message

Figure 5.5: Text based responses

TESTING RESPONSES IN THE FORM OF URL:



CVRBot

college is open 9am-3.30pm Monday to Saturday! Third saturday is holiday.

college fee details

Click Here

location

Click Here

where is the college located

Click Here

Message

google.com/maps/place/CVR+College+Of+Engineering/@17.1961632,78.5972361,17z/data=!3m1!4b1!4m5!3m4!1s0x3bcba659868b4727:0xf39a771705e23170!8m2!3d17.196163...

CVR College Of Engineering

CVR కాలేజ్ ఆఫ్ ఇంజనీరింగ్

4.1 ★★★★ 1,124 reviews

College

5HWW+FV9, Vastunagar, Mangalpalli, Ibrahimpatnam, Telangana

Open 24 hours

cvr.ac.in

084146 61601

Send to your phone

Figure 5.6: Responses in the form of URL

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

The goal of the project is to reduce man-power and to respond to user query at faster rate. Early days, the user’s use to send a query mail to the particular site administrator and it would take few days for the site administrator to reply to the mails. Chatbots can overcome this delay, chatbot satisfies the user request or query immediately with relevant responses. These days many websites of banks, educational institutions, business sectors have developed their chatbots to satisfy user request in a faster time. Chatbots are user-friendly artificial machines.

This project can be developed even more by adding multi languages, speech recognition. We can add many more tags to the data set, as the website gets developed. The chat history of a particular user can be sent as a mail to him/her after the conversation is over. This can be done by authorizing the users and receiving their mail id’s. This project is a small initiative to make the website user-friendly and easily understandable by the user.

REFERENCES:

[1] P. Shanmugapriya, Harini M., “Development of Chatterbot using Python”, International Journal of Computer Applications (0975 – 8887) Volume 176 – No. 21, May 2020,pp 18-20.

[2] Harshala Gawade, Prachi Vishe, Vedika Patil, Sonali Kolpe, “College Enquiry Chat-Bot System”, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181,Vol. 9 Issue 09, September-2020, pp 741-742.

[3] Ms.Ch. Lavanya Susanna, R. Pratyusha, P. Swathi, P. Rishi Krishna, V. Sai Pradeep, “COLLEGE ENQUIRY CHATBOT” , International Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395-0056, p-ISSN: 2395-0072, Volume: 07 Issue: 3 | Mar 2020,pp 784-788.

[4] Hrushikesh Koundinya K, Ajay Krishna Palakurthi, Vaishnavi Putnala, “SMART COLLEGE CHATBOT USING ML AND PYTHON”, 10.1109/ICSCAN49426.2020. 9262426, December 2020.[Online] Available “https://www.researchgate.net/publication/347423315”.

[5] Gandhar Khandagale, Meghana Wagh, Pranali Patil, Prof. Satish Kuchiwale ,“INTELLIGENT CHATBOT FOR COLLEGE ENQUIRY SYSTEM”,International Research Journal of Engineering and Technology, e-ISSN: 2395-0056, p-ISSN: 2395-0072, Volume: 09 Issue: 04 | Apr 2022, pp 2400-2404.

[6] Emanuela Haller, Traian Rebedea, “Designing a Chat-bot that Simulates an Historical Figure” , 2013 19th International Conference on Control Systems and Computer Science,2013, [Online]. Available https://ieeexplore.ieee.org/document/6569323.

[7] Maria Jo˜ao Pereira, Lu´ısa Coheur, “Just.Chat - a platform for processing information to be used in chatbots”,May 2013, [Online] ,Available “https://www.inesc-id.pt/publications/7307/pdf”.

[8] https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022

[9] https://medium.datadriveninvestor.com/python-programming-language-ac762a3b5977?gi=b16d46149c5b

[10] https://www.ibm.com/cloud/learn/natural-language-processing

[11] M. Rahman, A. A. Mamun and A. Islam, "Programming challenges of chatbot: Current and future prospective," 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Dhaka, 2017,pp.75-78. doi: 10.1109/R10-HTC.2017.8288910.

[12] https://www.projectpro.io/article/python-chatbot-project-learn-to-build-a-chatbot-from-scratch/429

[13] B. Setiaji and F. W. Wibowo, "Chatbot Using a Knowledge in Database: Human-to-Machine Conversation Modeling," 2016, 7th International Conference on Intelligent Systems, Modeling and Simulation (ISMS), Bangkok, pp. 72-77.

[14] https://iq.opengenus.org/relu-activation/

[15] https://deepai.org/machine-learning-glossary-and-terms/softmax-layer

[16] https://python.plainenglish.io/create-a-deep-learning-chatbot-with-python-and-flask-d75396a4382a

[17] B. R. Ranoliya, N. Raghuwanshi and S. Singh, "Chatbot for university related FAQs," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi,2017,pp.15251530.doi:10.1109/ICACCI.2017.8126057