

A Project Report on

AI And Web-Based Interactive College Enquiry Chatbot

Submitted in partial fulfillment of the requirements

in

Computer Engineering

by

Rohan Parkar (17102022)

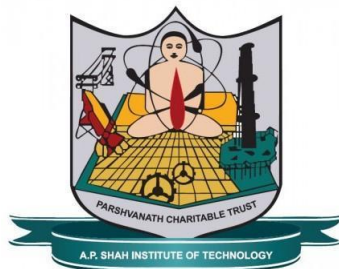
Yash Payare (17102064)

Keyur Mithari (17102014)

Jitesh Nambiar (17102017)

Under the guidance of

Prof. Jaya D. Gupta



Department of Computer Engineering

A.P. Shah Institute of Technology

G.B. Road, Kasarvadavli, Thane(W), Mumbai-400615

UNIVERSITY OF MUMBAI

Academic Year 2020-2021

Approval Sheet

This Project Report entitled “*AI And Web-Based Interactive College Enquiry Chatbot*” Submitted by “*Rohan Parkar*” (17102022), “*Yash Payare*” (17102064), “*Keyur Mithari*” (17102014), “*Jitesh Nambiar*” (17102017) is approved for the partial fulfillment of the requirement in *Computer Engineering* from *University of Mumbai*.

Prof. Jaya D. Gupta
Guide

Prof. S. H. Malave
Head, Computer Engineering Department

Place: A.P.Shah Institute of Technology, Thane
Date: 29th May 2021

CERTIFICATE

This is to certify that the project entitled “*AI And Web-Based Interactive College Enquiry Chatbot*” submitted by “*Rohan Parkar*” (17102022), “*Yash Payare*” (17102064), “*Keyur Mithari*” (17102014), “*Jitesh Nambiar*” (17102017) for the partial fulfilment of the requirement for award of a degree *Bachelor of Engineering in Computer* to the University of Mumbai is a bonafide work carried out during the academic year 2020-2021.

Prof. Jaya D. Gupta
Guide

Prof. S.H.Malave
Head, Computer Engineering Department

Dr. Uttam D.Kolekar
Principal

External Examiner

Place: A.P.Shah Institute of Technology, Thane
Date: 29th May 2021

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Rohan Parkar (17102022)

Yash Payare (17102064)

Keyur Mithari (17102014)

Jitesh Nambiar (17102017)

Date: 29th May 2021

Abstract

AI and web-based interactive College Enquiry Chatbot is a straightforward web application that aims to supply knowledge regarding college. The chatbot created here is a web-based application that uses tongue Processing Libraries and AI terminology to possess conversations with humans. “Eliza” and “Cleverbot” are several online applications that are created within the past. The College Enquiry Chatbot will engage in friendly conversations, respond to the course and college information, provide a link to the tutorial calendar, and answer frequently asked questions, among other things. This project is specialized in creating a chatbot to be employed by students to urge their queries responded to easily from the college website. A chatbot may be a program that may do real conversations with textual and/or auditory methods. Using AI, chatbots can simulate human conversations. Humans respond to others depending on their mood and emotions. Whereas chatbots are bound by some rules, resulting in them treating a customer in the most polite and perfect way. Students can ask questions to the chatbot at any time of the day and get a reply very quickly. At any time of day, chatbots can have simultaneous conversations with thousands of people. A chatbot can work 24x7 without getting tired. It is subjected to minimal errors thus increasing productivity.

Contents

1. Project conception and initiation	1
1.1 Introduction	1
1.2 Objectives	2
1.3 Literature review	3
1.4 Problem Definition	4
1.5 Scope	4
1.6 Technology stack	5
1.7 Benefits for environment and society	5
2. Project Design	6
2.1 Proposed system	6
2.2 Design (flow of modules)	7
2.3 Class diagram	8
2.4 Use-Case diagram	9
2.5 Modules	10
2.5.1 Feedforward Neural Network	10
2.5.2 Natural language processing	11
2.5.3 Dataset	13
3. Implementation	14
3.1 Algorithms	20
3.2 Pseudo code	21
3.3 Platforms for execution	22
4. Results	23
5. Conclusion and future scope	27
6. References	28
7. Bibliography	30
8. Publication	31

List of Figures

2.1 Flow Diagram	7
2.2 Class Diagram	8
2.3 Use-Case Diagram	9
2.4 Working of feed forward neural net	10
2.5 Stemming	11
2.6 Tokenization.	11
2.7 Bag of words	12
2.8 Dataset	13
3.1 Natural Language Processing functions	14
3.2 Neural Network Model	15
3.3 Creating training data and hyper-parameters	16
3.4 Training the model	17
3.5 Prediction of tags	18
3.6 Flask	19
4.1 Training loss graph	23
4.2 Screenshots of conversation with chatbot	24
4.3 FAQ system in chatbot	24
4.4 Suggestion system	25
4.5 Images as reply	25
4.6 Google maps as reply	26

List of Tables

3.1 Natural language processing algorithms	20
3.2 Software's used	22
4.1 Loss values	23

List of Abbreviation

AI : Artificial Intelligence
NLP : Natural Language Processing
GUI : Graphical User Interface
API : Application Programming Interface

1. Project conception and initiation

1.1 Introduction

Usually, we tend to pay our time interacting with distinct chatterboxes on the web, largely targeted at such functions or just entertainment. Chatbots extract relevant entities by analysing and defining the intent of the user's request. The college inquiry chatbot project is meaning to use natural language processing to answer the user's queries. It is extremely beneficial to users because it helps them to ask questions in their native tongue and receive the requested answers quickly. The project takes the user's queries in question-based format and processes them to produce the requested response as a message. It avoids the time-consuming approach of visiting colleges and collecting relevant information to meet the requirements. As students, we require many sorts of data regarding our college and university during our course. Traditionally we have a person's representative who answers all the doubts and questions of the scholars. This process is tedious and repetitive. It can reduce the efficiency of a person's representative. Instead, a chatbot can serve this purpose well. Artificial intelligence (AI) relates to the simulation of human intellect in instruments that are programmed to conceive like humans and imitate their actions. The term may also be applied to any machine that displays characteristics related to a person's mind like learning and problem-solving. AI may be a method of making a machine, a computer-controlled robot, or computer software think intelligently. Chatbots make use of machine learning to accomplish AI helping them to understand the user query and provide an appropriate response [5]. The chatbot developed here is a web-based application that converses with humans using Natural Language Processing Libraries and Machine Learning [3]. Since the curriculum of the college keeps on changing, there has to be a database that can be altered and updated. To serve this purpose a website is created in HTML and PHP to update the database from time to time. To improve efficiency and accountability for college students, the bot uses essential data given by the institute themselves. It reduces the paperwork, manpower for a person. For the User, there is no need to visit the college personally to inquiry about the college-related information. This technique aids the scholars to be updated with college-related activities [7]. This technique is developed aiming at reducing the time for the student, parents, and the faculty at the institution.

1.2 Objectives

1. Available to everyone at any time:

Students or parents can ask their queries to the chatbot at any time of the day and get the desired response. A chatbot will run 24 hours a day, 7 days a week, answering everyone's questions.

2. Reduce time consumption:

It can take a long time to get to college and gather all of the necessary details. Without having to go to college, a chatbot can provide a wealth of knowledge.

3. Cost effective:

There is no more work to be done once a chatbot has been deployed. In order to answer the students' questions, less human power is needed, making it more cost efficient.

4. Simple and easy use:

Parents and students who are unfamiliar with the college website can quickly access details using the chatbot.

5. Keep students updated:

In just a few minutes, students will receive the most recent news and updates about grades, tests, activities, and so on with the help of chatbot.

1.3 Literature review

[1] The paper titled "AI and Web-Based Human-Like Interactive University Chatbot (UNIBOT)," 2019 was published by N. P. Patel, D. R. Parikh, D. A. Patel and R. R. Patel. The project is about the communication between users and chatbots which can be accessed from anywhere anytime. The chatbot can be easily attached to any university or college website with few easy language translations. The chatbot provides different information related to university or college and also students-related information. Anyone with links to the university's website can use the chatbot. The project uses the idea of Artificial Intelligence and Machine Learning. PHP Language is utilized for the development of chatbot. Users may ask university-related questions, and the query is then used as an input to the algorithm, which processes the request and shows the resulting answer to the user. The Project GUI is alike to a Messaging Application.

[2] The paper titled "Chatbot for University Related FAQs", (2017) was published by B. R. Ranoliya, N. Raghuwanshi and S. Singh. Chatbot has become more widespread in business groups immediately as they will decrease customer service costs and handles multiple users at a time. But yet to realize many tasks there's a requirement to form chatbots as efficient as possible. To approach this problem, during this paper we offer the planning of a chatbot, which provides an efficient and accurate account any query supported the dataset of FAQs using AI Markup Language (AIML) and Latent Semantic Analysis (LSA). Template-based and general questions like welcome/ greetings and general questions are going to be skilled using AIML and other service-based questions use LSA to supply responses at any time that will serve user satisfaction. This chatbot are often employed by any University to acknowledge FAQs to inquisitive students in an interactive fashion.

[3] The article named "Recommending Moodle Resources Using Chatbots", (2019) was published by Kamal Souali, Othmane Rahmaoui, Mohammed Ouzzif, Ismail El Haddioui. In a conventional homeroom, at whatever point a student needs assistance or then again looks for extra data, he should consistently contact the teacher for more explanation or ask his kindred neighbors or associates. This test has pushed teachers and analysts to execute novel plans to assist students with improving their learning furthermore, their insight. New arrangements are utilizing Artificial Knowledge (AI) procedures like Machine Learning (ML) furthermore, Natural Language Processing (NLP). Utilizing one of the PHP put-together accessible systems with respect to the web, our chatbot can be coordinated into Moodle's foundation either as a custom menu thing or as a gadget inside an HTML block.

[4] The paper titled "NLP chatbot for Discharge Summaries", (2019) was published by Harsh Lal, Priyanshu Lal. In today's world, there is a growing interest in improving the usability of application interfaces. The human-machine interface as a technology combines various fields, and computational methodologies promote collaboration between users and machines using natural language. Improving the quality of treatment and support provided by the healthcare sector is one of the most significant priorities of healthcare. The use of HCI will benefit both patients and health-care providers. With so much information accessible to patients and clinicians, it can be difficult to find important information in a timely manner, particularly for providers who have busy schedules and patients who have minimal health literacy. We present a Natural Language Processing technology that uses subject modeling and word embeddings to build a chatbot that can answer patients' questions based on their discharge summary and the information representation gained from a corpus of various discharge summaries.

1.4 Problem Definition

The need for college enquiry system arises due to various reasons which include: the slow nature of college website, an outsider would not know where to search for a particular piece of information, difficult for the person outside college's domain to extract information.

The smart solution for all the drawbacks leads to the need of the system. The college enquiry Chatbot which will provide the response by summarizing the query and then output answers, it also provides selective information what the user wants. A college Chatbot will dispense all answers relating to domains such as admission, examination cell, notice board, attendance, placement cell and other miscellaneous domains. The major features of the chat bot are:

1. College admission related queries could be answered through it.
2. Viewing user profiles and retrieves attendance and grade/ pointers.
3. College students can get information about examinations to be held.
4. College students can fetch particulars about placement activities.

College enquiry Chatbot will act as a fast, standard and informative widget to enhance college website's user experience and bestow users with righteous information. The bot will analyze user's queries and understand users' message and then reply accordingly. It uses AI & NLP. This way users' time and efforts will be saved, and s/he will be equipped with effective answers.

1.5 Scope

As students, we require many types of information regarding our college and university during our course. Traditionally we have a human representative who answers all the doubts and questions of the students. This process is tedious and repetitive. It can reduce efficiency of a human representative. Instead, a chatbot can serve this purpose well. Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. Chatbots makes use of machine learning to succeed in AI helping them to know the user query and supply an appropriate response. The chatbot created here is a web-based application which uses Natural Language Processing Libraries and Machine Learning to have conversations with humans. Students can enquire about facilities and query related to exams, academics, fee structure, etc. Students can also ask questions related to placement activities. The result can be showed in the form of images and card format or in text format. The query will be answered based on questions asked and the language model built, and the response media created. The user can use speech recognition feature to ask queries. The chat bot also provides suggestions while typing questions. There is a FAQ button which allows user to directly ask that question to the bot on one click. Thus, the chatbot aids the students to be updated with college related activities and aims at reducing the time for the student, parents as well as the faculty at the institution.

1.6 Technology stack

Software's Used:

1. **Anaconda:** Anaconda is an open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.
2. **Pytorch:** PyTorch is an open-source machine learning library based on the Torch library, used for applications such as computer vision and natural language processing, primarily developed by Facebook's AI Research lab.
3. **Visual Studio Code:** Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.
4. **Flask:** Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

Algorithms:

1. **Natural Language Processing (NLP):** Natural language processing is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, how to program computers to process and analyze large amounts of natural language data.

1.7 Benefits for environment and society

1. Chatbots will continue to operate every day throughout the year without requiring taking a break thus requiring less manpower.
2. Chatbots on the other hand can simultaneously have conversations with thousands of people at any time of the day.
3. Humans react to others based on their mood and emotions. Whereas chatbots are bound by some rules, resulting them to treat a customer in the most polite and perfect way.
4. Chatbots can do repetitive tasks without errors unlike humans. This helps people save time and increase productivity.

2.Project Design

2.1 Proposed system

We created an integrated chatbot for college-related enquiries in this work. A dataset is created using tags, questions and responses accordingly. The dataset is then separated into x train and y train, where x train is the inputs that are processed using NLP and y train are all the tags in the dataset. The hyper-parameters are then defined and the Feedforward neural network model. The model is trained on x train and y train. Once the model is trained, it is saved and ready to answer user queries. In all cases, user discussion begins with greetings or general questions. NLP is used to process user requests first, and this operation is split into four sections.

- Tokenization is used to divide the sentence into words.
- Using porter stemmer, these words are clipped to their source words.
- The punctuation marks are removed.
- A bag of words is formed.

This processed data is then fed into a trained feedforward neural network model. Then it predicts the most appropriate tag. The model checks if the prediction is above 75% and if so, it finds the predicted tag in the dataset, and the user is then given an answer related to that tag. When the prediction is below 75% the bot gives a link to a google form where the user can give feedback of the question whose answer was not found. This paper introduces a chatbot for educational services, where users (students or parents) may ask questions about college enrollment, college information, and other academic-related topics. As previously the chatbots were based on SQL queries it was tough to get accurate responses when the questions were twisted. The use of AI helps the understand the human language and give correct responses even if the questions are twisted. This approach helps to create more interactive conversations with the chatbot as the chatbot is first trained on the data and then deployed to give responses to the users.

2.2 Design (flow of modules)

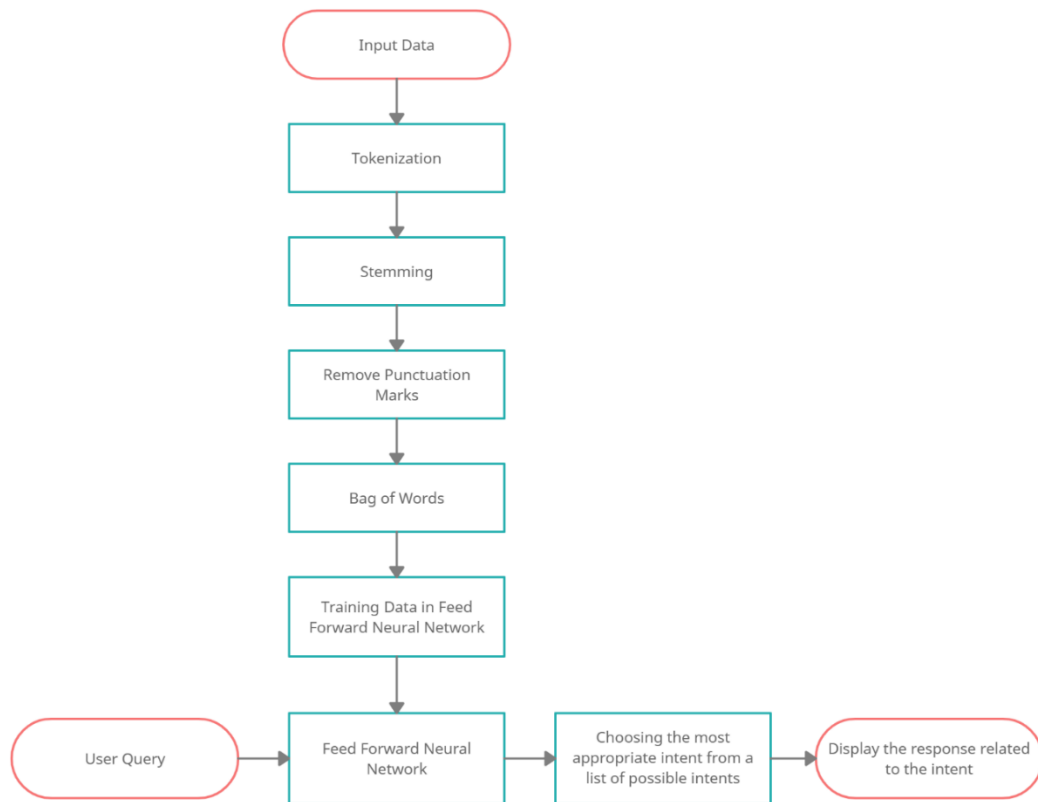


Figure 2.1: Flow Diagram

The flow of the project's various modules is depicted in the diagram above, beginning with the user posing a question and ending with the bot responding to it.

2.3 Class diagram

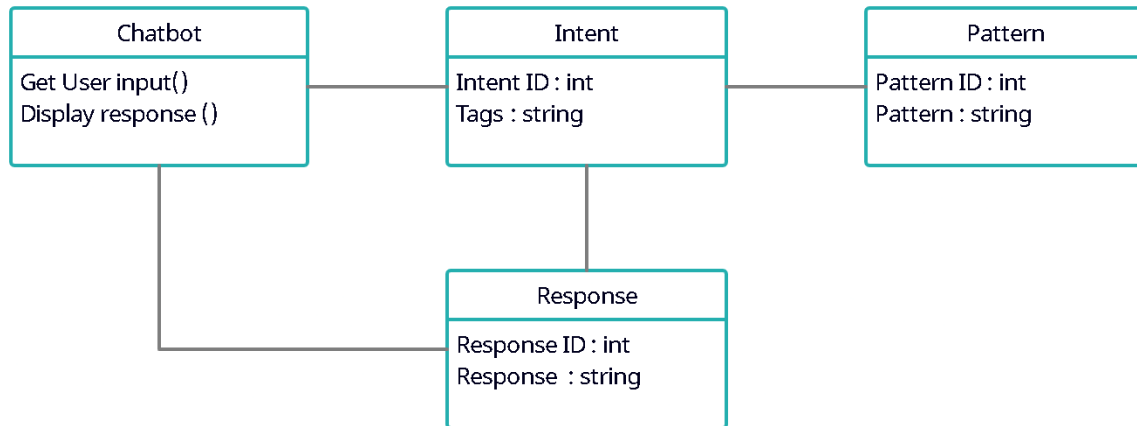


Figure 2.2: Class Diagram

The class diagram for this project is shown above, and it illustrates the project's operation by containing various scripts, their attributes, and methods.

2.4 Use-Case diagram

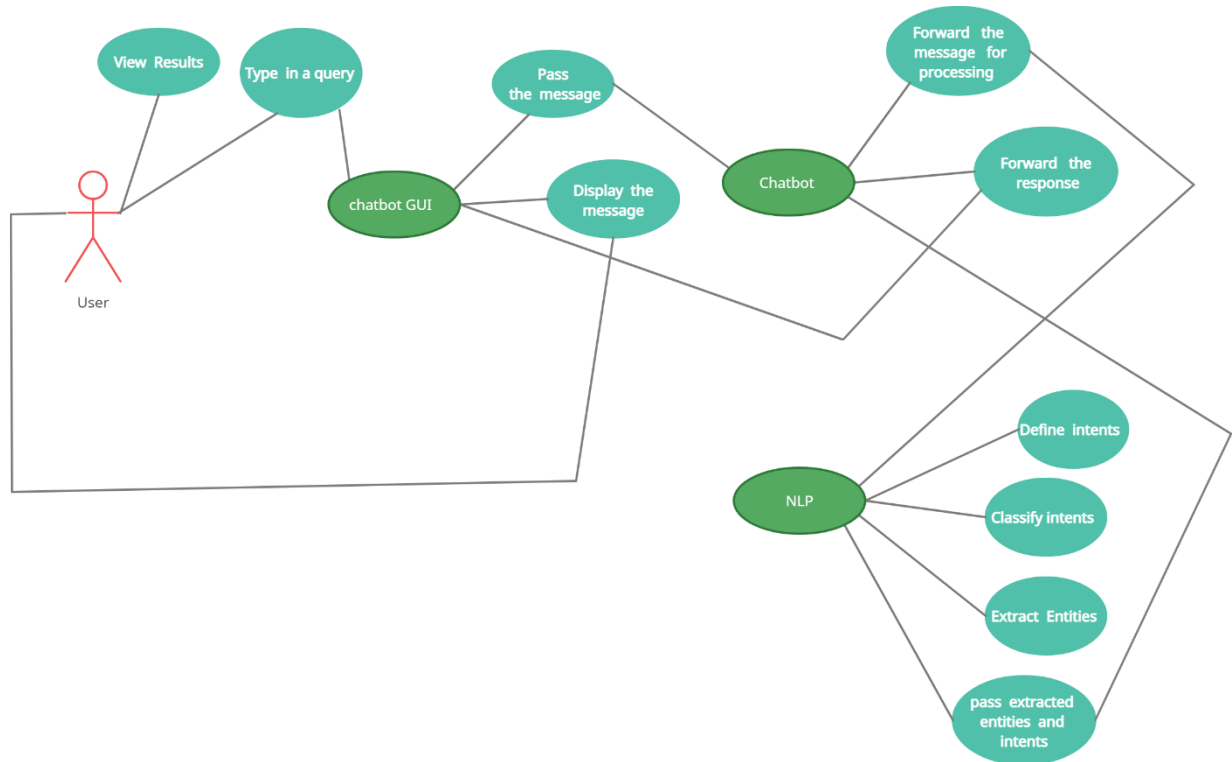


Figure 2.3: Use-Case diagram

The above figure is a use case diagram which is a graphical depiction of a user's possible interactions with the chatbot system.

2.5 Modules

2.5.1 Feedforward Neural Network

The chatbot uses Feedforward Neural Network for predicting the responses. Once the Natural Language Processing is performed on the patterns, X variable is created which has all the training dataset which is fed to the feedforward neural network model and it predicts the Y variable that is tagged. The dataset has around 202 unique words as the input and 54 tags as the output. The hyper-parameters like learning rate are set to 0.00017, the batch size is 8 and the number of epochs is 1000. Our feedforward neural network has one input layer and two hidden layers. Each layer has an activation function between them except the last layer as it has Softmax at the end. Once the training is complete the model is saved in a file named 'data.pth'. The training loss is calculated after each iteration and the final loss is calculated at the end, which is around 0.0002. This trained model is then used to predict the appropriate tags when the user asks a question to the chatbot. The working of the feedforward neural network is shown in Fig.2.4.

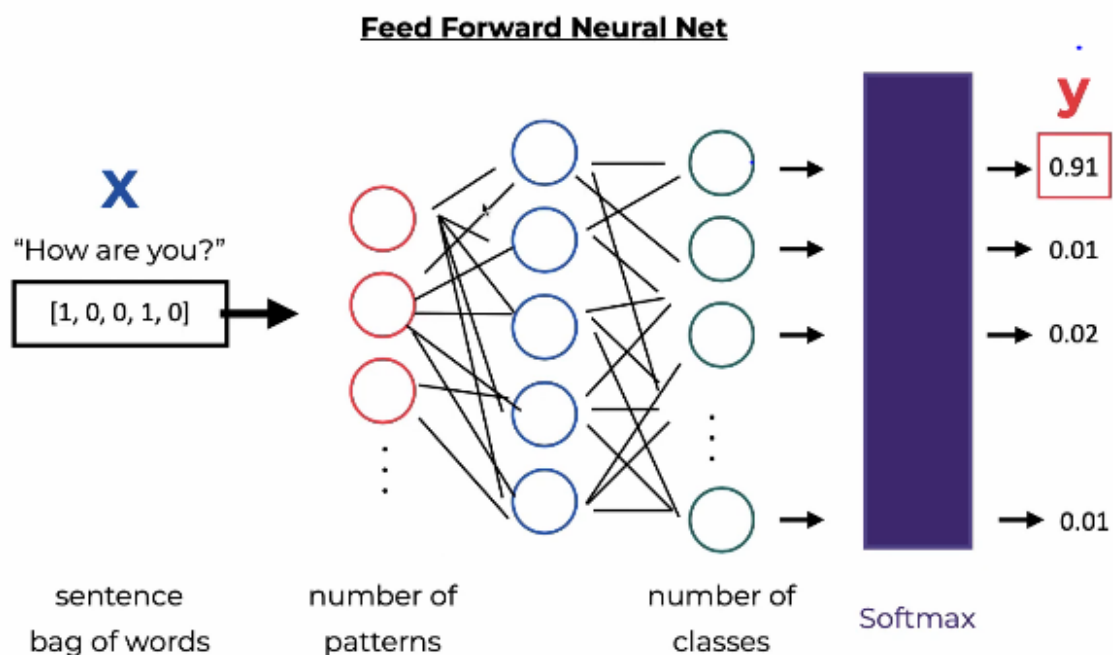


Figure 2.4: Working of feed forward neural net

2.5.2 Natural language processing

Natural language processing (NLP) facilitates human-to-machine communication without humans needing to speak Java or any other programming language as it allows machines to obtain and process information from written or verbal user inputs. In NLP there are various steps such as:

- **Tokenization:** Tokenization is a way of separating a piece of text into smaller units called tokens. Here, tokens can be either words, characters, or sub words. The formatting is based on spaces.
- **Stemming:** Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers. The input to the stemmer is tokenized words.
- **Bag of Words:** Bag of Words model is used to pre-process the text by converting it into a bag of words, which keeps a count of the total occurrences of most frequently used words. This model can be visualized using a table, which contains the count of words corresponding to the word itself.

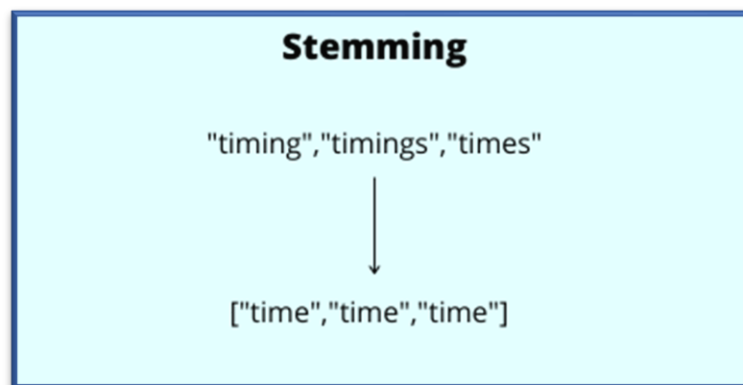


Figure 2.5: Stemming

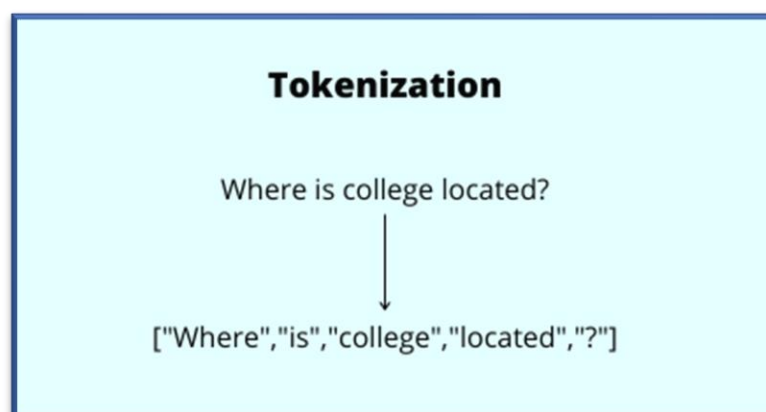


Figure 2.6: Tokenization

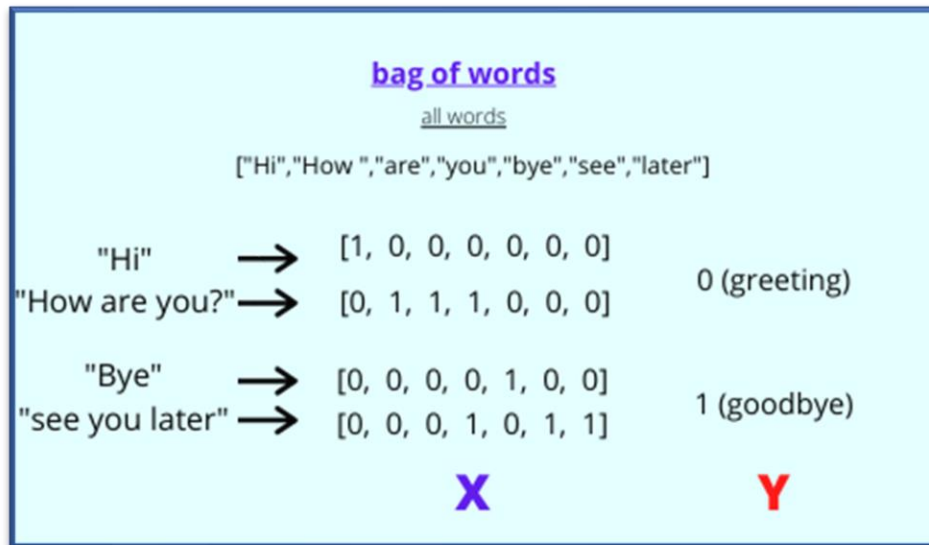


Figure 2.7: Bag of words

2.5.3 Dataset

The chatbot's storage is in the .json format. JavaScript Object Notation (JSON) is a text-oriented format that is widely used which uses JavaScript syntax. It stores and transmits data objects made up of attribute-value pairs and array data types using human-readable text. The .json file contains the intents i.e., the goal the user has in mind when typing in a question or query to the chatbot application. Every intent has a tag name saved in a variable named tag (for example greeting, committees, location, contact, etc.). Also, concerning every tag, there are several queries a user can ask under the heading patterns. There are different patterns in which a user can ask a single query. All these patterns are included in this pattern's variable. The responses to these queries are stored in the responses variable. The database is created to study the college website and analyzing what queries a user can ask and what responses he or she will expect. Also, google forms have been used to collect data for the database. Fig.2.8 shows two intents which are greetings and time. Here the questions in the pattern variable are input for the neural network model and tags are the output.

```
"intents":[
  {
    "tag":"greeting",
    "patterns":["Hi","hey","hello","Good morning","Good afternoon"],
    "responses":["Hey ! how can i help you ?"]
  },
  {
    "tag":"time",
    "patterns":["Timing of college ?","When can I visit the college ?"],
    "responses":["Monday To Friday","9am to 5pm"]
  }
]
```

Figure 2.8: Dataset

3.Implementation

```
import numpy as np
import nltk
# nltk.download('punkt')
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()

def tokenize(sentence):
    return nltk.word_tokenize(sentence)

def stem(word):
    return stemmer.stem(word.lower())

def bag_of_words(tokenized_sentence, words):
    # stem each word
    sentence_words = [stem(word) for word in tokenized_sentence]
    # initialize bag with 0 for each word
    bag = np.zeros(len(words), dtype=np.float32)
    for idx, w in enumerate(words):
        if w in sentence_words:
            bag[idx] = 1

    return bag
```

Figure 3.1: Natural Language Processing functions

```
import torch
import torch.nn as nn

class NeuralNet(nn.Module):
    def __init__(self, input_size, hidden_size, num_classes):
        super(NeuralNet, self).__init__()
        self.l1 = nn.Linear(input_size, hidden_size)
        self.l2 = nn.Linear(hidden_size, hidden_size)
        self.l3 = nn.Linear(hidden_size, num_classes)
        self.relu = nn.ReLU()

    def forward(self, x):
        out = self.l1(x)
        out = self.relu(out)
        out = self.l2(out)
        out = self.relu(out)
        out = self.l3(out)
        return out
```

Figure 3.2: Neural Network Model


```

# create training data
X_train = []
y_train = []
for (pattern_sentence, tag) in xy:
    # X: bag of words for each pattern_sentence
    bag = bag_of_words(pattern_sentence, all_words)
    X_train.append(bag)
    # y: PyTorch CrossEntropyLoss needs only class labels
    label = tags.index(tag)
    y_train.append(label)

X_train = np.array(X_train)
y_train = np.array(y_train)

# Hyper-parameters
train_losses=[]
num_epochs = 1000
batch_size = 8
learning_rate = 0.00017
input_size = len(X_train[0])
hidden_size = 8
output_size = len(tags)
print(input_size, output_size)

```

Figure 3.3: Creating training data and hyper-parameters

```

model = NeuralNet(input_size, hidden_size, output_size).to(device)

# Loss and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)

# Train the model
for epoch in range(num_epochs):
    for (words, labels) in train_loader:
        words = words.to(device)
        labels = labels.to(dtype=torch.long).to(device)

        # Forward pass
        outputs = model(words)
        loss = criterion(outputs, labels)

        # Backward and optimize
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

    if (epoch+1) % 100 == 0:
        print (f'Epoch [{epoch+1}/{num_epochs}], Loss: {loss.item():.4f}')
        train_losses.append(loss.item())

plt.figure(figsize=(10,5))
plt.title("Training Loss")
plt.plot(train_losses, label="train")
plt.xlabel("iterations")
plt.ylabel("Loss")
plt.legend()
plt.show()

data = {
    "model_state": model.state_dict(),
    "input_size": input_size,
    "hidden_size": hidden_size,
    "output_size": output_size,
    "all_words": all_words,
    "tags": tags
}

FILE = "data.pth"
torch.save(data, FILE)

print(f'training complete. file saved to {FILE}')

```

Figure 3.4: Training the model

```

model = NeuralNet(input_size, hidden_size, output_size).to(device)
model.load_state_dict(model_state)
model.eval()

def chat(msg):
    a=""
    while True:
        sentence = msg

        sentence = tokenize(sentence)
        X = bag_of_words(sentence, all_words)
        X = X.reshape(1, X.shape[0])
        X = torch.from_numpy(X).to(device)

        output = model(X)
        _, predicted = torch.max(output, dim=1)

        tag = tags[predicted.item()]

        probs = torch.softmax(output, dim=1)
        prob = probs[0][predicted.item()]
        if prob.item() > 0.75:
            for intent in intents['intents']:
                if tag == intent["tag"]:
                    responses = intent['responses']

            for ans in responses :
                a=a+ans+'<br/>'
            return([a])
        else:
            answer = "Sorry, but I don't understand your query; if you don't receive a response, please try to rephrase your question,"
            return(answer)

```

Figure 3.5: Prediction of tags

```

from flask import Flask, render_template
from flask_socketio import SocketIO, emit

app = Flask(__name__)

app.config[ 'SECRET_KEY' ] = 'jsbcfsbfjefebw237u3gdbdc'
socketio = SocketIO( app )

@app.route( '/' )
def hello():
    return render_template( './ChatApp.html' )

def messageRecived():
    print( 'message was received!!!' )

@socketio.on( 'my eventes' )
def handle_my_custom_event1( json1 ):
    import chatbot_final
    message = json1['message']
    answer=chatbot_final.chat(message)
    json1['answer'] = answer
    json1['bot']='APSIT BOT :'
    print( 'recived my event: ' + str(json1) )
    socketio.emit( 'my response', json1, callback=messageRecived )

if __name__ == '__main__':
    socketio.run( app, debug = True )

```

Figure 3.6: Flask

3.1 Algorithms

Natural language processing (NLP) is a branch of artificial intelligence that helps computers understand, interpret and manipulate human language. NLP draws from many disciplines, including computer science and computational linguistics, in its pursuit to fill the gap between human communication and computer understandings. Natural language processing includes many different techniques for interpreting human language, ranging from statistical and machine learning methods to rules-based and algorithmic approaches. We need a broad array of approaches because the text- and voice-based data varies widely, as do the practical applications. Basic NLP tasks include tokenization and parsing, lemmatization/stemming, part-of-speech tagging, language detection and identification of semantic relationships. If you ever diagramed sentences in grade school, you've done these tasks manually before. In general terms, NLP tasks break down language into shorter, elemental pieces, try to understand relationships between the pieces and explore how the pieces work together to create meaning.

Sentence	what are college timings?
Tokenization	['what', 'are', 'college', 'timings']
Stemming	['what', 'ar', 'colleg', 'time']
Bag of Words	what are college timings? College timings? [0 0 1 1]

Table 3.1: Natural language processing algorithms

3.2 Pseudo code

```
program start
take message from user
send message to model using Socketio
function chat(message)
    while True
        tokenize (message)
        stem (tokenized message)
        create bag of words of stemmed message
        give bag of words as X variable to model
        model predicts a tag
        if prediction >0.75
            loop through the tags in dataset
                if predicted tag is equal to tag in dataset
                    return response related to that tag from dataset
                end if
            end loop
        else
            return tag not found message
        end if
    end while
end function
```

3.3 Platforms for execution

Anaconda	Creating python environment.
Visual Studio Code	Executing and creating python codes.

Table 3.2: Software's used

1.Anaconda: Anaconda is an open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.

Here are the Anaconda requirements (minimum):

- 2+ cores or faster processor
- 2 GB of RAM

2.Visual Studio Code: Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

Here are the Visual Studio Code requirements (minimum):

- 1.6 GHz or faster processor
- 1 GB of RAM

4.Results

The ApsitBot uses NLP to process the user query and then this processed query is used by the neural network model to predict the response. The feedforward neural network model was trained with 54 intents with a learning rate of 0.00017. The number of epochs was set to 1000 and after 10 successful iterations, the final loss of the trained model was found to be 0.0002. Training loss was calculated after each iteration as shown in Table 4.1 and it is plotted using a graph in Fig.4.1. The plotted graph shows that the training loss is decreased to the point of stability. This indicated that this is an optimal fit.

Iterations	Loss
1	3.1367
2	1.5614
3	0.7708
4	0.3333
5	0.0911
6	0.0243
7	0.0073
8	0.0026
9	0.0005
10	0.0002

Table 4.1: Loss values

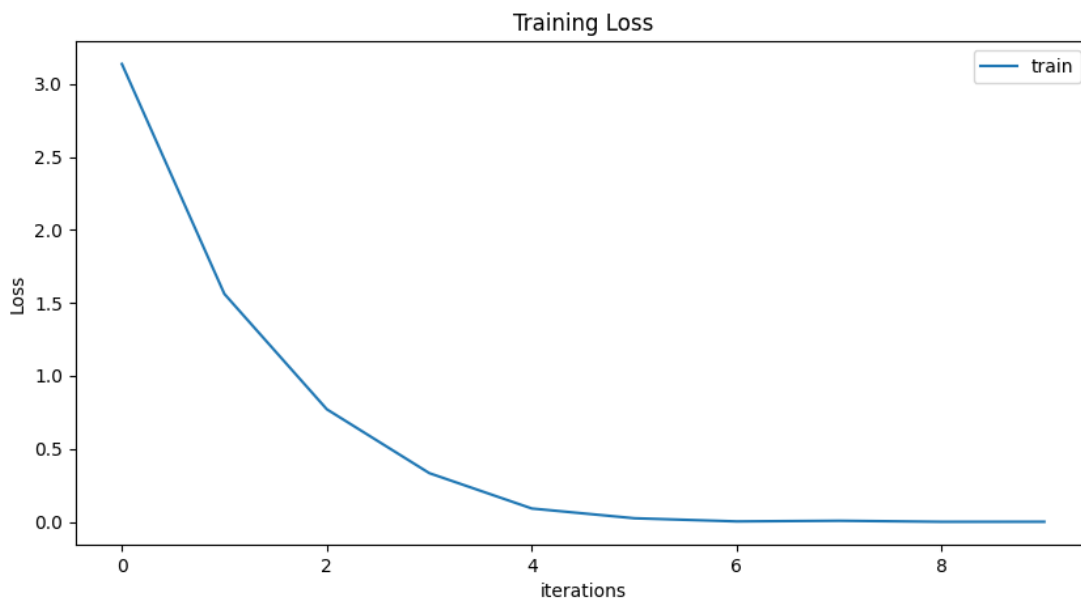


Figure 4.1: Training loss graph

Most of the chatbots which work on SQL queries [1][11] are unable to find a response when the questions are asked differently. As ApsitBot uses NLP and AI to find a response, the bot can find the response easily even if the questions are twisted. Fig.4.2 shows how the chatbot responds with the right answer even if the question was asked differently.

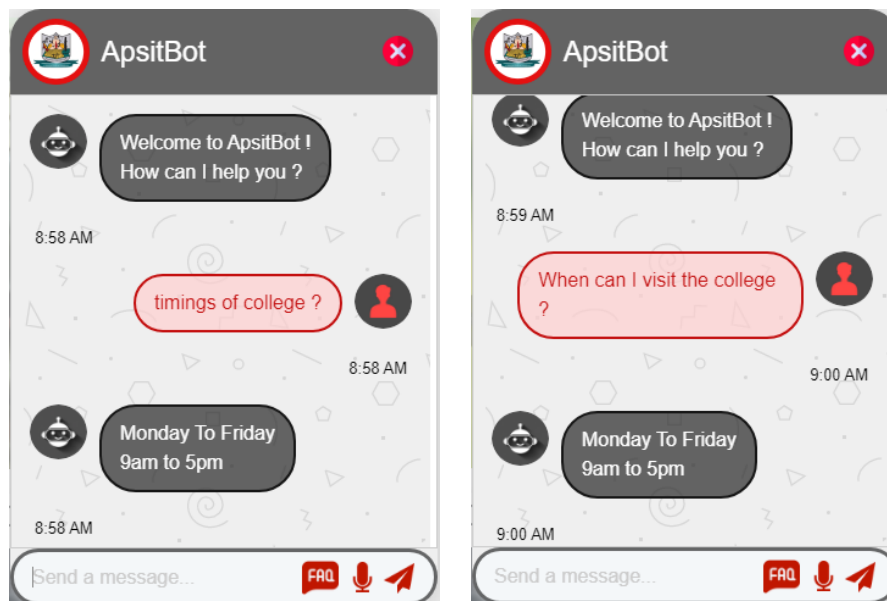


Figure 4.2: Screenshots of conversation with chatbot

The FAQ system as seen in Fig.4.3 helps the users to ask a predefined question without any need of framing a new one. These predefined questions are easily identified by the model and it gives accurate answers. This FAQ system helps in increasing the overall performance of the chatbot [2].

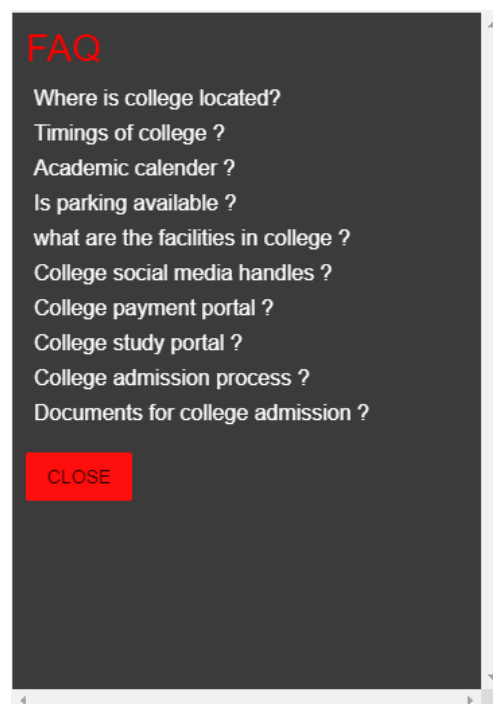


Figure 4.3: FAQ system in chatbot

The suggestion system implemented in the project as shown in Fig.4.4 suggests user's various questions while typing which helps them to ask appropriate questions and getting an accurate response. Also, messages can be received in the form of images from the chatbot. If the location of the college is queried, the message will include a Google map.

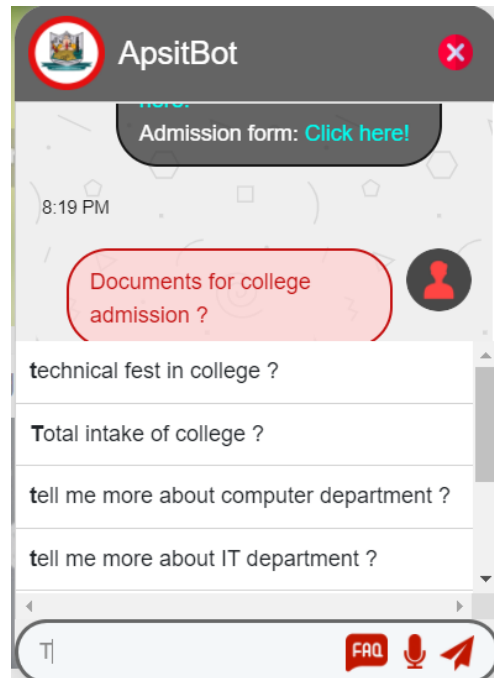


Figure 4.4: Suggestion system

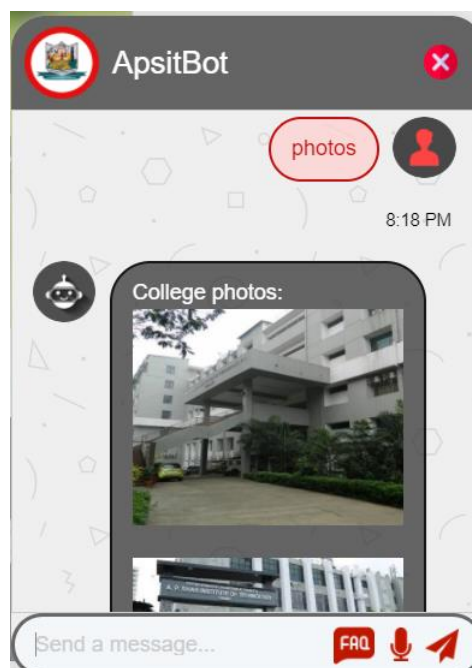


Figure 4.5: Images as reply

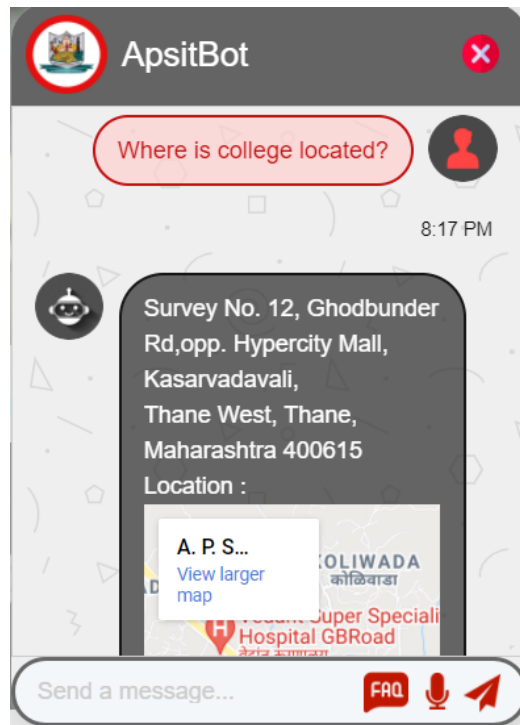


Figure 4.6: Google maps as reply

5. Conclusion

ApsitBot is a steppingstone technology developed with Natural Language Processing and Artificial Intelligence. It is a web-based application that uses Natural Language Processing Libraries and Machine Learning to have conversations with humans [9]. Even if not being familiar with the college website, the user can ask their queries to the bot and get desired answers. The result can be shown in the form of images and card format or text format. The query will be answered based on the questions that were posed, the language model that was developed, and the response media that was created. To ask any questions, the user can use the speech recognition functionality [5]. While typing questions, the chatbot offers suggestions that assist the user to ask the right questions and getting an accurate response to their queries. The bot has a FAQ button that allows users to ask the bot a commonly asked question with just one click.

The development of this chatbot addresses the issues that may arise when collecting information. This application can be relied on by students with considerable ease and without difficulties. As a result, the chatbot assists students in staying informed about college-related events and aims to save time for students, parents, and faculty at the institution

6.References

- [1] N. P. Patel, D. R. Parikh, D. A. Patel and R. R. Patel, "AI and WebBased Human-Like Interactive University Chatbot (UNIBOT)," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019.
- [2] B. R. Ranoliya, N. Raghuwanshi and S. Singh, "Chatbot for University Related FAQs", 2017 International Conference on Advances in Computing Communications and Informatics (ICACCI), 2017.
- [3] Souali, K., Rahmaoui, O., Ouzzif, M.,—&. El Haddioui, I. (2019). "Recommending Moodle Resources Using Chatbots", 2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS).
- [4] Lal, H., & Lal, P. (2019). NLP chatbot for Discharge Summaries. 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT).
- [5] Fernando A. Mikic Fonte, Mart'in Llamas Nistal, Juan C. Burguillo Rial and Manuel Caeiro Rodr'iguez, "NLAST: A natural language assistant for students" in , Vigo, Spain:Department of Telematics Engineering, University of Vigo.
- [6] Amey Tiwari, Rahul Talekar and S.M. Patil, "College Information Chat Bot System", International Journal of Engineering Research and General Science, 2017.
- [7] Balbir Singh Bani and Ajay Pratap Singh, "College Enquiry Chatbot Using A.L.I.C.E", International Journal of New Technology and Research, 2017.
- [8] Mondal, A., Dey, M., Das, D., Nagpal, S., & Garda, K. (2018). Chatbot: An automated conversation system for the educational domain. 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP).
- [9] H. K. K., A. K. Palakurthi, V. Putnala and A. Kumar K., "Smart College Chatbot using ML and Python," 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2020.
- [10] Chun Ho, C., Lee, H. L., Lo, W. K., & Lui, K. F. A. (2018). Developing a Chatbot for College Student Programme Advisement. 2018 International Symposium on Educational Technology (ISET).
- [11] Prof.K. Bala, Mukesh Kumar, Sayali Hulawale and Sahil Pandita, "ChatBot For College Management System Using A.I", International Research Journal of Engineering and Technology (IRJET), vol. 04, no. 11, pp. 2030- 2033, Nov 2017.
- [12] Guruswami Hiremath, Aishwarya Hajare, Priyanka Bhosale, Rasika Nanaware and K. S. Wagh, "Chatbot for education system", International Journal of Advance Research Ideas and Innovations in Technology (IJARIIT), vol. 4, no. 3, pp. 37-43, 2018.
- [13] Unnati Dhavare and Umesh Kulkarni, "Natural Language Processing using Artificial Intelligence", International Journal of Emerging Trends& Technology in Computer Science

(IJETTCS), vol. 4, no. 2, pp. 203-205, April 2015.

[14] L. Tommy, C. Kirana and L. Riska, "The Combination of Natural Language Processing and Entity Extraction for Academic Chatbot," 2020 8th International Conference on Cyber and IT Service Management (CITSM), Pangkal, Indonesia, 2020.

[15] Mutiwokuziva, M. T., Chanda, M. W., Kadebu, P., Mukwazvure, A., & Gotor, T. T. (2017). A neural-network based chat bot. 2017 2nd International Conference on Communication and Electronics Systems (ICCES).

[16] M. Dahiya, "A tool of conversation: Chatbot" in , Janakpuri, India:Dept. of Computer Science, Maharaja Surajmal Institute, 2017.

[17] Kumar, P., Sharma, M., Rawat, S., & Choudhury, T. (2018). Designing and Developing a Chatbot Using Machine Learning. 2018 International Conference on System Modeling & Advancement in Research Trends (SMART).

[18] Vamsi, G. K., Rasool, A., & Hajela, G. (2020). Chatbot: A Deep Neural Network Based Human to Machine Conversation Model. 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT).

[19] M, D., JVSL, J., E, S., R, L., & Manne, D. S. (2020). Interactive Transport Enquiry with AI Chatbot. 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS).

[20] Ajinkya Huddar, Chaitanya Bysani, Chintan Suchak, Uttam D Kolekar, Kaushiki Upadhyaya, "Dexter the College FAQ Chatbot", Convergence to Digital World - Quo Vadis (ICCDW) 2020 International Conference on, pp. 1-5, 2020.

7. Bibliography

[1] B. R. Ranoliya, N. Raghuwanshi and S. Singh, "Chatbot for University Related FAQs", 2017 International Conference on Advances in Computing Communications and Informatics (ICACCI), 2017.

[2] Youtube Channels:

(URL) PythonEngineer: <https://www.youtube.com/c/PythonEngineer>

(URL) edureka!: <https://www.youtube.com/c/edurekaIN>

[3] <https://www.ultimate.ai/blog/ai-automation/how-nlp-text-based-chatbots-work>

[4] <https://chatbotsmagazine.com/contextual-chat-bots-with-tensorflow-4391749d0077>

[5] <https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1>

8.Publication

Paper entitled “**AI and Web-Based Interactive College Enquiry Chatbot**” is accepted at “*The 3rd World Symposium on Artificial Intelligence (WSAI 2021)*” by Rohan Parkar, Yash Payare, Keyur Mithari, Jitesh Nambiar, and Prof. Jaya Gupta.