##### 

##### Chat Bot (PAAM)

**A PROJECT REPORT**

###### Submitted by

20MIM10036 - Prakhyat Nandan Srivastava

20MIM10004 - Aditya Baraiya

20MIM10058 - Abhrojit Sarkar

20MIM10045 - Manan Patel

*in partial fulfilment for the award of the degree*

*of*

Integrated MTech

*in*

Machine Learning and Artificial Intelligence

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**VIT BHOPAL UNIVERSITY**

**KOTHRIKALAN, SEHORE**

**MADHYA PRADESH - 466114**

**Vellore Institute of Technology Bhopal**

**DECEMBER 2021**

**VIT BHOPAL UNIVERSITY, KOTHRIKALAN, SEHORE**

**MADHYA PRADESH – 466114**

**Integrated MTech AI & ML**

**BONAFIDE CERTIFICATE**

Certified that this project titled **“Chat Bot (PAAM)”** is the bonafide work of **“Prakhyat Nandan Srivastava(20MIM10036), Aditya Baraiya(20MIM10004), Abhrojit Sarkar(20MIM10058) and Manan Patel(20MIM10045)”** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate

|  |  |
| --- | --- |
| **Dr. Mayuri** | **Dr. Paras Jain** |
| **Program Chair-Integrated M.Tech AI** | **Senior Assistant Professor** |
| School of Computer Science and Engineering  VIT BHOPAL UNIVERSITY | School of Computer Science and Engineering  VIT BHOPAL UNIVERSITY |

**ACKNOWLEDGEMENT**

We first and foremost would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

We wish to express my heartfelt gratitude to **Dr. Mayuri A V R,** Program chair integrated MTech, School of Computer Science for much of his valuable support and encouragement in carrying out this work.

We would like to thank my internal mentor **Dr. Paras Jain**, for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

We would like to thank all the technical and teaching staff of the School of Computer Science, who extended directly or indirectly all support.

Last, but not least, we are deeply indebted to my parents who have been the greatest support while we worked day and night for the project to make it a success.

**ABSTRACT**

A chatbot is an artificial intelligence computer program which performs communication using a chat system. A person can ask any questions and the chatbot will answer accordingly. Nowadays a chatbot is highly popular and takes speed as a computer communication application. Chatbot system is in trend, thus it is being used on many websites. With the chatbot, one doesn’t have to wait to talk to the customer helpline, they don’t even have to search for shopping through Websites. A chatbot is used in many areas like order food, product suggestions, customer support, weather, personal finance assistance, scheduled a meeting, search and track flights, send money, and many more. The College bot project is built using artificial algorithms that analyse user’s queries and understand the user's message. This System is a web application which provides answers to the query of the student. Students just have to query through the bot which is used for chatting. Students can chat using any format there is no specific format the user has to follow. The System uses built in artificial intelligence to answer the query. The answers are appropriate to what the user queries. The User can query any college related activities through the system. The user does not have to personally go to the college for enquiry. The System analyses the question and then answers to the user. The system answers to the query as if it is answered by the person. With the help of artificial intelligence, the system answers the query asked by the students. The system replies using an effective Graphical user interface which implies that as if a real person is talking to the user. The user can query about the college related activities online with the help of this web application. The user can query college related activities such as date and timing of college, Programs, Faculties, location and many more. This system helps the student to be updated about the college.

**List of Abbreviation:**

1. PAAM -Chatbot’s name
2. NLTK - Natural Language Toolkit
3. FAQ's - Frequently Asked Questions
4. Cmd - Command
5. Info - Information
6. OS - Operating System
7. RAM - Random Access Memory
8. POS - Part of Speech
9. UI - User Interface

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| 4.1  4.2 | Architecture Diagram  System Flow Diagram | 18  19 |
| 5.1  5.2  5.3  5.4  5.5 | Libraries Of Python  NLTK Library  TensorFlow Library  Data Analysis  Training Data | 21  22  22  24  24 |
| 6.1  6.2 | PAAM (chatbot) Outcome  PAAM (chatbot) Introduction | 27  27 |
| 7.1  7.2  7.3 | Problem Of PAAM (chatbot)  Resolving The Problem of PAAM (chatbot)  Resolved Problem of PAAM (chatbot) | 29  30  30 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | Abstract  List of Abbreviation:  List of Figures | 4  5  6 |
| 1 | **CHAPTER-1:**  **PROJECT DESCRIPTION AND OUTLINE** Introduction  * 1. Motivation for the work   2. Problem Statement   3. Objective of the work   4. Organization of the project   5. Summary | 9  9  10  10  11  11 |
| 2 | **CHAPTER-2:**  **RELATED WORK INVESTIGATION**  2.1 Introduction  2.2 Literature Review  2.3 Summary | 12  12  15 |
| 3 | **CHAPTER-3:**  **REQUIREMENT ARTIFACTS**  3.1 Introduction  3.2 Hardware and Software requirements  3.3 Specific Project requirement | 16  16  17 |
| 4 | **CHAPTER-4:**  **DESIGN METHODOLOGY AND ITS NOVELTY**  4.1 Methodology and goal  4.2 Novelty of the Project  4.3 Functional modules analysis  4.4 Architecture Diagram  4.4.1 System Architecture diagram  4..4.1 System Flow diagram  4.5 Summary | 18  18  19  19  20 |
| 5 | **CHAPTER-5:**  **TECHNICAL IMPLEMENTATION & ANALYSIS**  5.1 Technical coding and code solutions  5.2 Working Layout of Forms  5.3 Test and validation  5.4 Performance Analysis (Graphs/Charts)  5.5 Summary | 21  21  22  25  25 |
| 6 | **CHAPTER-6:**  **PROJECT OUTCOME AND APPLICABILITY**  6.1 key implementations of the System  6.2 Significant project outcomes  6.3 Project applicability on Real-world applications  6.4 Inference | 26  28  28  29 |
| 7 | **CHAPTER-7:**  **CONCLUSIONS AND RECOMMENDATION**  7.1Limitation of the System  7.2 Future Enhancements  7.3 Inference | 30  30  31 |
|  | References | 32 |

**CHAPTER-1**

**PROJECT DESCRIPTION AND OUTLINE**

* 1. **– Introduction:**

CHATBOTS are automated systems which replicate users’ behavior on one side of the chatting communication. They are mimic systems which imitate the conversations between two individuals. They provide a simulating platform for effective and smart communications with the user on the other end. They copy marketers, sales person, counsellors and other mediators and work to provide service that the above-mentioned people provide. There are wide ranges of chatbots catering in many domains some of them are as follows: business, market, stock, customer care, healthcare, counselling, recommendation systems, support system, entertainment, brokering, journalism, online food and accessory shopping, travel chatbots, banking chatbots, recipe guides, etc. The most famous chatbots like Alexa or Google assistant are the best examples that can be given for smart communicating chatbots. Chat has become the center of focus in this current era; thus, the bots are being utilized to deliver information engagingly and conveniently. A chatbot is standout amongst the most progressive and promising tools of communication among people and machines A user wants more automation in the chatbot. Although every system is not perfect there is always a flaw in the system, so as in the chatbot there are some problems that the user has experienced while using a chatbot. Chatbot can be described as an answering system where a system will be able to answer questions or statements submitted by users and allow users to control over the content to be displayed. A bot is trained on and according to the training, based on some rules on which it is trained, it answers questions. It is called ruled based approach. Using these ruled based approach, creation of these bots becomes relatively straight forward.Our chatbot we named it PAAM, which is made for college website for helping students who have FAQs about college. This PAAM (chatbot) is developed using Python. The project file contains a python script (main.py, trainingData.py, JSON file, and pkl file). Talking about this chatbot, it allows the user to provide suitable queries about the college and replies with suitable answers. You change the data by changing the given JSON file. Also, this is a simple cmd-based project which is easy to understand and use.

* 1. **Motivation for the work:**

As we all know chatbots square measure a really fashionable approach of communication between the user the system text interface. we have a tendency to wished to use this system to implement college web site for an additional economical communication between the student and administration. Students will use school web site chatbot for Queries rather than about to workplace or authorities directly for info. College web site based mostly bot’s square measure accustomed develop effective communication between different branches and facilitate management to produce innovative solutions. Our Project to create a university web site chatbot would undoubtedly have a bright scope as this could bring a modification within the approach of interaction between school and students This Project would inspire alternative schools still to develop internet chatbots to Ease the communication between Management and Students

* 1. **Problem Statement:**

After visiting many schools or college websites we have found so many different types of chatbots by different company and we noticed that the chatbot are slow and glitched and also some got hanged while chatting with it so we try to make a chatbot which is faster and our chat bot is run on very basic software and hardware requirements so it never going to stuck while some chatting with it. we also try to make an easy-to-use chatbot. Our chatbot giver answer to FAQs asked about college or school by their students.

**1.4 Objective of The Work:**

The purpose of faculty info chatbots is to support students and school in their relations with school. Doing this helps save heaps of your time {for students, for college, for school students} as they don’t ought to visit college workplace for info. Instead, they'll raise Queries to the Chatbots within the college web site and acquire info they need with Ease. Our Aim is to use Chatbot Technology to develop our college web site for additional economical approach of Communication between the school and Students. Students will use school web site chatbot for Queries rather than about to workplace or authorities directly for info. school web site based mostly bots square measure accustomed develop effective communication between totally different branches and facilitate management to produce innovative solutions. Our Project to create {a school |a university} web site chatbot would undoubtedly have a bright scope as this could bring a modification within the approach of interaction between college and students. This Project would inspire alternative schools still to develop internet chatbots to Ease the communication between Management and Students.

**1.5 Organization of The Project:**

The Documentation of project contains the full clarification of project in varied aspects. The Documentation is organized in an order that a project ought to be. the primary section contains the introduction of the project that explains concerning the motivation, drawback definition, objectives and limitations of the project. The project Introduction is followed by the literature survey that has existing systems that is followed by the analysis section of the project that contains all the wants & diagrams of the project. Then comes style| the planning |the look section that explains concerning the diagrams concerning the module design of the project. The implementation and results section explains concerning the key functions, output screens, tables and results of the project.

* 1. **Summary:**

As you can see CHATBOTS are automated systems which replicate users’ behavior on one side of the chatting communication. They are mimic systems which imitate the conversations between two individuals. Our chatbot we named it PAAM, which is made for college website for helping students who have FAQs about college. This PAAM (chatbot) is developed using Python. we have a tendency to wished to use this system to implement college web site for an additional economical communication between the student and administration. Students will use school web site chatbot for Queries rather than about to workplace or authorities directly for info. The purpose of faculty info chatbots is to support students and school in their relations with school. Doing this helps save heaps of your time {for students for college | kids for college students} as they don’t ought to visit college workplace for info. Instead, they’ll raise Queries to the Chatbots within the college web site and acquire info they need with Ease.

**CHAPTER-2:**

**RELATED WORK INVESTIGATION**

**2.1 Introduction:**

In recent years, chatbot systems have been widely used and have received much attention. However, chatbot projects have often It has been found to be complex and very time-consuming to implement on college or school websites. The organizational relevance Chatbot projects make it a part of an organization’s focus to look for ways to make chatbot implementation a success. We collected and analyzed a number of key articles discussing and analyzing how to implement a chatbot. From a time, management point of view, the different approaches taken in the literature were compared to highlight the key codes and their impact on the chatbot. Literature was further classified in order to address and analyze each code and its relevance during the stages of the chatbot project.

**2.2 Literature Review:**

Naeun Lee et al. [2017] proposed the implementation of word segmentation using NLTK. Natural Language Toolkit (NLTK) is a python package which caters to provide services for NLP. It has inbuilt tokenizers. Users need to import the package and use the required type of tokenizer which is present in the form of functions. The NLTK includes a wide range of tokenizers which are as follows: standard, letter, word, classic, lowercase, N-gram, pattern, keyword, path, etc. The most commonly used tokenizer is the word-punkt tokenizer which splits the sentences at the blank spaces. The accuracy, speed and efficiency of the NLTK tokenizers is commendable. Also, it does not require any algorithm implementation as the package executes them at the backend [Ref no. 3].

A review on Word Segmentation, also referred to as tokenization, is the process of splitting text into smaller and meaningful units. These units could be paragraphs, sentences, clauses, phrases, words or letters. The smallest unit is the lette27rs. Word segmentation is the splitting of sentences into individual words separated by blank spaces. The tokenized units of the sentences are called tokens. The tokenizers split the sentences into words and punctuation marks as independent units. The most commonly used tokenizer is of space type, i.e., it splits the sentences into words at the blank spaces. It is also required that the tokenizer should consider abbreviations, acronyms, dates, numbers in decimal formats, etc., which cannot split at punctuations and blank spaces, as they will lose their meaning if done so.

Mohammed Javed et al. [2015] explained a method to implement word segmentation. He proposed in his algorithm to calculate character spaces in the sentences. The character spaces should include all types of gaps between characters. They include the gaps between letters, punctuations and the words. The algorithm functions on the basis of the amount of gap or character space between each unit in the sentence. After the calculation of character spaces, an average of the gaps is calculated to know the mean average between characters in the sentence. This average gap distance is then applied to the sentence which is to be segmented. The places at which the character space is more than the average character space are said to be the points of tokenization. The gap between words is always more than the average gap and hence tokenization takes place at the blank spaces between words in the sentences [Ref no. 2].

A review on POS Tagging POS Tagging is the process of assigning grammatical annotations to individual words in the sentences. These annotations include the Parts-Of-Speech Tags. They denote the grammatical importance of the word in the sentence based on the dependency of that word with other words in that phrase, clause, sentence, paragraph, etc. The common POS tags are noun, verb, pronoun, etc. There are a number of ways which can be used to perform POS Tagging. Some of them are explained below.

Jerome R. Bellegarda [4] [2010] proposed a method called latent analogy for POS Tagging. In this algorithm, latent semantic mapping (LSM) technique is used. It requires the training on the available corpus. The LSM maintains a feature space of the trained corpus which has been tagged. Now, new sentences are provided to the LSM for tagging and the analysis is performed so as to determine the sentences from the training data which are closest to the test sentence. This is called a sentence neighbourhood. Sentence neighbourhood holds true for two sentences if they share the same intent matter. Once the intent matching sentences are found from the trained data, the POS tags attached to those sentences are then mapped to the test sentences [Ref no. 6].

Liner Yang et al. [2018] put forth the technique of implementing the POS Tagger using Neural Networks. This algorithm consists of “n” numbers of hidden layers. These layers are determined by the number of iterations or combinations required to tag the required sentence correctly. At each layer of the algorithm, each word in the sentence is tagged with an appropriate POS tag and then passed to the next later for checking the correctness of the tags. This keeps happening unless the next layer provides the same tags as provided by the previous layer. Another technique to implement the POS tagger is following the traditional approach i.e., of maintaining a dictionary of tags for the given language. Python NLTK provides an inbuilt Tagger which can be used just by importing the NLTK package. The NLTK has a pre-defined set of tags and trained data of its own. It tests the sentence and applies an appropriate tag to it. On comparing the above three algorithms, the NLTK tagger proves to be speed and usage efficient. But highest accuracy is provided by the neural network algorithm as it undergoes many iterations [Ref no. 5].

LinHua Gao et al. [2018] explains the traditional dictionary method of synonym extractions. In this method, the system database maintains a dataset of synonyms for important keywords in that domain. The sentence sent by the user is then mapped on to that synonym dataset. The keywords detected from the sentence are then checked in that synonym set to check for the same intent. All

possible synonyms of that keyword are then looked out for a match in the main database. The sentence which is closest to the user sentence is extracted. This method is time consuming and requires more storage and complexity [Ref no. 7].

Sijun Qin [2015] proposed a feature selection method for synonym extraction. In this method, among all the parts of speech tags, Words having the tags as nouns, verbs and adjectives are marked as positive tags and the others as negative tags. The polarity for Each feature (word) is then carried out by using the POS tags. If the overall feature polarity is positive, then it can be identified categorically. All the positive features are then grouped together and the synonyms detection for the group of features will be relatively strong, as an entire clause is checked for its synonymic meaning. The synonym sets which are extracted for that clause of features are then calculated for information gain. The one with the highest information gain is the strongest synonym extracted [Ref no. 8].

**2.3 Summary:**

As you can see CHATBOTS are automated systems which replicate users’ behavior on one side of the chatting communication. They are mimic systems which imitate the conversations between two individuals. Our chatbot we named it PAAM, which is made for college website for helping students who have FAQs about college. This PAAM (chatbot) is developed using Python. we have a tendency to wished to use this system to implement college web site for an additional economical communication between the student and administration. Students will use school web site chatbot for Queries rather than about to workplace or authorities directly for info. The purpose of faculty info chatbots is to support students and school in their relations with school. Doing this helps save heaps of your time {for students for college | kids for college students} as they don’t ought to visit college workplace for info. Instead, they’ll raise Queries to the Chatbots within the college web site and acquire info they need with Ease.

**CHAPTER-3:**

**REQUIREMENT ARTIFACTS**

**3.1 Introduction**

Our Chatbot is dependent on the following software and operating systems: Windows 7 (with. .NET Framework 4.5.2), 8.0, 8.1 and 10, 11 (32-bit and 64-bit) Microsoft Visual C++ Redistributable 2017 x64 package.

**3.2 Hardware And Software Requirements**

Our Chatbot is dependent on the following software:

* Visual Studio Code (to build the chat bot)
* Version 11.3.5.5 Python 3.8.8 (64 bit)
* Microsoft .NET Framework v4.7.2. The system prompts for a restart to complete the update.
* Microsoft Visual C++ Redistributable 2017 x64 package

**Hardware:**

* 1.6 GHz or faster processor
* 1 GB of RAM

1. OS X El Capitan (10.11+)
2. Windows 7 (with .NET Framework 4.5.2), 8.0, 8.1 and 10, 11 (32-bit and 64-bit)
3. Linux (Debian): Ubuntu Desktop 16.04, Debian 9
4. Linux (Red Hat): Red Hat Enterprise Linux 7, CentOS 8, Fedora 24
5. Additional Windows requirements

Microsoft .NET Framework 4.5.2 is required for VS Code. If you are using Windows 7, please make sure .NET Framework 4.5.2 is installed.

**3.3 Specific Project Requirements**

This automated communication system is developed using Python. The project file contains a python script (main.py, trainingData.py, JSON file, and pkl file). Talking about this chatbot, it allows the user to provide suitable queries about the college and replies with suitable answers. You change the data by changing the given JSON file. Also, this is a simple cmd-based project which is easy to understand and use.

**Library Requirements**:

Pip, NumPy, random, nltk and TensorFlow

**CHAPTER-4:**

**DESIGN METHODOLOGY AND ITS NOVELTY**

**4.1 Methodology:**

Successful chatbots are designed to learn, making maintenance an integral part of the chatbot development methodology.

Once the chatbot is interacting with real users it is important to analyze user feedback and sentiment, along with other insights the interactions may produce.

Insight analysis may give us an understanding of possible usability issues or areas of improvement, but it also may provide us with possible market opportunities to implement more data-driven solutions.

**4.2 Novelty of The Project**

1. PAAM Should Provide Faster Solutions:

PAAM can easily save time and effort. It can help resolve base level complex issues without student intervention. Instead of having a student going through page after page and analyzing data. One of the core features of PAAM should be to provide efficient solutions to student’s many problems on request.

1. We Keep a Simple & Easy User-Interface for PAAM:

PAAM is easy to find, proactive, engaging, and aesthetically appealing.

1. It helps the students who are facing problem to search information about college from their website:

PAAM will help students where students can ask direct questions and get their answers what they need.

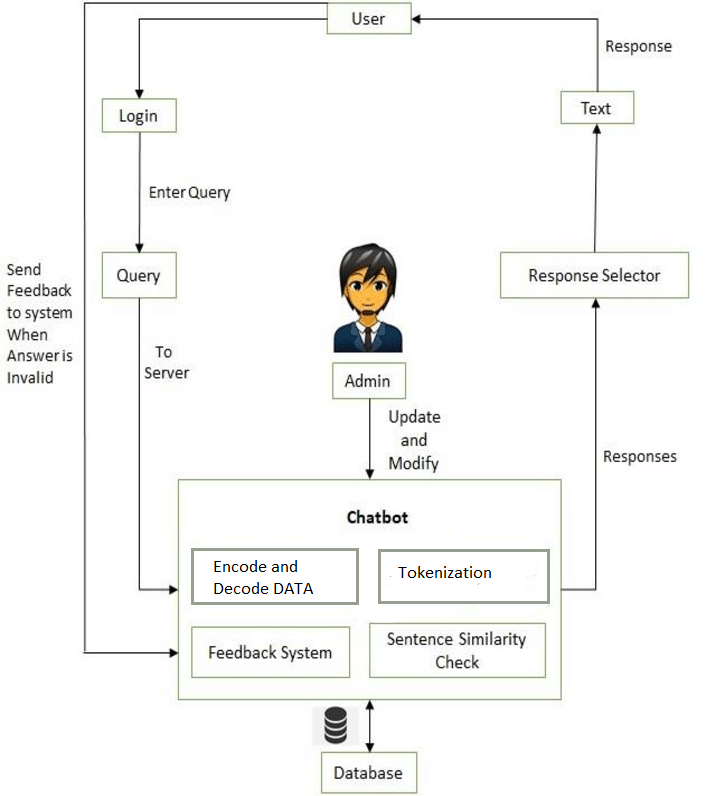
1. Machine learning based system:

PAAM will run on a neural network so it can compare and give answers correctly.

**4.3 Functional Modules Analysis**

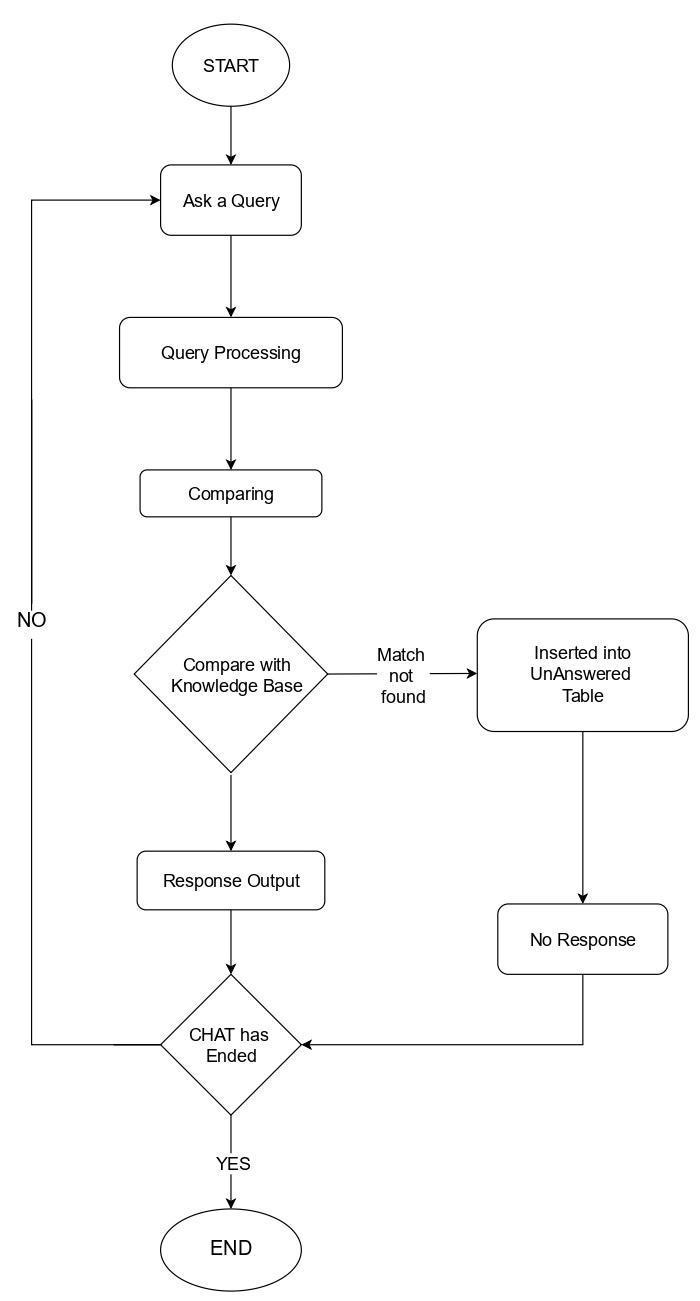
The project file contains a python script (main.py, trainingData.py, JSON file, and pkl file). Talking about this chatbot, it allows the user to provide suitable queries about the college and replies with suitable answers. You change the data by changing the given JSON file. Also, this is a simple cmd-based project which is easy to understand and use.

**4.4 Architecture Diagram**



**Fig. 4.1 Architecture Diagram**

**4.2 Subsystem Services**



**Fig.4.2 System Flow Diagram**

**4.5 Summary**

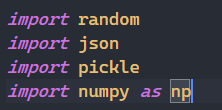
Analyzing user feedback and sentiment is an integral part of any chatbot's development and post-mortem will help identify potential areas for improvement. PAAM Chatbot will run on a neural network so it can compare and give answers correctly. It helps students who are facing problems to search information about college from their website.

**CHAPTER-5:**

**TECHNICAL IMPLEMENTATION & ANALYSIS**

**5.1 Technical Coding and Code Solutions:**

We have used different libraries in our chatbot. The random module is another library of functions that can extend the basic features of python. Other modules we have seen so far are string, math, time and graphics. With the exception of the graphics module, all of these modules are built into python. For a full list of python modules, see the online documentation listing all of the default modules. To get access to the random module, we add from random import \* to the top of our program (or type it into the python shell).



**Fig. 5.1 Libraries of Python**

Another module is json. Json is file that contains data set of the chatbot. Json is saved on that system which run the code for chatbot. JSON is a syntax for storing and exchanging data. SON is text, written with JavaScript object notation.

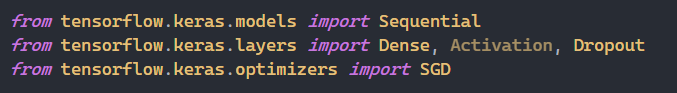
Pickle is also one of the modules. Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it “serializes” the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.

NumPy library is also used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open-source project and you can use it freely. NumPy stands for Numerical Python. n Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ND array, it provides a lot of supporting functions that make working with ND array very easy. Arrays are very frequently used in data science, where speed and resources are very important.



**Fig. 5.2 NLTK Library**

NLTK (Natural Language Toolkit) Library is a suite that contains libraries and programs for statistical language processing. It is one of the most powerful NLP libraries, which contains packages to make machines understand human language and reply to it with an appropriate response. Lemmatization is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item. Lemmatization is similar to stemming but it brings context to the words. So, it links words with similar meanings to one word. Text pre-processing includes both Stemming as well as Lemmatization. Many times, people find these two terms confusing. Some treat these two as the same. Actually, lemmatization is preferred over Stemming because lemmatization does morphological analysis of the words.



**Fig.5.3 Tensorflow Library**

**Fig.5.3 Tensorflow Library**

TensorFlow is a popular JavaScript library for Machine Learning. Let’s us train and deploy machine learning models in the Browser. Let’s us add machine learning functions to any Web Application.

**5.2 Test and Validation:**

Getting numerical values (0 and 1) of each word for neural network for classes and bags bag is a group of words for particular sentence.

To check the word, exist or not. This will check that if word exist in our data set i.e. (bag) it will return 1 and if not, it will return 0 After checking the word is correct it will append the word to bag and all the words which are in bag will process for giving the correct answer.

**Code:**

for document in documents:

bag =[]

word\_patterns = document[0]

word\_patterns = [lemmatizer.lemmatize(word.lower()) for word in word\_patterns]

for word in words:

bag.append(1) if word in word\_patterns else bag.append(0)

# to append the data to training list

output\_row = list(output\_empty)

output\_row[classes.index(document[1])] = 1

training.append([bag, output\_row])

This code explains notice that our data is shuffled. TensorFlow will take some of this and use it as test data to gauge accuracy for a newly fitted model. If we look at a single x and y list element, we see ‘bag of words’ arrays, one for the intent pattern, the other for the intent class.

**5.3 Performance Analysis (Graphs/Charts)**

We’re ready to build our model. For performance analysis we build the neural network.We built a neural network in Python (no frameworks), and we showed how machine learning could ‘learn’ from patterns of data. Recall the ‘json’ data is purposefully simple so that we can intuitively grok the patterns within it.

The notebook for this “zero abstraction” code is here. Each mathematical operation within the model is detailed.Extending our model to use 2 hidden layers and Gradient Descent such as the one we built for analyzing text. This is an example of “Deep Learning, the “depth” comes from the hidden layers.The definition of our model is relatively straight-forward, albeit laborious, most of the code is applied to training:

**Code:**

# building neular network model

model = Sequential()

# defining layers of neular network

model.add(Dense(128, input\_shape=(len(train\_x[0]),), activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(64, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(len(train\_y[0]), activation='softmax'))

# training neular network

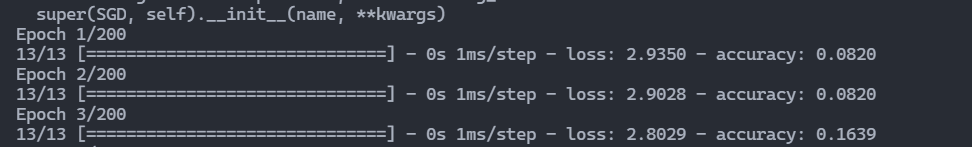
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)

model.compile(loss='categorical\_crossentropy', optimizer=sgd, metrics=['accuracy'])

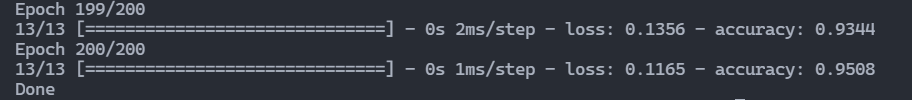
hist = model.fit(np.array(train\_x), np.array(train\_y), epochs=200, batch\_size=5, verbose=1)

model.save('chatbotmodel.h5', hist)

we built for analyzing text, we have ~80 lines of code



**Fig. 5.4 Data Analysis**



**Fig. 5.5 Training Data**

**5.4 Summary:**

As you saw we used different libraries(modules) of python to make the chatbot. We took help of random, pickle, json, NumPy and TensorFlow. We took help of Natural Language Toolkit (NLTK) to give appropriate answer to user. Lemmitizer is also used for the words whose meanings are the same. Different types of models like Sequential, Dense, dropout and SGD is also used. We made a bag which contains the words of the sentences.

**CHAPTER-6:**

**PROJECT OUTCOME AND APPLICABILITY**

**6.1 Key Implementations of The System:**

To convert numerical data to words what we need to do that we define four different functions. function that cleans up the sentences. It will delete the same words and ignore punctuation.

We will load our saved TensorFlow (tflearn framework) model. Notice you first need to define the TensorFlow model structure. Before we can begin processing intents, we need a way to produce a bag-of-words from user input.

**Code:**

def clean\_up\_sentence(sentence):

sentence\_words = nltk.word\_tokenize(sentence)

sentence\_words = [lemmatizer.lemmatize(word) for word in sentence\_words]

return sentence\_words

# funtion of getting bag of words

# it will convert sentences to a bag of words which is numerical values

def bag\_of\_words(sentence):

sentence\_words= clean\_up\_sentence(sentence)

bag = [0] \* len(words)

for w in sentence\_words:

for i, word in enumerate(words):

if word == w:

bag[i] = 1

return np.array(bag)

We are now ready to build our response processor.

# Function for predicting the class based on the sentence essentially

def predict\_class(sentence):

bow = bag\_of\_words(sentence)

res = model.predict(np.array([bow]))[0]

ERROR\_THRESHOLD = 0.25

results = [[i,r] for i, r in enumerate(res) if r > ERROR\_THRESHOLD]

# it will return the best predicted value of answer for particular quetion

results.sort(key=lambda x:x[1], reverse=True)

return\_list = []

for r in results:

return\_list.append({'intent': classes[r[0]], 'probability': str(r[1])})

return return\_list

# Function for getting response

# It will return the correct answer

def get\_response(intents\_list,intents\_json):

tag= intents\_list[0]['intent']

list\_of\_intents =intents\_json['intents']

for i in list\_of\_intents:

if i['tag'] == tag:

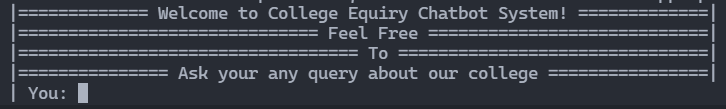
result = random.choice(i['responses'])

break

return result

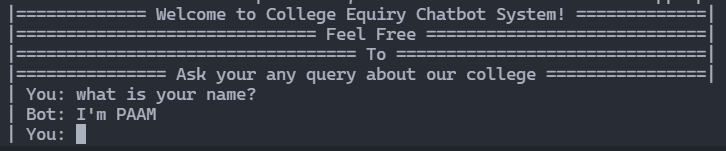
**6.2 Significant Project Outcomes:**

PAAM (chatbot) will represent in initially like this



**Fig. 6.1 PAAM (chatbot) Outcome**

And if we say like what is your name PAAM will give answer like



**Fig. 6.2 PAAM (chatbot) Introduction**

**6.3 Project Applicability in Real-World Applications:**

The College bot project is built using artificial algorithms that analyses user’s queries and understand user’s message. This System is a web application which provides answer to the query of the student. Students just have to query through the bot which is used for chatting. Students can chat using any format there is no specific format the user has to follow. The System uses built in artificial intelligence to answer the query. The answers are appropriate what the user queries. The User can query any college related activities through the system. The user does not have to personally go to the college for enquiry. The System analyses the question and then answers to the user. The system answers to the query as if it is answered by the person.

With the help of artificial intelligence, the system answers the query asked by the students. The system replies using an effective Graphical user interface which implies that as if a real person is talking to the user. The user just has to register himself to the system and has to login to the system. After login user can access to the various helping pages. Various helping pages has the bot through which the user can chat by asking queries related to college activities. The system replies to the user with the help of effective graphical user interface. The user can query about the college related activities through online with the help of this web application. The user can query college related activities such as date and timing of annual day, sports day, and other cultural activities. This system helps the student to be updated about the college activities.

**6.4 Inference:**

In this chapter, the word is tokenized and lemmatize got converted into bag of words. The words in the bag are now going to check in our dataset. If the dataset contains the words, it will return 1 and if not, it will return 0. we made a simple look interface for outcome and also welcoming the user. Lastly, we discussed the real-life applications and uses in real-world.

**CHAPTER-7:**

**CONCLUSIONS AND RECOMMENDATION**

**7.1 Limitation/Constraints of The System:**

* They Can’t Make Decisions:

Another limitation of chatbots is that they lack decision-making. They don’t have the right know-how to differentiate between the good and the bad. On March 23, 2016, the tech biggie Microsoft attracted many controversies due to its chatbot Tay. The chatbot posted offensive Tweets and landed Microsoft in huge troubles. So, they have to shut down the chatbot temporarily. Similarly, chatbots have done a lot of damage to multiple brands due to their poor decision-making capability.

* Chatbots Have the Same Answer for a Query:

Most customers don’t proceed with the chat when they know they are chatting with a chatbot. Chatbots are easily identifiable because they have the same answer for multiple queries. Suppose you are asking something to a bot that is not available in the data server so that you will get an apology. The same is the case with other queries. No matter how many different questions you ask, it will deliver you with the same apology, which is quite irritating.

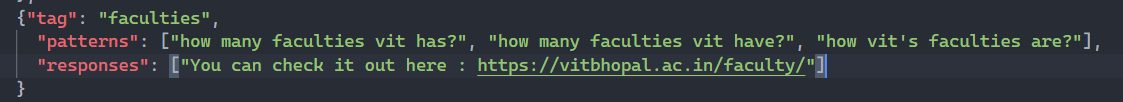
**7.2 Future Enhancements:**

We are trying to build self-learning PAAM. So that students will get accurate answers.



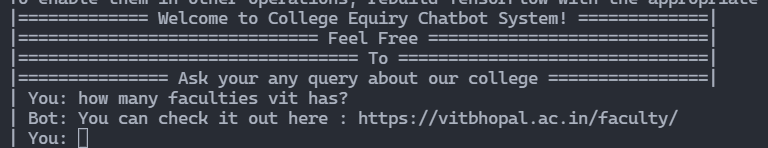
**Fig. 7.1 Problem of PAAM (chatbot)**

For that we have to give plenty amount of dataset to avoid errors.



**Fig. 7.2 Resolving Problem of PAAM (chatbot)**

Like this if we give the data to PAAM then it will recognize the data, not get confused and give accurate answer.



**Fig. 7.3 Resolved Problem of PAAM (chatbot)**

**7.3 Inference:**

Our main motive is to overcome all the errors and problems of PAAM (chatbot) so that all students are confused they will fulfil their questions.

The goal of the system is to help the students to stay updated with their college activities. Artificial Intelligent is the fastest growing technology everywhere in the world, with the help of Artificial Intelligent and Knowledgeable database. This system is developing chat bot based on android system so with the combination of Artificial Intelligent Knowledgeable database and virtual assistance. We can develop such chat bot which will make a conversion between human and machine and will satisfy the question raised by user. The main motive of the project is to reduce the work load on the college's office staff and reduce the response time to a user's query.

**REFERENCES**

1. Shin, K.G. and Mckay, N.D. (1984) ‘Open Loop Minimum Time Control of Mechanical Manipulations and its Applications’, Proc.Amer.Contr.Conf., San Diego, CA, pp. 1231-1236.
2. Mohammed Javed, P. Nagabhushan, B.B. Chaudhari, “A Direct Approach for Word and Character Segmentation in Run-Length Compressed Documents with an Application to Word Spotting”, 13th International Conference  on  Document  Analysis  and Recognition (ICDAR), 2015.
3. Naeun Lee, Kirak Kim, Taeseon Yoon, “Implementation of Robot Journalism by Programming Custombot using Tokenization and Custom Tagging”, 2017.
4. Sachin  S.  Gavankar,  Sudhirkumar  D.  Sawarkar,  “Eager Decision Tree”,  2nd International Conference  for Convergence in Technology (I2CT), 2017.
5. Liner Yang, Meishan Zhang, Yang Liu, Maosong Sun, Nan Yu, Guohong Fu, “Joint POS Tagging and Dependency Parsing with Transition-based Neural Networks”, 2018.
6. Jerome r. Bellagarda, “Parts-Of-Speech tagging by Latent Analogy”, IEEE Journal of Selected Topics in Signal Processing, Vol. 4, No. 6, 2010.
7. LinHua gao, HePing Chen, “An Automatic Extraction Method Based on Synonym Dictionary for Web Reptile Question and Answer” 2018.
8. Sijun Qin, Jia Song, Pengzhou Zang, Yue Tan, “Feature Selection for Text Classification Based on Parts-Of-Speech Filter and Synonym Merge”, 12th International Conference on Fuzzy Systems and Knowledge Discover (FSKD), 2015.
9. https://pdfs.semanticscholar.org/90a9/93a09479705f63147eea427e0e048964dbf0.pdf?\_ga=2.34781409.1975834925.1639895268-1134497788.1639895268

We took help from NeuralNine for our Chatbot code:

NeuralNine -

https://www.youtube.com/channel/UC8wZnXYK\_CGKlBcZp-GxYPA

NeuralNine website -

https://www.neuralnine.com/