



## Admission Portal Chatbot System

Abhinav, Akshay Mathur, Golda Dilip  
 Student, student, Associate Professor,  
 Department of Computer Science and Engineering(4th Year),  
 SRM Institute of Science and Technology, Vadapalani, Chennai,India

**Abstract**— Admission system in college nowadays is a great hassle to deal with, for the students as well as the college staff. After completing the secondary board exams, a huge number of students apply for admission in various colleges and it becomes hectic for them to reach every college and fetch the info about the college's admission process. The college staff as well need to answer many students and solve their queries. So to deal with this problem, we came up with a solution of creating a chatbot that can work on the college's admission portal and answer the queries of each student which will decrease the load on college staff as well as help students ease up in the process of applying for admission in a college. With the use of chatbot there would be zero doubt space since it would be a direct communication between the student and college as all the answers that the chatbot would come up with to answer the doubts would be fetched from college's admission portal database.

### I. INTRODUCTION

Humans have been constantly fascinated with auto operable Artificial Intelligence driven gadgets. And with the latest advancements and researches in this field, the programming has wound up more human-like on top of being automated[3]. The mix of immediate response reaction and consistent connection has made it an engaging change to web application trends.

In common and normal terms, a bot is nothing but a software that performs automated tasks. In other terms a bot is a software that communicates with human users through the internet[3]. The chatbot works in the same way as instant messaging where on the other side a machine works and process with analytical thinking rather than a human user. By artificially replicating the patterns of human speech and interactions using machine learning it learns how to respond when a human user inputs a query or a statement[2]. The interaction between the human and the machine can happen via text commands or voice commands.

A chatbot is designed to work independent of a human operator[1]. It can answer questions formulated to it in natural language and respond like a real human. It provides responses based on a combination of predefined scripts and machine learning applications[2]. When it is asked a question the chatbot will respond based on the knowledge database available to it at that point of time. If the conversation involves a concept it will deflect from the conversation or it will rather pass forward the conversation to the human administrator.

As we all know that the admission processes for college is very hectic, it is here where we realize the need of chatbot to answer the queries of the students and to reduce the load on college staff.

Admission portal chatbot system is a software that works on college's admission portal on the college's website. It requires the student to sign up before it becomes operable, this step helps in validating user and verifying if it is a human or a machine. After the sign-up a page loads where the queries could be entered. Here the students may use voice or text command to input the query, after which the chatbot responds accordingly. If there is an issue with the response the chatbot gives or if the database chatbot fetches responses from does not have any response for the particular query then the chatbot passes the conversation to the human admin using a call connection or a text connection.

Obvious advantages of admission portal chatbot system includes:

- i) More human interactive conversations.
- ii) Less load on college staff.
- iii) More accurate Info.
- iv) Less chances of doubt arousal.
- v) Constant upgradation of database so much rich human interactions[2]

## II. EXISTING SYSTEM

During admission times it is seen that college staff need to answer many students and have take single queries from them each at a time. The process is very long and tiresome

### 1) Architecture Description: -

Admission portal is a system where the students who have queries regarding admissions in college go to resolve their queries but the current way is a bit cumbersome because there are too many steps to the procedure of resolving the queries.

### 2) Methodology:-

Admission in colleges is a long and cumbersome process for both the college staffs and students. I the process of taking admissions in the college. There are too many queries that might arise in the student's mind and to resolve these queries, the colleges decide to design an admission portal in college's website where students can go and solve there queries but still this system is bit unrefined because there are too many steps involved to get a single query solved. First, the students need to go to the college's website and then they need to go to the admission portal and search for the type of their query and enter it. After which the query is forwarded to staff member through the portal and the staff member replies to the student. But in this process, there is no certain period under which the reply will be given to the student since there are too many students and there is too much of workload on a single staff member. As well as this facility does not work for 24x7 period. The colleges will also need too much of human resource to start a 24x7 facility.

### 3) Challenges of existing system:-

- The existing system is not transparent. There is no surefire way of checking if any false information is being circulated and students are being bluffed for financial gains.
- The current admission system is unable to provide 24x7 facility for students to inquire.
- The current admission process is a very expensive affair that requires heavy prior planning, organization, Time, Money of Students.
- Students right now often get confused and apply to colleges on a whim and regret their decision as even after trying to fetch right info they can't get into college of their choice.
- Since admission is a very cumbersome procedure so it requires time and human labour as to deal with all the student's queries.

## III. PROPOSED SYSTEM

The primary aim behind the implementation of chatbot into the current admission portal is to ease up the whole process of admission of students and to decrease the load on admission staff. Some of the extra goals that can be achieved are like removal of false information from that the third might use for cash benefit from students during their admission process. Thus, this will make the whole process more secure, smooth and doubt-free.

### 1) Participants:

- *Students:-* The students who have queries regarding admission procedure and who will login to admission portal to resolve their queries.
- *Chatbot interface:-* The chatbot that will interact with the students about their queries and take their queries as an input and provide the solution by fetching results from database by using mapping respectively.
- *Human admin:-* The person who controls the backend of the chatbot system and deals with the queries that AI cannot map with the results in database and passes it back to student using the admission portal.

### 2) Modules:

- *Personal Query Response System:-* Upon receiving personal query, the authenticity of the user is checked through user-id and password. If the user detail is invalid, an appropriate response is sent. If the user authenticates successfully, the input text is processed to extract keywords. Based on the AI chatbot keywords, information required by the user is understood and the information is provided from the database.
- *AIML Response System:-* This module will work If the user is trying to make a normal conversation with the bot, the input is mapped to an appropriate pattern in Artificial Intelligence Modeling Language (AIML) files. If the response is available, it is sent to the user. Other data provided to the chatbot such as username, gender, etc. are also saved. If the pattern is not available in AIML files, a random response is sent suggesting "Invalid Input" or the conversation is forwarded to a human operator.
- *Query Analysis and Response System:-* This module is for When a user wants some information pertaining to query, the response will be provided through this module. If the input matches a pattern in the AIML files, the appropriate response will be sent to the user. If the AIML

files have no entry for that particular query pattern, keywords are fetched from the input. An algorithm to check sentence similarity (NLP) is applied to the modified input to check its similarity with the questions of a predefined question-set, whose answers are available[4]. If a sentence is retrieved with confidence > 0.5, we return the answer of that question as the response. If no questions map to the user input, the input is saved in a log file for improvement of the system by the admin. The administrator can incorporate the answer to that query.

#### IV. SYSTEM ARCHITECTURE

The system architecture is used to explain the working of the software or module according to its principles, concepts, and characteristics which are logically related and consistent with each other. The design of the approach has the attributes, properties, and functions that solve as many challenges as possible or the potential provided by the job method and the definition of the life cycle and applied by the technologies.

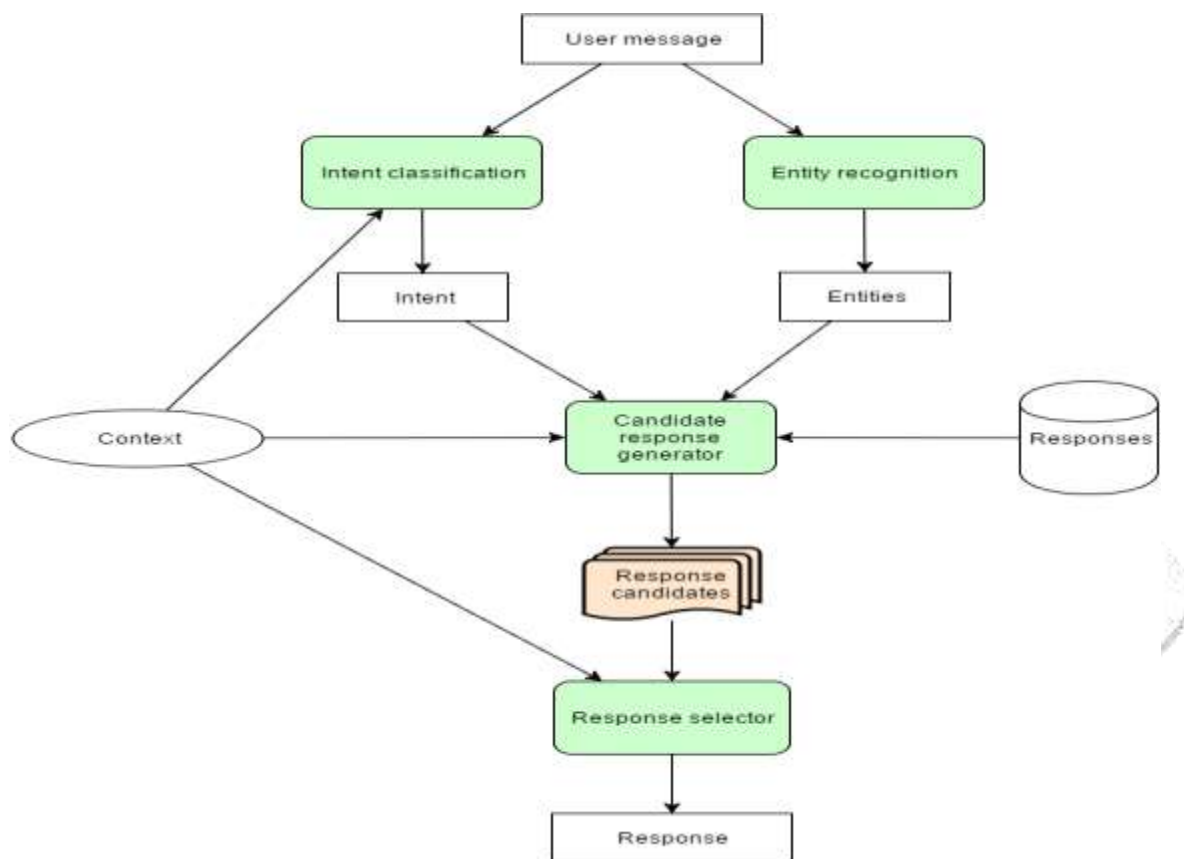


Fig 1: Architecture of AI chatbot based admission portal system

This is an abstract, conceptual-based, regional, and program-oriented word to attain the goal and work-life span of the method. System architecture also concentrates on the high-end structure in the system and system elements. One architecture can be used for representing the common structures, pattern and set of requirements for similar classes and families.

Referring to Fig. 1, We can see that the framework mentioned in this paper is about the User, Messaging channel, chatbot API, Natural language processor where the user first login into the page and then enter the admission portal where he/she enters the query and submits it as an input. After the query has been input the chatbot takes the query and checks the intent and entities using NLP and matches with the response DB and responses to the user with an accurate result. If there is no response present for the query than the chatbot passes the conversation to the human admin who handles the backend.

#### V. IMPLEMENTATION

For the implementation, we have used Visual Studio code as a platform to run Html, Css, javascript and python codes along with Nltk library and Aiml file. We are using python programming language to convert voice to text and we use javascript code to validate the microphone for listening the voice. To integrate the web-based application with the Chatbot we are using Visual studio code, python library Nltk and Aiml algorithm. The visual Studio output has been shown in Fig.2 . The code snippets can be seen in Fig.3 .



Fig. 2: Output with college fee information

```

D:\> npm list (C:\Users\Pratik > # main.js) -
1  const voice = document.querySelector(".voice");
2  const voiceText = document.querySelector(".voiceText");
3
4  const SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;
5  const recorder = new SpeechRecognition();
6
7  function addHumanText(text) {
8    const chatContainer = document.createElement("div");
9    chatContainer.classList.add("chat-container");
10   const chatBox = document.createElement("p");
11   chatBox.classList.add("voiceText");
12   const chatText = document.createTextNode(text);
13   chatBox.appendChild(chatText);
14   chatContainer.appendChild(chatBox);
15   return chatContainer;
16 }
17
18 function addBotText(text) {
19   const chatContainer1 = document.createElement("div");
20   chatContainer1.classList.add("chat-container");
21   chatContainer1.classList.add("darken");
22   const chatBox1 = document.createElement("p");
23   chatBox1.classList.add("voiceText");
24   const chatText1 = document.createTextNode(text);
25   chatBox1.appendChild(chatText1);
26   chatContainer1.appendChild(chatBox1);
27   return chatContainer1;
28 }
29
30 function botVoice(message) {
31   const speech = new SpeechSynthesisUtterance();
32   speech.text = "Sorry, I get lost understand that.";
33 }

```

Fig. 3: Code Snippets

## VI. FUTURE ENHANCEMENTS

Considering the current situation of admission systems the chatbot system that has been made as a prototype is enough to handle the workload on query office but in future more enhancements could be made to the chatbot where the chatbot would be able to handle not only the admission processes in the college but other several work processes, further they would not only be able to give responses to the user but they would also be able to analyse the query of the user in much depth and and redirect the user to the page where the user needs to go to and execute something, this would be great advantage to the college since most of the college offline processes are being turned to online and with digitalization so the chatbot would be able to run on multiple portals and ease up the workload.

## VII. CONCLUSION

As described earlier, the chatbot that has been made for the college admission portal and is still in its development phase. There are many more enhancements that can be made to the chatbot. Further, this chatbot has been designed in a way that can be used in many other websites, to use on a different website, only the queries need to be changed. If this chatbot is developed further, it will surely ease up the load on the college staff as well as the students.

## VIII. REFERENCES

- [1] "Anon, (2016)". [online] Available at: [https://www.chatbots.org/ai\\_zone/](https://www.chatbots.org/ai_zone/) [Accessed 11 Dec. 2016].
- [2] Foundation, A. (2016). "AIML - The Artificial Intelligence Markup Language." [online] Alicebot.org. Available at: <http://www.alicebot.org/aiml.html> [Accessed 11 Dec. 2016].
- [3] Vignesh Jayapalan, H. (2016). "Android — Baking a simple ChatBot in 30 minutes (AIML)." [online] Medium. Available at: <https://medium.com/@harivigneshjayapalan/android>
- [4] Wallace, R. (2016). "AIML 2.0 Working Draft." [online] Docs.google.com. Available at: <https://docs.google.com/document/d/1wNT25hJRyupcG51aO89UcQEiGHkXRXusukADpFnDs4/pub> [Accessed 11 Dec. 2016].

