

ADMISSION CHATBOT

PROJECT FOR @ZENSE RECRUITMENT 2020

Details:

Name	Roll Number	Project Name
Vijay	IMT2019525	Admission Chatbot

Origin of this Project:

Innumerable students are writing many entrance exams and seeking admission to the premier engineering institutions of the country. However, more often than not, they are overwhelmed by the intricacies of this complex process. Every college seems to have its criteria for joining and additionally, the students have to consider other factors like distance, hostel, fees and amenities, among others. This project is an attempt to make the process a little more streamlined; the current generation of people want information and they want it as fast as possible. This chatbot gives them information regarding the college and its admission process, so they can clear at least some of their doubts in a self-sufficient manner.

Proposed Idea:

This project involves a basic chatbot that revolves around the concepts of *neural networking*, *text-to-speech* and *speech-to-text*. Through this project, prospective students can communicate with the *bot* about admission-related queries. The computer will predict the domain of the query and answer accordingly.

Making Admission stress free, one conversation at a time,

makes for a compelling case, doesn't it? Indeed, I firmly believe that, with the further refinement, extension, and deployment of this concept, this could become a reality.

Tech Stack:

The front-end(user interface) is the *command line interface* i.e the terminal. The back-end data is stored in .pickle and .tflearn files, note that there is no physical RDBMS software used directly. The following modules are used in this program: 1. For Speaking and Language Processing: 1.1 `nltk` : This is used for *stemming* the words(converting a string into its *root word*) 1.2 `SpeechRecognition` : This is used to convert the user's voice signal into text 1.3 `pyttsx3` : This is used to convert text to speech signal, i.e "computer's voice"

2. Neural Network Aspect: 2.1 `tflearn` : Input data is processed and assigned a probability of it belonging to a particular *label*/based on the sample cases provided 2.2 `numpy` : The data is processed using numpy arrays for the segregation of features,labels and combinations.
3. Storing the data and test cases: 3.1 `json` : The data we provide the model to *learn* from is outlined systematically in the .json file, which we can manipulate through *json*. 3.2 `pickle` : The words, labels, training and output data are stored in .pickle file, which we can manipulate through *pickle* module.

Progress of the Project:

So far, a rudimentary version of this project having all functionalities mentioned above, is completed. A .json file containing *patterns* and *responses* of five labels are included, however, it is expandable according to the needs of the institution. The `webbrowser` aspect is yet to be implemented, along with a GUI using `tkinter` , if possible.

You can find the project [here](#)

Future Prospects:

The sky is the limit, with this project: * One can deploy this in a college website * One can also deploy this on Whatsapp, Discord, etc. * One can change the theme of this chatBot by simply altering the .json file

With the advent of Artificial Intelligence, this project serves as an example of communication with machines, for a cause; it also becomes more relevant.

References:

The following sources proved to be *very* helpful:

`pyttsx3` `speech_recognition` `tflearn` `markdown`

Additionally, YouTube and StackOverflow helped a lot.

What I learnt from this project:

Through this project; I learnt a lot, and developed a liking for software development, particularly in the Website and Machine Learning domains. Through the seemingly inevitable installation glitches and bugs in code, I understood the importance of YouTube; it baffles me that technology has advanced so far, that a person who might as well be on the other side of the world goes out of his way and uploads a video that helps to fix installation errors, which helps another aspiring developer fix *his* code. Everyone, collectively, makes everyone better off; something beautifully demonstrated by another thing that I have come to love- *github*.

Vijay Jaisankar

IMT2019525