An Intelligent College Enquiry Bot using NLP and Deep Learning based techniques

Conference Paper · January 2022

DOI: 10.1109/ICONAT53423.2022.9725865

7 authors, including:

A. Kousar Nikhath
K L University
16 PUBLICATIONS **72** CITATIONS
SEE PROFILE

Vijaya Saraswathi Redrowthu Ph.D
VNR Vignana Jyothi Institute of Engineering & Technology
27 PUBLICATIONS **203** CITATIONS
SEE PROFILE

An Intelligent College Enquiry Bot using NLP and Deep Learning based techniques

A Kousar Nikhath [1]
Associate Professor
Computer Science and Engineering
VNR VJIET
Hyderabad, India
kousarnikhath@vnrvjiet.in

Vijaya Saraswathi R [2]
Assistant Professor
Computer Science and Engineering
VNR VJIET
Hyderabad, India
vijayasaraswathi_r@vnrvjiet.in

MD Abdul Rab [3]
B-Tech Graduate
Computer Science and Engineering
VNR VJIET
Hyderabad, India
mohdrab01@gmail.com

N Venkata Bharadwaja [4]
B-Tech Graduate
Computer Science and Engineering
VNR VJIET
Hyderabad, India
bharadwajlive@gmail.com

L Goutham Reddy [5]
B-Tech Graduate
Computer Science and Engineering
VNR VJIET
Hyderabad, India
gouthamlyagala@gmail.com

K Saicharan [6]
B-Tech Graduate
Computer Science and Engineering
VNR VJIET
Hyderabad, India
saicharank018@gmail.com

C Venkat Manas Reddy [7]
B-Tech Graduate
Computer Science and Engineering
VNR VJIET
Hyderabad, India
rmanas244@gmail.com

***Abstract*— Every year freshers and their parents do visit our college website in order to get their queries clarified. Also, college students do visit the website in order to get their queries resolved. Thus we felt the need for an 'Intelligent Enquiry Bot' to be associated with the official college website. The Bot should be intelligent enough to resolve the queries of freshers, parents, students, and faculty. The college enquiry chatbot is designed using certain algorithms which understands and analyzes the user queries. This System is basically a web application that provides valid responses to the various queries of the users, which will make use of Natural Language Processing (NLP) and Long Short Term Memory (LSTM) networks, which are a special kind of recurrent Deep Neural Networks (DNN). In this paper, we have completed building a quite intellegent chatbot based on NLP and DNN for basic college-related enquiries and admission related queries especially.**

***Keywords*— Chatbot, NLP, Deep Learning, RNN, LSTM.**

I. INTRODUCTION

Chatbot is an intelligent software program that interacts with humans[6]. A chatbot works similar to human-like conversations on chat. Its primary task is to help users by providing answers to their questions by understanding what human wants and guides them to their desired outcome. Nowadays, various chatbots are responsible to solve a number of business related tasks in order to improve customer experiences across many industries like Insurance, E-Commerce, Banking, Healthcare, and many others[3].

A Deep Learning chatbot uses Natural Language Procesing (NLP) in order to map user inputs to some intents. It will classify the messages to send a prepared response. Thus, using deep learning and natural language processing, the chatbot becomes an intelligent software piece that enables it to process, comprehend and as well response using the natural language understanding. Usually, we use special RNNs called LSTMs to build a chatbot.

When using NLP to develop a chatbot, the main thing one should achieve is to create a chatbot that requires very little or say no human interaction at all[1]. However, it is tough to improve answers and selecting best model to guarantee the most relevant response in the field of chatbots.

The Aim of taking up this project is to provide a chat-bot system that deals with academic activities like inquiring about admissions, fees structure, getting details of departments, etc. And using this chat-bot system, the freshers, students and faculty can directly clear their queries in lesser time.

To give a roadmap of this paper, in section 1 there is an introduction about the paper, then it consists of literary survey in section 2, the methodology in section 3, followed by the system design in section 4, implementation & testing in section 5, later there is results & discussion in section 6 and finally in section 7 there is conclusion & future scope.

II. RELATED WORK

According to Ms. Ch. Lavanya Susanna Et al. A Student chatbot project is developed with the help of a code igniter which is widely called as a php framework. It analyzes the user queries and also perceives user messages. [1]. As per Prof. Ram Manoj Sharma Et al. proposal, a college enquiry chatbot system that was made using AI (Artificial Intellegence) algorithms and included few modules like Online chatbot, Online Noticeboards, etc. [2]. P.Nikhila Et al. have collectively designed a chatbot using AIML (Artificial Intelligence Mark-up Language) to make a response to user queries. Here, to customize the Alice bot that could be a chat-bot application supported Alice free code the AIML was used [3].

Payal Jain Et al. has built up a database that consisted of all the related information and also created a web interface (UI) that has two sections. One of them was for basic clients and another was for the admin [4].Authors Sagar Pawar, Omkar Rane have developed a UI for which the users have to register before accessing the chatbot. This paper [5] uses the Bigram, sentence similarity score, and retrieval of the matched template to give the response. As per Harsh Pawar Et al.[6] a chatbot was designed by them using the database knowledge. Their proposed system had online enquiry and chatbot system. They used various programming languages in development. They created a user-friendly graphical user interface to send and receive user responses. In another paper [7], chatbot designed was a

hospital appointment system whose goal was to get a doctor's appointment. That system used a end-to-end gated memory networks model, which was an end-to-end supervised model with automatically learned gating mechanism in order to perform dynamic regulation of the memory interaction.

An artificial chatbbot proposed by Nitesh Thakur Et al.[10] used NLP (Natural Language Processing) for making chatbot. It could be done in two ways, namely written text and verbal or voice communication. However, written communication is easy than verbal communication. This paper had some emerging capabilities for increasing the speed, and processing in virtual human interaction systems. Another chatbot developed by Amit Tiwari Et al. [11] made use of artificial intellegence based algorithms to analyse users queries and understand their message. Its answered appropriately to the user queries. If the answers were found to be invalid, the user just needed to select the invalid answer button which would notify the admin about it. Admin could then access a portal to view invalid answers and delete them or to add a specific answer to that equivalent question. Nidhi Sharma Et al. [14] have developed a website and has three modules front end, chatbot, the backend is admin login and chatbot is a college enquiry bot that provides the information regarding fee structure of the different courses using NLP, pattern matching.

In a paper,[15] Question Answering Systems were used as information access systems in order to try to answer NLP based queries. This type of system selects the most appropriate answers by making use of linguistic features available in the NLP techniques. A paper [16] proposed another approach to identify the facts that are most important in the texts. It consisted of a method to build a chatbot which could simulate some historical figure and which could answer all kinds of questions about the life experience of that figure. Another paper used a sentence-similarity score [17] to obtain similarity between both the input pattern and the output pattern. This process was done in the Relational Database Management Systems (RDBMS). This chatbot consisted of a core and an interface which is accessing that core. And the database had tables to store the knowledge, and the interpreter is a stored program of function and procedure sets in order to require the pattern matching.

III. PROPOSED SYSTEM

While building this intelligent college enquiry bot system, we have followed the below mentioned steps.

A. Building front-end
B. Building back-end
C. Creating the proposed chatbot model
D. Connecting the front-end to back-end

The below is a brief discussion on each of the steps followed in the mentioned methodology:

A. Building Front End

We used ReactJS to build the front-end and Python as a back-end. The ML-Model for chatbot was build and incorporated into the backend. Coming to the front-end the components mentioned in the hierarchy below were used to build the UI for the chatbot. Moreover, the UI is also responsive for the different screen sizes.

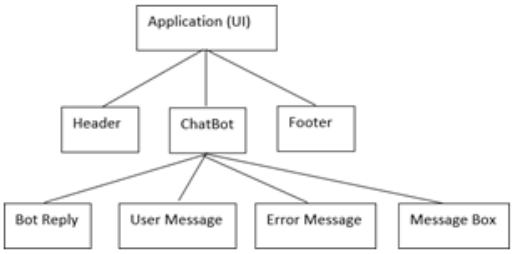

Fig.1 Frontend Modules Hierarchy

As shown in the above "Fig. 1", there are three main components for the React App:

a) **Header Component** – which displays the college's name and logo as a header

b) **Footer Component** – which displays a footer for the webpage which usually has a copyright

c) **Chatbot Component** – which is the main component for this project. It is further subdivided into 4 other components as mentioned below. This component is responsible for the rendering of the chatbot's UI for the end-users. Its sub-components are:

i. **UserMessage Component** – which is responsible for displayingthe user queries that are entered by the user in the message box provided after sending to the chatbot

ii. **BotReplyComponent** – which is responsible for displaying the replies of the chatbot for the queries made by the user after getting a response from the backend

iii. **MessageBox Component** – which provides an input box where the user can write their queries and send the message to the chatbot by clicking a send button provided by the side of it.

iv. **ErrorMessage Component** – whichdisplays the various error messages captured in different circumstances and notifies the user about it by error text.

B. Building Back -end

Then speaking of the backend, we have written several methods for several specific functionalities. The below are a few modules and their descriptions we had used in our backend.

a) **Intents Module –** which is the data file containing all the predefined patterns and their corresponding responses. It is stored in the form of a JSON (JavaScript Object Notation) file.

b) **Preprocessing Module –** which consists of the preprocessing methods in order to clean the sentences and perform NLP techniques like tokenization and lemmatization.

c) **Chatbot Training Module –** which is responsible for using preprocessing module methods to clean the data and split the data into train and test sets. It also involves creating, training and saving the deep learning model

d) **Model Related Module –** which is basically a wrapper to help the chatbot training module to deal with the deep learning model and perform operations on it. In it there is contained the information regarding the model and also the weights of the neurons.

e) **Chatbot Module –** It can be called the main module that is responsible for connecting the chatbot to the Flask app. It serves as a backend which receives user query from UI and sends back the bot's responses through REST APIs.

C. Creating the proposed chatbot model

It is basically a retrieval-based chatbot making use of technologies like NLTK, Keras, Python, etc. In order to compile the model we used the stochastic gradient descent (SGD) along with nesterov accelerated gradient as an optimizer and also it gave good results for this model.

The below are the five steps followed in order to create a chatbot in Python:

a) **Importing and loading libraries**

The first and foremost step in the process of creating the chatbot is importing the necessary packages and initializing necessary variables. Below mentioned are a few important required packages:

```
Flask==2.0.1
Keras==2.6.0
Nltk==2.6.0
Numpy==1.19.5
Tensorflow==2.6.0
Tflearn==0.5.0
```

b) **Preprocess data**

Pre-processing the data is yet an important step for an accurate response from the chatbot. When we are working with textdata, we should use certain pre-processing techniques and perform them on the data before building a machine earning or deep learning model.

A few pre-processing techniques used are:

i. **Tokenization** – where it is the process of breaking down the complete input text (i.e., user query) into smaller parts i.e., words. Then we iterate through the patterns that are provided in the intents file and then tokenize the sentences using "nltk.word_tokenize()" function. After that, we append every word contained in the words list to create a classes list for the tags in the intents file.

ii. **Lemmatization** – This is a process of converting a word to its lemma (root) form. We also then create a pickle file to store the python objects which we used while predicting the responses. It generally finds the root word from the given word.

c) **Creating training and testing data**

As a next step, we create the training and testing data by splitting the available data. Our input data is the patterns and output data is the classes to which the pattern belongs. As the computer couldn't understand textdata, we will convert the text into numbers.

d) **Building the deep learning model**

As we now have the data i.e., both training and testing data ready, we now concentrate on building a Deep Neural Network model. As far as speaking about the Deep Learning model, we created a model consisting of 3 layers as follows:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 128)               14720
_________________________________________________________________
dropout (Dropout)            (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                8256
_________________________________________________________________
dropout_1 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_2 (Dense)              (None, 11)                715
=================================================================
Total params: 23,691
Trainable params: 23,691
Non-trainable params: 0
```

Fig. 2 Model summary

Thus we trained the chatbot on the dataset containing categories (intents), patterns and responses. We made use of 'Long Short Term Memory' (LSTM), a special 'Recurrent Neural Network' (RNN) in order to classify the user's message and then we provide some random response from the list of predefined responses. We used Keras Sequential API for doing this. And as mentioned earlier, the model is saved as "chatbot_model.h5" file for future purposes. The summary of saved model is as shown in "Fig. 2".

e) **Predict the response**

Now, the final phase of a chatbot is to make prediction of the response. Now, we load the saved model and use React App User Interface in order to predict the response and display the same. The model only tells us the class to which it belongs, then we make use of a few helper functions to identify the class label and retrieve a random response from

the available list of responses corresponding to the input pattern. Also here we import "words.pkl" and "Classes.pkl" pickle files that were created during the model training.

In the below "Fig. 3", there is the architecture of the system design.

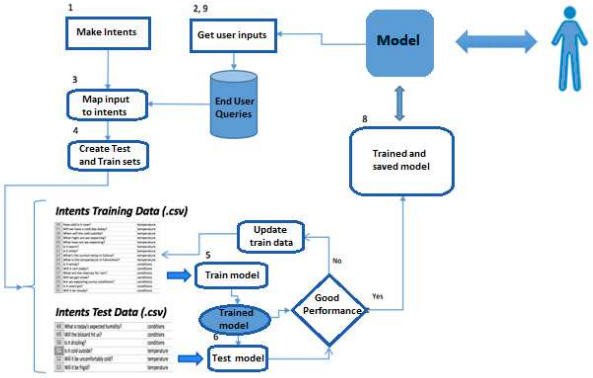

Fig. 3 System architecture

D. Connecting front-end to the back-end

This is again a crucial part of the chatbot application. Whenever a user enters a query through the message box provided in the UI, we send a POST request to a backend API. We provide the user query as a payload in the header contents of the POST request.

At the backend, whenever we receive a user query at the endpoint through a POST API request, we first extract the query from the payload in the header part of the request received. Then, with the help of a trained deep learning model, we get the response for the corresponding user query. This process involves some pre-processing of the received payload and then predicting the class label. Finally, we get an appropriate response, which is then appended to a chats list at the backend. Then, the chats list is sent back to the frontend as a JSON object.

At last, in the frontend, we try to capture any possible errors or exceptions if any, and send the received chats list to the UI in order to update the chats. This is how the frontend and backend are connected.

The workflow of the chatbot is depicted in the below figure.

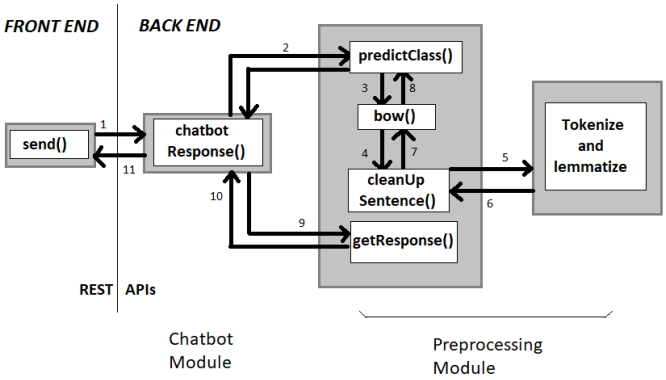

Fig. 4 Workflow

From the above "Fig. 4", you can identify the flow of control for our chatbot once a user enters a query from the UI.

IV. IMPLEMENTATION & TESTING

After the chatbot was build successfully and the UI is connected to the back-end the chatbot looked like you can see in the below "Fig. 5" & "Fig. 6".

Fig. 5 Chatbot App – large screens

The UI is also made responsive to different screen sizes and it looks pretty well and user-friendly too.
Below is how the UI looks when concerned with small screens of mobile phones.

Fig. 6 Chatbot App – small screens

Not only we had just built the development code, but also we had written a few test-cases to check whether the expected functionality is achieved in most of the circumstances.

We have written a few test-cases to ensure that expected functionality of the app is retained. For this, we have used Jest framework to test the React front-end and Pytest and Unittest libraries to test the Python back-end.

The back-end test cases are written for the API requests being sent from UI to backend and receiving the proper response from backend back to the UI.

Below "Fig. 7" & "Fig. 8" are a few snapshots of the checking those test-cases:

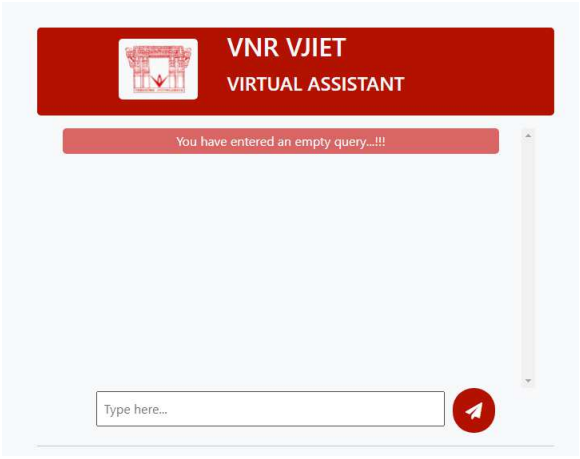

Fig. 7 Testcase- user enters empty query

Fig. 8 Testcase- backend is offline

V. RESULTS & DISCUSSION

The React UI's Front-end and Python Back-end unit-testing reports are as follows:

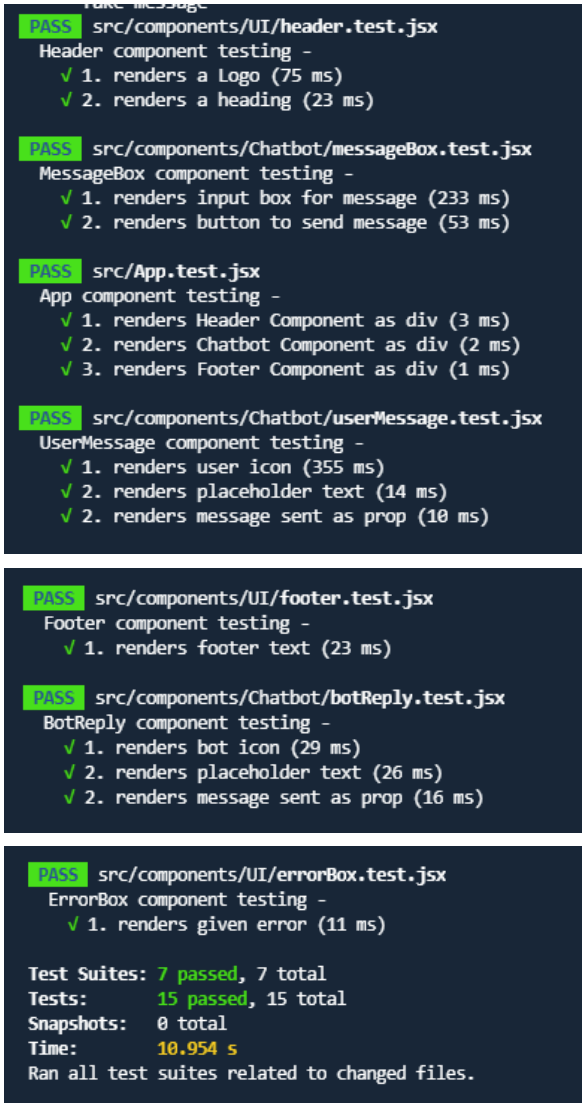

Fig. 9 React UI Testing Report

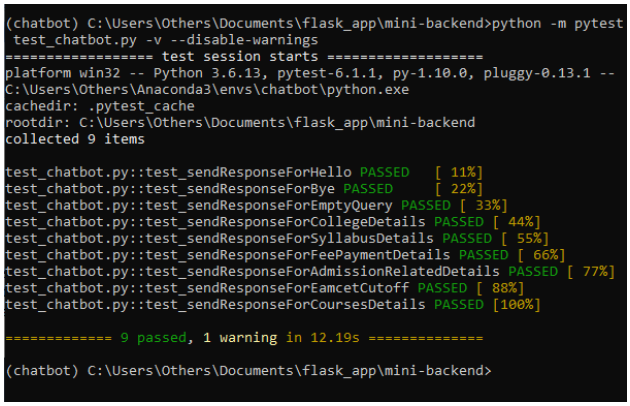

Fig. 10 Flask App Testing Report

"Fig. 9" & "Fig. 10" represent the testing reports for the front-end and back-end respectively.

The deep learning model we have used was trained to be very accurate in returning appropriate responses to the user queries. It almost achieved an accuracy of 99% after training it for about 200 epochs.

The below "Fig. 11" consists of few plots depicting the model's accuracy and loss based on the training and validation data sets. As you can see, after 200 epochs, the accuracy has been maximized whereas loss has been minimized.

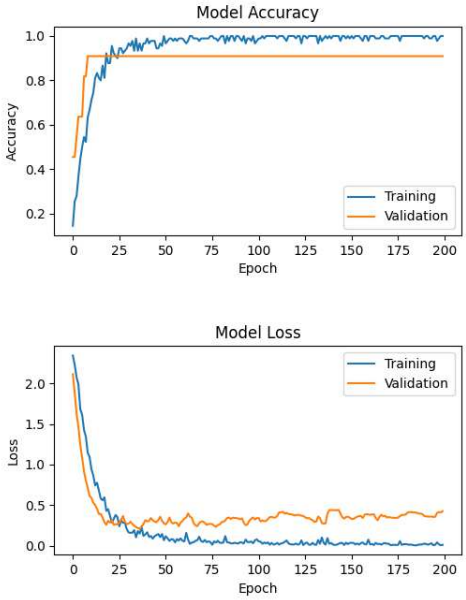

Fig. 11 Model Accuracy & Loss

VI. CONCLUSION & FUTURE SCOPE

The goal of our proposed system is to help the students to get information about their college activities and to post their admission-related queries on the go from anywhere, even outside the college. Another main motive is to reduce the workload on the college staff and reduce the response time for a user queries. For this, we have proposed a web-based chatbot system with the combination of Deep Leaning and NLP based techniques. It had almost 99% accuracy score in giving appropriate responses to the users for their queries. The performance as well as accuracy is very considerable for our chatbot system along with very small response time.

In future, using AI/ML/DL, chatbots can provide students with learning material in an interactive manner on any topic, help them learn quicker through visuals, speech or video and evaluate their responses to gauge their learning. They can be used for –

a. **Admission process** – To assist through documentation guidelines, enrollment procedures, campus info, generate more inquiries, etc.
b. **Recommending courses** – A personalized assistance to students on courses offered and resolving queries on curriculum, credits, internship opportunities, etc.
c. **On boarding students** – FAQ resolution on orientations, campus visits,events, etc.
d. **Student/Faculty Support point** – to offload the staff's work of solving repetitive queries.

Student/Faculty Feedback - Collect students' feedback through conversations on learning sessions, campus environment, course improvement, etc.

ACKNOWLEDGMENT

This research was supported by VNR VJIET. We are thankful to our colleagues who provided expertise that greatly assisted the research, although they may not agree with all of the interpretations provided in this paper.

We are also grateful to our Mini Project Guide, Dr. A Kousar Nikhath Maam, for her assistance, and who moderated this paper and in that line improved the manuscript significantly.

REFERENCES

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

[2] Assistant Prof Ram Manoj Sharma, Chatbot based College Information System", RESEARCHREVIEW International Journal of Multidisciplinary, ISSN: 2455-3085 (Online), Volume-04, Issue03, March-2019, pp 109-112.

[3] P.Nikhila, G.Jyothi, K.Mounika, Mr. C Kishor Kumar Reddy and Dr. B V Ramana Murthy on , "Chatbots Using Artificial Intelligence", International Journal of Research and Development,Volume VIII, Issue I, January/2019,ISSN NO:2236- 6124, pp1-12.

[4] Payal Jain, "College Enquiry ChatBot Using Iterative Model", International Journal of Scientific Engineering and p 80-8Research (IJSER),ISSN (Online): 2347-3878, Volume 7 Issue 1, January 2019, p3

[5] Sagar Pawar, Omkar Rane, OjasWankhade, Pradnya Mehta, "A Web Based College Enquiry Chatbot with Results", International Journal of Innovative Research in Science, Engineering and Technology, ISSN(Online): 2319- 8753, ISSN (Print): 2347-6710, Vol. 7, Issue 4, April 2018,pp 3874-3880.

[6] Prof.K.Bala, Mukesh Kumar ,SayaliHulawale, Sahil Pandita, "Chatbot For College Management System Using A.I", International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056, p-ISSN: 2395- 0072, Volume: 04 Issue: 11 | Nov -2017, pp2030-2033.

[7] Jincy Susan Thomas, Seena Thomas, "Chatbot Using Gated End-to-End Memory Networks", International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056, p-ISSN: 2395-0072, Volume: 05 Issue: 03 Mar 2018, pp 3730- 3735.

[8] Prof. Suprita Das, Prof. Ela Kumar, "Determining Accuracy of Chatbot by applying Algorithm Design and Defined process", 4th International Conference on Computing Communication and Automation (ICCCA), 2018, 978-1-5386-6947-1/18/2018 IEEE, pp 1-6.

[9] Prof.K.Bala, Mukesh Kumar ,SayaliHulawale, Sahil Pandita, "Chatbot For College Management System Using A.I", International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056, p-ISSN: 2395-0072, Volume: 04 Issue: 11 | Nov -2017, pp 2030-2033.

[10] Nitesh Thakur, AkshayHiwrale, Sourabh Selote, Abhijeet Shinde and Prof. Namrata Mahakalkar, "Artificially Intelligent Chatbot", Universal Research Reports , ISSN : 2348 – 5612, Volume : 04 , Issue : 06, July - September 2017, pp 43-47.

[11] Amey Tiwari, Rahul Talekar, Prof.S.M.Patil, "College Information Chat Bot System", International Journal of Engineering Research and General Science, ISSN 2091-2730, Volume 5, Issue 2, March-April, 2017, pp 131-137.

[12] Malusare Sonali Anil, Kolpe Monika Dilip, Bathe Pooja Prashant, "Online Chatting System for College Enquiry using Knowledgeable Database", Savitribai Phule Pune University, Preliminary Project Report, pp 1-53,2017.

[13] Balbir Singh Bani, Ajay Pratap Singh, "College Enquiry Chatbot Using A.L.I.C.E (Artificial Linguistic Internet Computer Entity)", International Journal of New Technology and Research (IJNTR), ISSN:2454-4116, Volume-3, Issue-1, January 2017 Pages 64-65.

[14] Ms.Ch.Lavanya Susanna, R.Pratyusha, P.Swathi, P.Rishi Krishna, V.Sai Pradeep, "College Enquiry Chatbot", International Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395-

0056, p-ISSN: 2395-0072, Volume: 07 Issue: 3 Mar 2020 pp 784-788.

[15] Agnese Augello, Giovanni Pilato, Alberto Machi' ICAR - Istituto di Calcolo e Reti ad Alte Prestazioni CNR - Consiglio Nazionale elleRicercheVialedelleScienze , 978-0-7695-4859-3/12 $26.00 © 2012 IEEE . "An Approach to Enhance Chatbot Semantic Power and Maintainability: Experiences within the FRASI Project"

[16] Emanuela Haller, Traian Rebedea Faculty of Automatic Control and Computers university Politehnica of Bucharest, 978-0-7695-4980-4/13 $26.00 © 2013 IEEE. "Designing a Chat-bot that Simulates a Historical Figure".

[17] BayuSetiaji, Ferry Wahyu Wibowo , Department of Informatics Engineering STMIK AMIKOM Yogyakarta, Yogyakarta, Indonesia, 2166-0670/16 $31.00 © 2016 IEEE "Chatbot Using A Knowledge in Database-Human-to-Machine Conversation Modeling".