

# Admission BOT: Unleashing the Potential of Neural Networks in Streamlining the Admission Process

Mahendra Uttam Dabhade<sup>1</sup>, Prof D.V. Thombre<sup>2</sup>

<sup>1</sup>Research Student, Computer Engineering, Terna Engineering college, Navi Mumbai

<sup>2</sup>Professor, Dept. of Computer Engineering, Terna Engineering college, Navi Mumbai

## Abstract:

This paper discusses the problems identified in the admission guidance process conducted by the Government Agency where students, parents, and other stakeholders of admission do not understand complicated language used in the policy documents and are misinformed by the lack of knowledge. This problem can be reduced by the creation and development of an admission Auto-Reply Bot designed to reduce human efforts and provide automated solutions to queries related to the admission process.

For this dataset, questions and answers are created using information brochure data available in the public domain which is conducted by one of the government agencies [4]. Different NLP techniques such as tokenization, lemmatization, and stemming are applied to the data, which is later stored in JSON. An artificial neural network is used to train this dataset. An ANN model is trained, and the chatbot performed well with the selected question set, with the accuracy of the model increased to 0.9918 using hyperparameter adjustments such as dropout and Adam optimizer.

**Keywords-** Chatbot, Natural Language Processing, Artificial Intelligence, Deep Neural Network, Artificial Neural Network, Admission Queries, Feed forward Network, Admission

## 1. Introduction

Chatbot systems are utilized in industry for providing services to solve customer queries and information collection. The admission process is a complicated process, and an admission system guidance chatbot can serve as a guide and offer solutions to students who want to take admission through the mentioned process. The development of this chatbot aimed to address several existing problems in the admission process conducted by government agencies, including limited access to the knowledge of the admission process, as users most of the time visit the expert people in the field personally or spend time browsing the information. While browsing, most of the time they are landing on incomplete or irrelevant information websites [3]. This chatbot is made to serve as a solution to queries of students and parents or institutes related to admission processes. They can also get information available on legitimate sites. More information can be available for institutes, courses, and intake availability, hostel availability, information related to any college. A chatbot is software that communicates with humans using natural language processing. Nowadays, chatbots are used in almost every field. In the education domain, for teaching and learning process, a model for chatbot is used [9].

The purpose of this paper discusses implementation of the Admissions Auto-Reply Bot, which can be able to provide answers to queries arises during the Admission process. The chatbot is using the Feed-Forward Neural Network, used to process queries and inputs to provide the best able information. Section 2 explains how model is created and what are the steps used for implementing the Admission auto-reply bot application, Section 3 explains the process of creation of dataset for the admission related queries and the major libraries used in python for development of chatbot. Section 4 explains input dataset (data supplied to the model) used for the application. It consists of user questions, answers dataset & pre-processing steps. Section 5 Gives the details about model implementation steps and associated process. Section 6 provides summary on the results obtained while exploiting the data in the testing phase. Finally, summarised using conclusion and future scope sections.

Table 1: Literature survey

Sr no	Title	Inference
1	Intelligent chatting service using AIML[5]	AIML templates based services available for different chatbots development, which may reassemble a personality & act as consumer care website also provide API's for Weather
2	Eliza—a computer program for the study of natural language communication between man and machine [1]	the first chatbot, is created by Joseph Weizenbaum. It simulates a psychotherapist and uses pattern matching and template-based responses.
3	NEU-chatbot: Chatbot for admission of National Economics University [3]	Developed AI-based chatbot for IELTS help , this chatbot developed using the Rasa framework
3	Smarter Child [14]	a Chatbot capable of retrieving information from databases about various topics, is created. It is available on messaging platforms like AOL and MSN.
4	AI Based Chat-Bot Using Azure Cognitive Services[12]	A Chatbot is created using framework available for implementing Chatbot system

The motivation for this work is to model a chatbot which can be used for various admission process to answer the queries related to admission in legitimate and effective manner. Focus of this experiment to check the correctness of information delivery to the students seeking admission.

## 2. Architecture

Figures 1 illustrate the components involved in the process of training stage, which gives general overview of the auto-reply bot. The testing of this trained model explained in Figure 2.

The initial step involves pre-processing the dataset for the training stage. The detailed general view of the process explained below. The pre-processing steps involved techniques identified in NLP listed such as Tokenization of input, later on Stemming/Lemmatization of the data is done, and then Regex Extraction is done on same data. The resulting processed data is then used to train the Feedforward neural network.

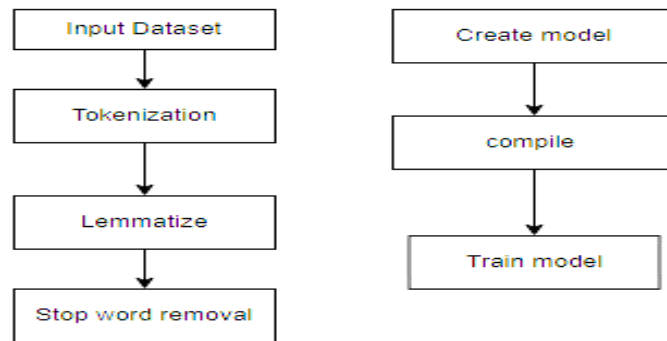


Figure 1: NLP task on input dataset and Model Training

Later in training stage, dense type of neural network model used to create input layer accordingly with the appropriate count of hidden layers are calculated and lastly output layers of neural networks neuron size is determined.

The model is then compiled using the method available and fit method is used for relevant parameters such as Activation Function for accuracy, Loss Function to calculate loss, Optimizer for optimizing the model, Epochs decides the learning size & Batch size for learning process improvement, and also Early stopping criteria used for reducing loss. In testing or prediction stage (Figure 2), user submits a query using Interface provided to the user. The query is pre-processing using techniques used to the train the model, then passed to the predicting/testing stage. The model responds to the queries with the probabilities associated for the tags which are crafted into the dataset, and the tag which is having maximum probability value is considered as the correct output by the model and send as the answer to the query raised by the user. In training and prediction stages techniques used are explained in later sections.

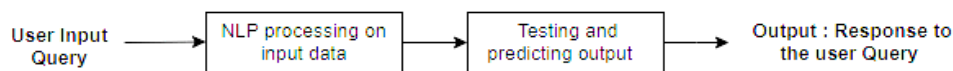


Figure 2: Response to the user Query

### 3. Input Dataset

The input dataset is created using Information brochure [4] available in the public domain which is later on compiled into the collection of Questions-Answer patterns and are stored in a JSON file. As illustrated in the figure 3. Each Questions are divided as per the intention of the users and intent was assigned a tag. The “patterns” contained different questions that users with a specific intention might ask to the chatbot, while the responses contains the appropriate answers to the questions. When a user query matches a questions, the chatbot sends a response in the form of answers replies. However, if a matching question and answer pattern is not in the provided list, the bot provides a default response to the query, indicating that, chatbots model not able to understand the query and reply back to the user to give more information or ask question in different way to get the more specific information which can relate to the user’s intent. The dataset includes a total 62 intent tags and 1059 questions that might be asked.

Below Is the figure illustrating how ‘dates ’related information is arranged in JSON file to create the question-answer scenario.

```

{
  "tag": "dates",
  "patterns": [
    "important dates",
    "tell me the dates",
    "what is the Schedule ",
    "Schedule for admission",
    "Please tell Schedule" ],
  "responses": [
    "For important dates please visit <a target=\"_blank\" href= \"
    https://poly22.dte.maharashtra.gov.in/diploma22/index.php/hp_controller/impDates\">
    here</a>"
  ],
},

```

Figure 3: description for tag ‘dates’ having intention of finding important dates related to admission process

#### 4. Pre-processing of input data

Natural language processing is an area of artificial intelligence that involves a machine's ability to analyse and comprehend human language input, process it using Natural Language Processing techniques [5], and creates a human-like response.

The main steps in processing the input text includes:

1. **Tokenization:**

Tokenization is the process of dividing a text or document into individual units, called tokens, which can be words, phrases, or even characters.

Example: The sentence "I love dogs" is tokenized into ["I", "love", "dogs"].

2. **Remove Frequent Words:** Most words in our text will only appear one or two times. It’s a good idea to remove these infrequent words as having a huge vocabulary will make our model slow to train.

3. **Lemmatization:** Lemmatization is the process of reducing words to their base form or lemma, considering their part of speech.

Example: The word "running" is lemmatized to "run" in the base form.

4. **Stemming:** Stemming is the process of reducing words to their root or stem form, often by removing suffixes and prefixes.

Example: The word "running" is stemmed to "run" by removing the "-ning" suffix.

5. **Stop words Removal:** this is the process of eliminating commonly used words that do not carry significant meaning in a text or document.

Example: In the sentence "The cat is on the mat," stop word "the" is removed, resulting in "cat is on mat."

Figure 4: Pre-processed input data

#### 5. Model Implementation

To achieve correct pre-processing of the questions input of the stakeholders of the admission process and get the most fitting response, we need to employ the Feedforward Model, which is implemented through a feedforward neural network. Prior to training the model, stored in a structure, so that it could be provided to the training model.

Sr No	Items	Description
1	Tags [13]	'Candidature Type', 'Contact', 'Fees', 'Notification', 'Scrutiny', 'Seat Matrix', 'allotment status', 'colleges', 'creator', 'cutoff', 'dates', 'greeting', 'information Broucher'
2	unique words (patterns) [67]	['admission', 'allotment', 'anyone', 'application', 'are', 'available', 'broucher', 'brouchers', 'by', 'candidature', 'check', 'college', 'contact', 'create', 'created', 'creator', 'cutoff', 'date', 'day', 'designed', 'developer', 'e-scrutiny', 'fee', 'for', 'good', 'hello', 'hey', 'hi', 'how', 'important', 'in', 'info', 'information', 'is', 'last', 'list', 'made', 'matrix', 'me', 'mode', 'much', 'my', 'name', 'notification', 'of', 'physical', 'schedule', 'scrutiny', 'seat', 'status', 'tell', 'the', 'there', 'to', 'type', 'up', 'waht', 'what', 'whats', 'whatsup', 'where', 'who', 'whom', 'ya', 'year', 'you', 'your']

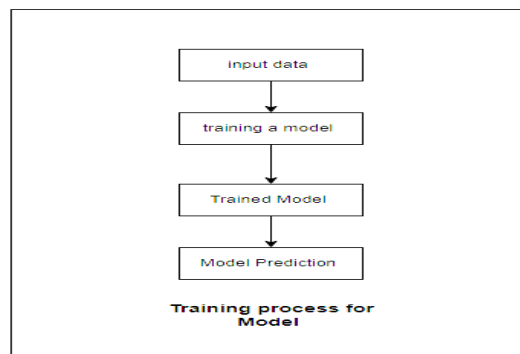
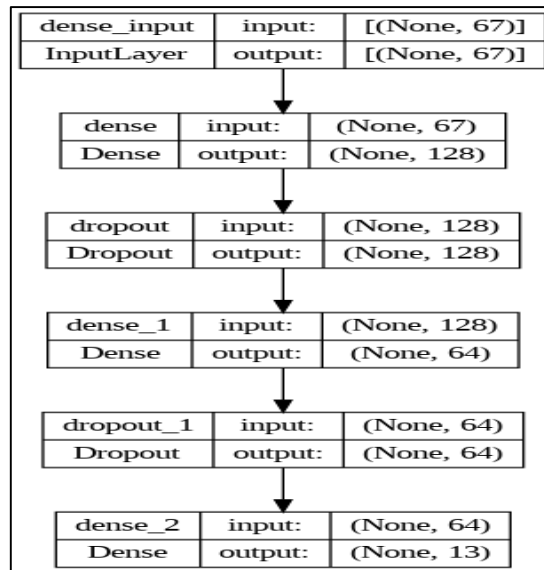


Figure 5: Steps of the training process for Model

Training data, which is to be fed into the Artificial Neural Network model, was constructed by combining the unique words and tags as shown in equation below:

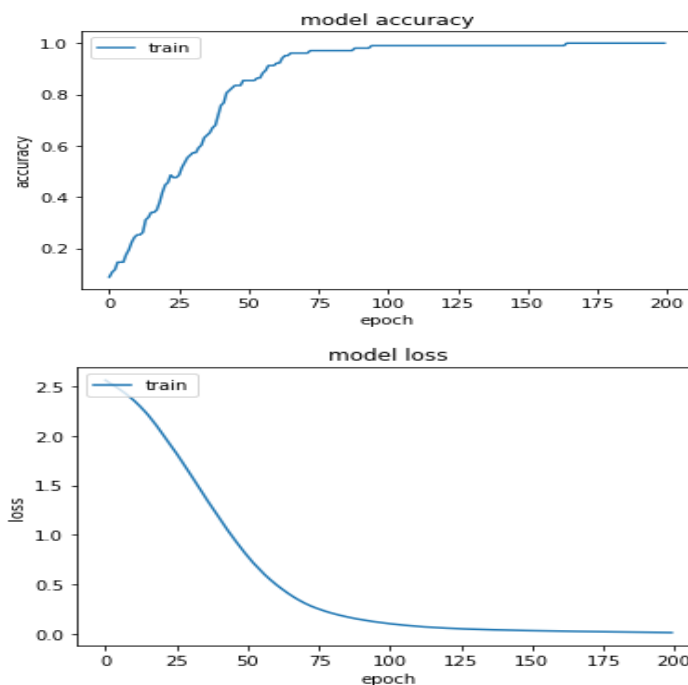
$$\text{Training data} = (\text{unique words}) + (\text{tags})$$

The Artificial neural network architecture for model is illustrated in Figure 7. The first layer of the network is called the input layer to the neural network, which is made up of 67 neurons as per the unique words pre-processed in above steps. For calculating the output size of the created input layer is calculated as 2/3 of the total number of output neurons which is around 13 tags, which can be calculated in a middle hidden layer with 8 neurons. Softmax activation function used for output layer use and has 13 neurons that correspond to the 13 intents. The Softmax function, used in output layer as it maps the non-normalized output to a probability distribution over the predicted output classes, and the probabilities range between 0 to 1, giving the sum of all probabilities equal to one [10].



**Figure 6 Neural Network used in Project**

After the total number of layers is specified for the model, we have to configure the learning process and is configured using method available as the compile method. In this step, an optimizer for getting accuracy and a loss function to reduce the loss are specified. This model is using the ADAM optimizer, which is used as first-order gradient-based optimization for the stochastic objective functions. ADAM is uses an adaptive learning rate process for computing the individual learning rates for training the model. It uses different parameters for this processing.

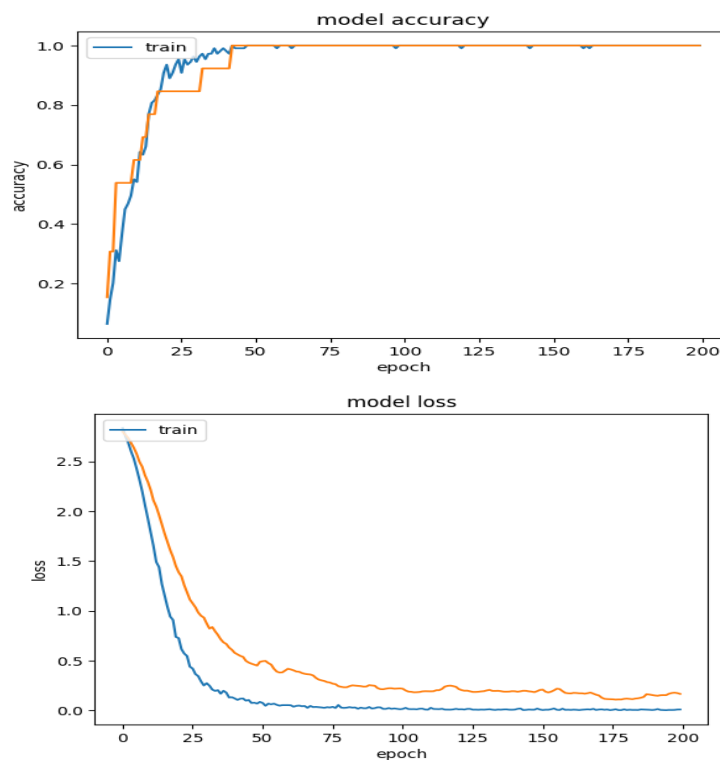


**Figure 7: Model performance before corrective actions**

For training this model we have configured methods for Epochs equals to 200, set batch-size to 20, as the process started for learning, model has taken a correct distribution of the data, and it is observed that after some time, model results show getting overfitting. Generally, this overfitting of data is occurring when a

model starts to learn and incurred noise in training data to the such amount that it starts to impact overall performance after using the unseen data or new data. There are a couple of methods to tackle the problem of overfitting, which refers to a situation when used machine learning model starts to performs well on the training data and starts to perform poorly on the test data. One approach is early stopping, which involves terminating the training process when model starts to perform worst on a validation data set. This is a widely adopted and straightforward method.

Another technique is called dropout, which is a regularization method that randomly omits certain neurons during training, temporarily removing their assistance to the activation of related neurons. The figures 11 and 12 illustrate the outcomes of applying these methods to resolve overfitting to increase the models accuracy and reduce the models loss.



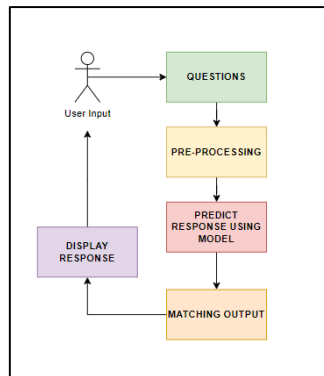
```

✓ [22] loss,accuracy = model.evaluate(train_x, train_y)
0s
4/4 [=====] - 0s 3ms/step - loss: 0.0178 - accuracy: 0.9918
    
```

**Figure 8: Model accuracy after adding dropout using ADAM optimizer**

In order to evaluate the effectiveness of early stopping and dropout in addressing overfitting, total 13 test cases were applied on to the model, and the output tags generated are compared against the expected output tags. Prior to implementing these techniques, the model was able to provide 10 correct responses. However, after incorporating early stopping and dropout, the model demonstrated improved generalization, with 12 correct responses being generated.

Following the creation of the model, the system executes user query pre-processing and output prediction process tasks, as depicted in Figure 14. The input to the model consists of stemmed and tokenized questions pre-processed dataset, while the output is represented by their corresponding generated tags. This models neurons weights are fixed to enable the prediction of the most appropriate intent tag for a given user query, which uses the probability generated by applying the Softmax activation function [10].



**Figure 9. Query input and response generation model**

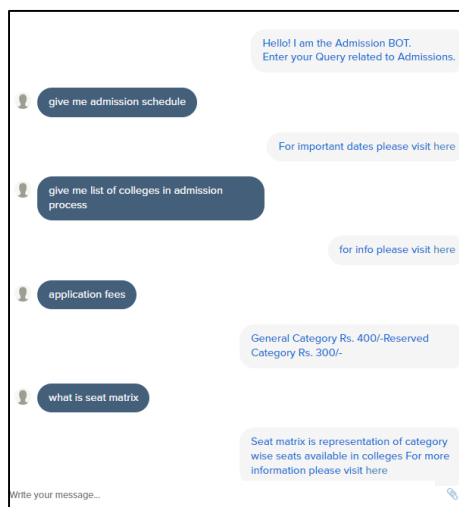
As an illustration, Figure 15 depicts the intent and corresponding probability generated by the model for the user query "Which colleges are available for admission?". The input to the model is the pre-processed query, which has undergone NLP processing. The feed-forward model processes the input and creates an output in the form of the highest probability indicating the matching intents tag, from a pool of 62 intents and tag classes.

```

You :college list
1/1 [=====] - 0s 21ms/step
[{'intent': 'colleges', 'probability': '0.9999807'}]
Bot : Here you can find the list of colleges participating in admission process , For more information please visit
  
```

**Fig 10: Output for the Finding colleges in admission query**

The figure presented demonstrates that our model has achieved an 99% accuracy match with the "colleges" intent tag, which is responsible for handling queries related to admissions. Consequently, the model has successfully identified the perfectly matched intent from the available intent asked in the user question, by selecting the intent class with the highest probability. A response is then randomly chosen from the intent picked up and given to display on the interface for the user. Below is the output received for some of the queries.

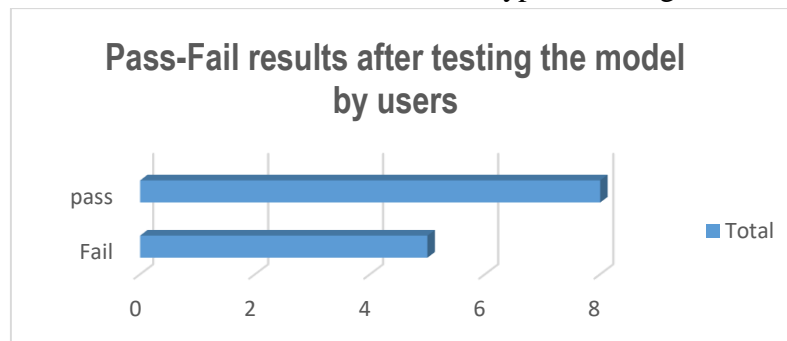


**Figure 10: GUI for Question and Answer**



## 6. TESTING

To understand the accuracy and effectiveness of chatbot, testing phase plays an important role. We focused on whether our chatbot performs correctly with all the information in the form of intents classes or not. Whether responses generated are correctly delivered by the chatbot is checked in this step. For this we have tested chatbots responses with 13 questions and given to the expert people in the field of admission process. After checking we found that out of 13 questions chatbot is able to answer the queries of 8 questions and 5 questions were answered wrong. In Fig 16 graph of the pass-fail results of live testing done shows the performance of chatbot after questions were asked by the group of 5 people depending upon the 13 known intents. Which are asked with different types wordings.



**Fig 11: Results of chatbot queries raised for known question related queries**

## 7. Conclusion

the chatbot developed in this study proved effective in addressing Admission -related user queries, with hyper parameter adjustments made to optimize the model's accuracy. End-user testing demonstrated an improved number of correct responses, with a generated probability score 0.99 achieved after the corrective actions of change in training phrases and accommodating all the parameters correctly which leads the user to correct information. The use of AI-powered chatbots in Admission systems expected to provide more personalized and efficient problems resolutions with respect to students and colleges in providing more legitimate and correct information delivery to the stakeholders, which may lead to increase in involvement of stakeholders and also can help to reduce the response time.

## 8. Future work

The chatbot prototype developed in this study was designed to guide in the student's admission process which are complicated to understand. However, the application has the potential to be deployed by other government institutions which serves for the citizens and can expand at a wider scale. The model's performance will be improved by training model with domain experts and understanding the focused queries raised by the stakeholders during the process. It will help to increase accuracy, while there is need to address potential risks and vulnerabilities associated with the deployment of chatbot with government websites.

## 10. References

1. Joseph Veizenbaum, "ELIZA: A Computer Program For the Study of Natural Language Communication Between Man And Machine.", Communications of the ACM, Volume 9, January 1966.
2. Bayan AbuShawar, Eric Atwell, "ALICE Chatbot: Trials and Outputs", Computación y Sistemas, Vol. 19, No. 4, 2015, pp. 625–632
3. Trung Thanh Nguyen, Anh Duc Le, Ha Thanh Hoang, Tuan Nguyen, NEU-chatbot: Chatbot for admission of National Economics University, Computers and Education: Artificial Intelligence, Volume 2, 2021, 100036, ISSN 2666-920X, <https://doi.org/10.1016/j.caeai.2021.100036>
4. [https://poly22.dte.maharashtra.gov.in/diploma22/admin/uploads/Diploma\\_Information\\_Brochure\\_English\\_2022-23\\_2.pdf](https://poly22.dte.maharashtra.gov.in/diploma22/admin/uploads/Diploma_Information_Brochure_English_2022-23_2.pdf)
5. Saqib, G., K. Faizan, and N. Ghatte. "Intelligent chatting service using AIML." 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT). IEEE, 2018.
6. Vamsi, G. Krishna, Akhtar Rasool, and Gaurav Hajela. "Chatbot: A deep neural network based human to machine conversation model." 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2020.
7. Anki, Prasnurzaki, et al. "High Accuracy Conversational AI Chatbot Using Deep Recurrent Neural Networks Based on BiLSTM Model." 2020 3rd International Conference on Information and Communications Technology (ICOIACT). IEEE, 2020.
8. Lee, Naeun & Kim, Kirak & Yoon, Taeseon, "Implementation of robot journalism by programming custombot using tokenization and custom tagging", (ICACT.2017.), pp.566-570, 2017.
9. Pérez, J. Q., Daradoumis, T., & Puig, J. M. M. (2020). Rediscovering the use of chatbots in education: A systematic literature review. Computer Applications in Engineering Education, 28(6), 1549-1565.
10. G. Cybenko, "Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems", 2(4):pp.303–314, 1989
11. Michael Nielsen, "Neural networks and Deep learning: Introduction to core principles", 2(3):pp.39–53, 2019.
12. K. Tajane, S. Dave, P. Jahagirdar, A. Ghadge and A. Musale, "AI Based Chat-Bot Using Azure Cognitive Services," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-4, doi: 10.1109/ICCUBEA.2018.8697737.
13. Bhartiya, Namrata, et al. "Artificial neural network based university chatbot system." 2019 IEEE Bombay Section Signature Conference (IBSSC). IEEE, 2019.