2019 IEEE Bombay Section Signature Conference (IBSSC)

ARTIFICIAL NEURAL NETWORK BASED UNIVERSITY CHATBOT SYSTEM

Namrata Bhartiya
bhartiya_namrata@yahoo.com

Namrata Jangid
namratanj97@gmail.com

Sheetal Jannu
sheetal.jannu123@gmail.com

Purvika Shukla
pshukla1997@gmail.com

Radhika Chapaneri
radhika.chapaneri@nmims.edu

Mukesh Patel School of Technology Management and Engineering,
NMIMS University

Abstract- **Chatbots have effectively reduced human efforts by providing automated human-like solutions for various business and societal problems. This paper is an elaborate description of the design and implementation of a University Counselling Auto-Reply Bot, that is capable of providing answers to queries related to the field of Engineering at our University level. The appropriate NLP techniques are applied to our University Data, developed in the JSON format and the Feedforward neural model is used for training this dataset, the issue of overfitting was handled. The Chat Application is then deployed on Facebook Messenger, the response is visible to the User on the Facebook Messenger interface, to provide them with an effective interaction platform. The end-user testing was conducted in two phases; the probability scores of the correct responses were improved to 0.72 in the second phase from 0.46 in the first phase after devising additional training phrases and keywords to the dataset.**

Keywords- Artificial Intelligence; Artificial Neural Network; Chatbot; Natural Language Processing.

I. INTRODUCTION

Chatbots are applications that are used in dialog systems for several practical purposes such as customer service and information acquisition. The "University Counselling Chatbot System" plays the role of a Counsellor and provides the most appropriate guidance to students who aspire to join the particular university. The implementation of our Chabot was aimed at solving several existing problems in the field of education, such as a large number of users being unable to access professional career guidance, since it is expensive; users spending hours browsing websites looking for relevant required information; and third-party website information not being genuine and reliable or regularly updated. This bot intends to become a one-stop solution for answering queries related to the admission process, course & streams available, student intake, the importance of a particular stream or course and job placement related queries. Our aim is solving the dilemma of students wanting to join a university by providing them with consolidated, authentic and precise information through a simple chat window. In this paper we discuss the design and implementation of a University Counselling Auto-Reply Bot, that is capable of providing answers to queries related to the field of Engineering at our University level. Our bot is based on the Feed-Forward Artificial Neural Network model, a framework which is used for processing complex data inputs and has a unidirectional flow of information, with no feedback loops. Section II of this paper describes the basic architecture of the workings of the model for implementing the chatbot application. Section III discusses the creation of the University dataset and importing the major libraries used throughout the implementation. Section IV explains the preprocessing steps involved in the dataset & the user query. Section V of the paper is a brief summary of the stages involved in the actual model implementation. Section VI explains the deployment and integration of the implemented model with the User Interface. Section VII is an overview of the results obtained in the Analysis and Testing phases of the chatbot application. Table 1 depicts the Literature survey related to Chatbots existing in the field of Education.

Table 1: Literature Survey

Research Papers	Inferences
ELIZA: A Computer Program For the Study of Natural Language Communication Between Man And Machine.	1. Identifies keywords, discovers minimal context and chooses appropriate transformations. 2. Generates responses in the absence of keywords, and provides editing capability for ELIZA "scripts".[1]
ALICE Chatbot: Trials and Outputs	1. ALICE stores knowledge about English conversation patterns in Artificial Intelligence Markup Language(AIML) files. 2. The AIML interpreter tries to match word by word to obtain the longest pattern match.[2]
AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine	1. AliMe Chat uses an attentive Seq2Seq based rerank model to optimize the joint results 2. The retrieval model employs search technique to find the most similar question for input and

978-1-5386-7401-7/19/$31.00 ©2019 IEEE

	then obtain the paired answer.[3]
Automated Web-Bot Implementation using Machine Learning Techniques in e-Learning Paradigm	1. Extensive approach that concentrates on supervised and unsupervised machine learning strategic learning methods. 2. The elements used are NLP techniques, Data Mining techniques, Machine learning approaches.[4]

II. ARCHITECTURE

The architecture of the training and testing modules of the auto-reply bot, as shown in Figures 1 and 2, represent the various functional and logical components of the training and testing procedures which forms the blueprint of the chatbot implementation process.

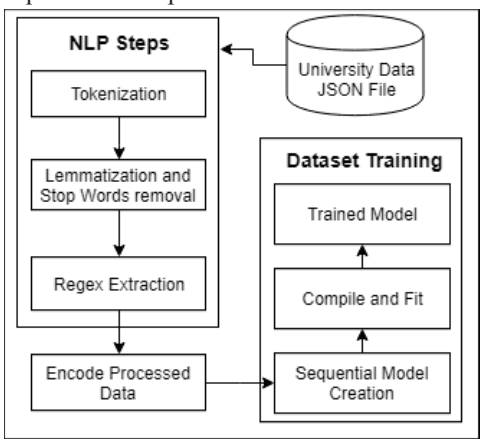

Figure 1: Architecture for Training Module

In the initial part of the training phase, as shown in Figure 1, the constructed dataset is preprocessed. The preprocessing stage includes Natural Language Processing(NLP) steps: Tokenization, Stemming/Lemmatization, and Regex Extraction. The resulting processed data is encoded using the One-Hot Encoding technique. The encoded data is then utilized in the training stage of the Feedforward Model. In the later part of the training phase, a dense neural network model is created with a suitable number of hidden layers and appropriate output neuron shape. The model is compiled and fit employing the pertinent Activation Function, Loss Function for classification, Optimizer, Number of Epochs, Batch size and Early stopping criteria for loss.

The second phase is the testing or the prediction phase, as shown in Figure 2, where the end user is involved. The user query is recorded using a suitable User Interface and transferred to the query preprocessing stage. In the preprocessing stage, techniques similar to that of the training phase are employed. The preprocessed query is encoded and passed on to the predicting or the testing stage. Thereafter, the model responds with a list of probabilities for all the tags present in the dataset.

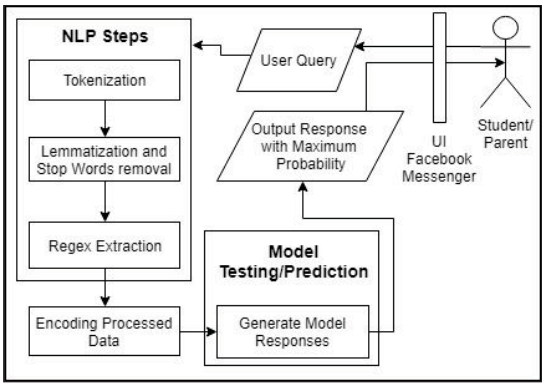

Figure 2: Architecture for Testing or Predicting using the model.

The tag with the maximum probability is provided as the response to the end user. The techniques mentioned in the training and the prediction phase are explained further.

III. DATASET CREATION AND TOOLS

A. Dataset Creation: Our dataset comprises of various query patterns and their corresponding responses stored in a JSON file. An intent determines the intention of the user interacting with the system. While storing an intent, a tag field defines the intention of that particular intent. The patterns field comprises of various types of questions a user having a particular intention may ask the chatbot, and the responses field comprises of corresponding relevant answers to those questions as shown in figure 3. If the bot can find a matching pattern for a given user query, it will send the response back to the user. However, if a matching pattern is not found for a given user query, the bot will provide a default fallback response, specifying that it cannot interpret the query and will ask the user to be more specific. The dataset consists of 62 intents tags and 1059 query patterns in all.

```
{"tag": "whyMechatronics",
 "patterns": ["prerequisites for mechatronics engineering",
              "skills required for mechatronics engineering",
              "i am confused if i should take mechatronics",
              "benefits of taking mechatronics"],
"responses": ["Let me help you! If you hold interest in
               Electronic Concepts, Programming, Mathematics
               and Physics, then Mechatronics is for you!"]
},
```

Figure 3: Intent for 'why Mechatronics' tag

IV. DATASET AND USER QUERY PREPROCESSING

Natural language processing is a branch of AI and it is the capability of a machine to perform analysis by understanding the natural language input of the user, to process that input with appropriate NLP techniques [5], and to generate natural language human-like response.

The major steps in Natural Language Processing, as shown in Figure 4 are listed below:

1. *Tokenization*: Tokenization involves converting the character stream into a set of individual tokens. Tokens can be words, numbers, identifiers, special characters or punctuation.

978-1-5386-7401-7/19/$31.00 ©2019 IEEE

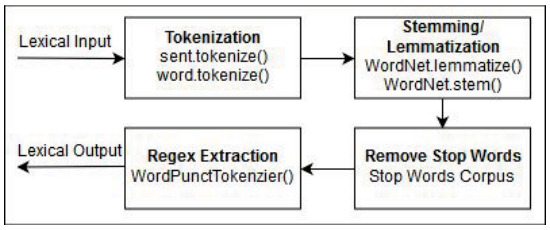

Figure 4: Flow Diagram for Processing Input in NLP

2. *Lemmatization*: Lemmatization involves reducing word variations into simpler forms. Lemmatization uses a language dictionary to perform an accurate reduction to root words Lemmatization was found to be better than stemming which uses simple pattern matching to remove suffixes of tokens.
3. *Removal of stop words:* Stop words like 'is', 'are', 'the', etc. do not add value to the meaning of the sentence. They are not as important as the keywords, hence they can be removed from the text for better processing.
4. *Regex Extraction:* Words such as 'can't', 'don't', 'b.tech', 'm.tech' can be specified in different ways; therefore, they should be normalized and tagged. Therefore, using a regular expression such words are converted to one standard form [6].

```
#Sample Queries
query1 = "I want to know the fee structure for MBA. Tech"
query2 = "What are the available undergraduate programs?"
query3 = "Why should I opt for Computer Science?"
query4 = "Please provide a list of international companies"
```

```
Tokenized Sentences:
['i', 'want', 'to', 'know', 'the', 'fee', 'structure', 'for', 'mbatech']
['what', 'are', 'the', 'available', 'undergraduate', 'programs', '?']
['why', 'should', 'i', 'opt', 'for', 'computer', 'science', '?']
['please', 'provide', 'a', 'list', 'of', 'international', 'companies']
```

```
Stemmed/Lemmatized Sentences:
['i', 'want', 'to', 'know', 'the', 'fee', 'structure', 'for', 'mbatech']
['what', 'are', 'the', 'available', 'undergraduate', 'program', '?']
['why', 'should', 'i', 'opt', 'for', 'computer', 'science', '?']
['please', 'provide', 'a', 'list', 'of', 'international', 'company']
```

```
Stop words free Sentences:
['want', 'know', 'fee', 'structure', 'mbatech']
['available', 'undergraduate', 'program', '?']
['opt', 'computer', 'science', '?']
['please', 'provide', 'list', 'international', 'company']
```

Figure 5: Implementation of NLP steps on the University data

The above NLP steps are performed on each pattern of every intent present in the dataset. Similar techniques are applied to each query submitted by the user to the chatbot as well.

V. MODEL IMPLEMENTATION

We are using the Feedforward Model for carrying out the accurate processing of the input user query to generate the most appropriate response. We have implemented a simple feedforward neural network.

Before implementing the training model, the dataset was encoded using one hot encoding and stored in a structure in order to be supplied to the training model. One hot encoding transforms a single variable with n observations and d distinct values, to d binary variables with n observations each [13].

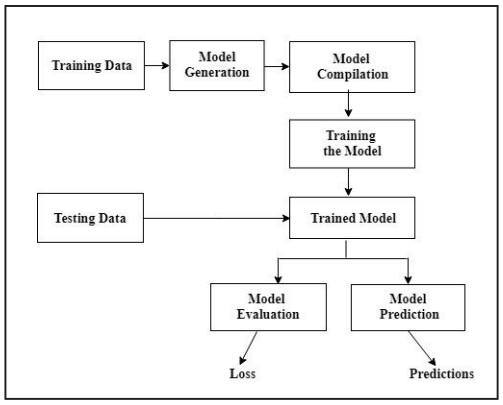

Figure 6: A flowchart depicting the steps of the training process for Sequential Model [9].

The Different Lemmatized Words List and Intent Classes List were encoded using one hot encoding method. The training which is to be passed to the ANN model is designed such that it is a combination of the encoded distinct lemmatized words and tags as shown in (1).

Training Set = {encoded(different lemmatized list) +encoded(tag list)} (1)

$$\text{Train\_set} = \begin{matrix} 0 \\ \downarrow \\ 1059 \end{matrix} \left\{ \begin{matrix} \{bag_0, bag_1 \ldots bag_{339}\}\{tag_0, tag_1 \ldots tag_{62}\} \\ \{bag_0, bag_1 \ldots bag_{339}\}\{tag_0, tag_1 \ldots tag_{62}\} \\ \{bag_0, bag_1 \ldots bag_{339}\}\{tag_0, tag_1 \ldots tag_{62}\} \\ \vdots \qquad\qquad\qquad\qquad \vdots \end{matrix} \right\}$$

Figure 7 depicts an intended Neural Network specific for our model. The first layer in our Neural net is the input layer, which consists of 339 neurons, which are the unique lemmatized words. The output shape of the input layer is approximately 2/3rd of the number of output possibilities ie. 62 Tags (⅔ of 62 = 41). Thus the middle hidden layer consists of 41 neurons. The Softmax activation function is used for the output layer, which comprises 62 neurons, that correspond to the 62 intent classes. The main advantage of using Softmax is the output probabilities range, as it helps to map the non-normalized output to a probability distribution overpredicted output classes as seen from the mathematical equation (2). The range is between 0 to 1, and the sum of all the probabilities will be equal to one [10].

$$softmax(z_i) = \frac{exp(z_i)}{\sum_j exp(z_j)} \quad (2)$$

After specifying the number of layers for the model, we configure the learning process using the compile method wherein we specify an optimizer, a loss function, and a list of metrics. Our system uses ADAM optimizer. Adam is an algorithm for first-order gradient-based optimization of stochastic objective functions. It is an adaptive learning rate method, which means, it computes individual learning rates

978-1-5386-7401-7/19/$31.00 ©2019 IEEE

for different parameters[12].

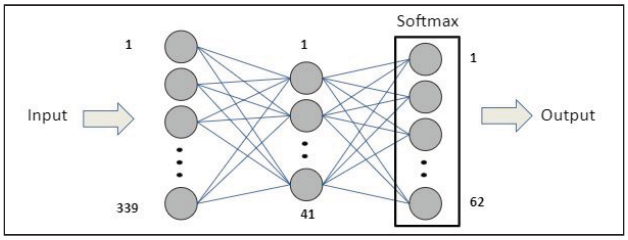

Figure 7: Neural Network created for the project

We use categorical Cross-entropy loss or log loss. It measures the performance of a classification model whose output is a probability value between 0 and 1. This loss increases as the predicted probability diverge from the actual label.

Once the learning process is configured, we train the model by specifying the input, output and setting the hyperparameters that are the number of epochs and the batch size used for the training process.

In order to assess the performance of our model, we considered two metrics: loss and accuracy. From Figure 8 it can be observed that the model accuracy goes on increasing till it becomes almost constant after a certain number of epochs. Conversely, from Figure 9 it can be observed that the model loss gradually decreases until it becomes almost constant after a certain number of epochs.

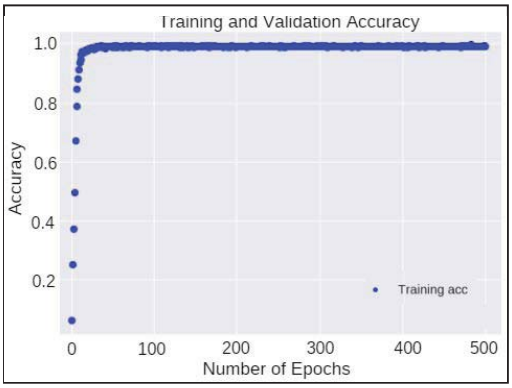

Figure 8: Result of Model Accuracy.

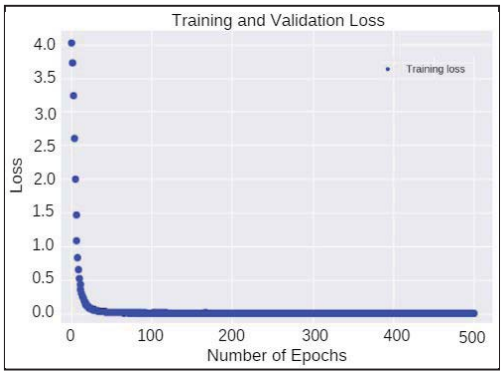

Figure 9: Result of Model Loss.

With the number of Epochs set to 500, with a batch size of 16, the model started by learning a correct distribution of the data, but at some point, starts overfitting the data. Overfitting occurs when a model learns the detail and noise in the training data to an extent that it negatively impacts the performance of the model on new data.

```
Epoch 490/500
1016/1016 [==============================] - 0s 93us/step - loss: 0.0127 - acc: 0.9892
Epoch 491/500
1016/1016 [==============================] - 0s 95us/step - loss: 0.0128 - acc: 0.9882
Epoch 492/500
1016/1016 [==============================] - 0s 92us/step - loss: 0.0130 - acc: 0.9902
Epoch 493/500
1016/1016 [==============================] - 0s 94us/step - loss: 0.0128 - acc: 0.9872
```

Figure 10: Accuracy and Loss result of Model.

Solutions for addressing the issue of Overfitting:

A. *Early Stopping:* To stop the training at the point when performance on a dataset starts to degrade. This is a simple, effective, and widely used approach.

B. *Dropout:* We used this simple and powerful regularization technique. It is a technique where randomly selected neurons are ignored during training. They are "dropped-out" randomly. This means that their contribution to the activation of downstream neurons is temporarily removed [14].

The model Accuracy and Loss results after resolving to overfit are shown in Figures 11 and 12.

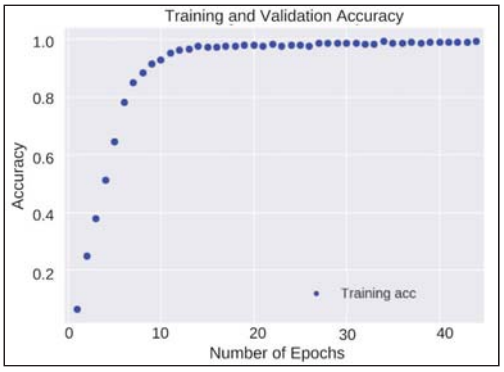

Figure 11: Result of Model Accuracy after solving overfitting

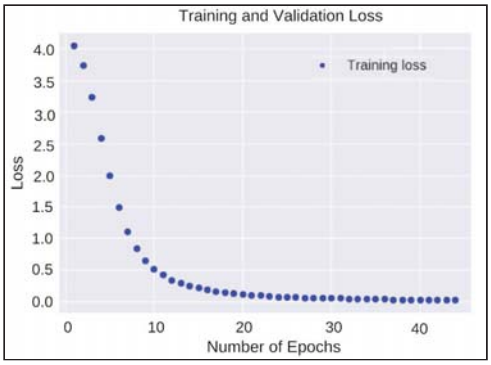

Figure 12: Result of Model loss after solving overfitting.

```
Epoch 42/500
1016/1016 [==============================] - 0s 97us/step - loss: 0.0397 - acc: 0.9902
Epoch 43/500
1016/1016 [==============================] - 0s 100us/step - loss: 0.0364 - acc: 0.9911
Epoch 44/500
1016/1016 [==============================] - 0s 97us/step - loss: 0.0329 - acc: 0.9931
Epoch 00044: early stopping
```

Figure 13: Loss and Accuracy result after solving overfitting.

978-1-5386-7401-7/19/$31.00 ©2019 IEEE

A user query list of 100 test cases was passed to the model, the response tags were compared to its expected response tag list before and after implementing early stopping and dropout. The result showed 64 correct responses provided before the issue of overfitting was handled and 78 correct responses after addressing the overfitting issue as the model were able to generalize better.

After the model generation, user query preprocessing and response prediction tasks are performed shown in fig. 14. One hot encoded stemmed and tokenized query patterns act as the input and their corresponding one hot encoded intent tags act as the output of the model. The weights of the model are fixed such that for a given user query, the model predicts the best-matching intent tag based on probabilities provided by the softmax activation function [10].

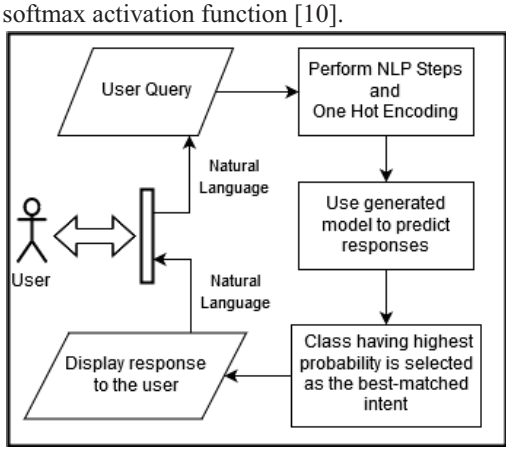

Figure 14: User Query processing and Model implementation to predict class with the highest probability as the most suitable response.

For example, Figure 15 shows the resulting intent and probability for the user query "Which is better CS or IT?". This user query is preprocessed by tokenization, lemmatization, and one-hot encoded; which is the input to the feed-forward model. The output is the highest probability of the input query pattern belonging to the best-matched intent tag, out of the existing 62 intent classes.

```
Resulting Intent: itVsCS
Probability: 0.8515985
Intent Tag: itVsCS
```

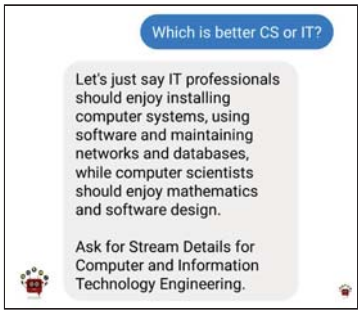

Fig.15: Result Tag Probability and Response.

Our trained model provides 85.15% match to "ITvsCS" intent tag, as shown in the figure, of the above user query belonging to the intent handling all queries related to admissions. Therefore, it successfully finds the best-matched intent tag for the query by selecting the class having the highest probability according to the model. A response is picked out at random from the selected intent and displayed to the user on the interface.

VI. DEPLOYMENT

We are hosting our application on Heroku, a cloud platform as a service(PaaS) which enables support by allowing developers to build, run and scale their applications. It deploys applications with Git, a version control system. As shown in Figure 16, we begin by installing Git and Heroku CLI. The next step is to clone the GitHub repository. We initialize the local git repository in our application's root directory and commit the changes in each phase locally. The function of Git remotes is storing different versions of our repository. We deploy our application by pushing its code to a special Heroku-hosted remote, associated with the application. Heroku only deploys code that is pushed to the master branch of the Heroku remote. For deploying the latest committed version of the application to Heroku, we use the *git push* command to push the code from the local repository master branch to the Heroku remote.

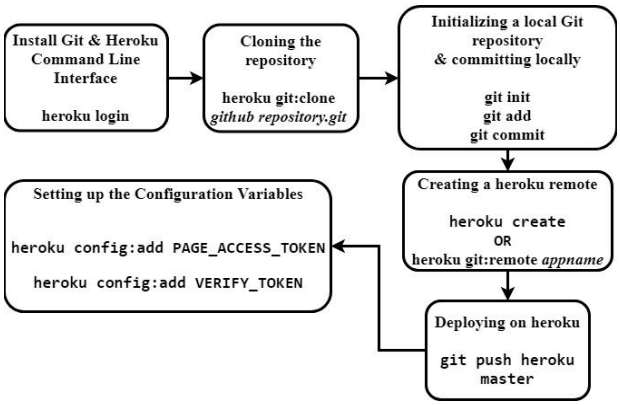

Figure 16: Deployment using Heroku.

Then we create the Flask app, which uses the send and receives API of Facebook for integration with Facebook Messenger. The next step is webhook registration on our Facebook Messenger application page. We set up webhooks by entering the link of our Heroku application under the callback URL section and specifying the verify_token as used in the code deployment process [12].

VII. TESTING AND ANALYSIS

Testing is a critical step to determine the success of the effective working of the chatbot. Our aim was to analyze if all the features of the bot were correctly incorporated and whether appropriate responses are being generated for each user query. The two types of testing performed are discussed below:

A. *Developer Testing / System Testing*: This testing involved verification and validation of a set of user queries covering the scope of the chatbot. A user query list of 100 test cases was tested by comparing its actual response tag list to the expected response tag list.

978-1-5386-7401-7/19/$31.00 ©2019 IEEE

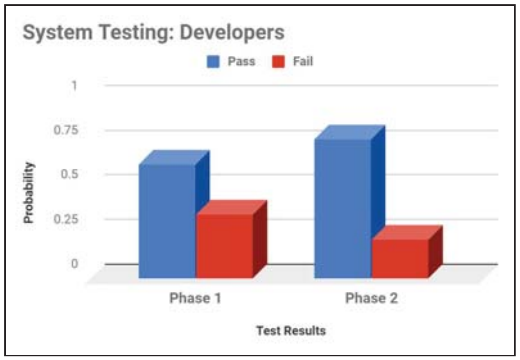

Figure 17: System Testing Result.

The results were noted in two phases, phase 1 is before correcting overfitting and phase 2 is after overfitting. According to the results generated, 64 test cases passed while 36 failed in phase 1 and 78 passed and 22 failed in phase 2 depicting better generalization. A graphical representation of the test results is shown in Figure 17.

B. *Alpha Testing:* Our approach involved getting the chatbot application tested by a few of the faculty members & students at our University. The test results (Pass or Fail for each user query) of each tester were observed and documented with their corresponding summary of the defect. The expected and the provided outputs for every test case were analyzed. The feedbacks of the testers were also recorded, following which the required changes were incorporated to improve the chatbot. The alpha testing for the bot was performed in two phases. The failures discovered in Phase I was rectified before moving on to Phase II of the testing.

The results of both phases are depicted in Figure 18 in the form of probabilities of each defect summary. From the results, it can be observed that the performance of the bot has significantly improved from Phase I to Phase II of testing with a probability score increase to 0.72 from 0.46 for correct matches. Figure 18 shows the type of defect and their occurrences as a result of alpha testing:

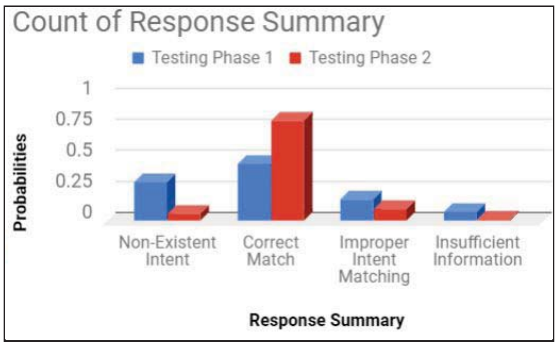

Figure 18: Summary of Defect Histogram.

VIII. CONCLUSION

Our chatbot was capable of solving user queries related to the university to a great extent as observed in the test results obtained. In order to optimize the model used in the implementation, several hyperparameter adjustments were made until the most effective values to minimize the cost function and to maximize the accuracy were found. The end-user testing was done in two phases, the number of correct responses was improved in the second phase exhibiting a probability score of 0.72 after devising additional training phrases and keywords.

The introduction of artificial intelligence in the field of education and counseling by means of application based chatbots as one discussed in our paper will give rise to more personalized, efficient and faster query solutions leading to increased user engagement and saving resources such as time and money in the near future.

IX. FUTURE WORK

The current scope of our Chatbot was limited to building a prototype for guiding those students who were aspiring to seek admission to the Engineering program of our University. However, the application developed by us has the potential to be used at a wider scale and the flexibility to be deployed by other institutions too. We intend to improve the robustness of our Chatbot Model by accommodating the best suitable parameters that help in reducing the loss incurred, and at the same time, increasing the accuracy of the predictions. We also intend to develop a Contingency Plan to manage the potential risks and vulnerabilities that our Bot could be exposed to.

X. 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]Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, Wei Chu, “AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine”, *55th AMACL*

[4]Iqbal Muhammad, Munwar & Farhan, Muhammad & Saleem, Dr. Yasir & Muhammad, Aslam, **“**Automated Web-Bot Implementation using Machine Learning Techniques in eLearning Paradigm**”**, *Journal of Applied Environmental and Biological Sciences*, 2014.

[5]Paul Nelson, “Natural Language Processing (NLP) Techniques for Extracting Information”, *searchtechnologies.com* [Online].

[6]Tharindu Weerasooriya, Nandula Perera, S.R. Liyanage, “A Method to Extract Essential Keywords from a Tweet using NLP Tools”, International Conference on Advances in ICT for Emerging Regions, 2016

[7]Bayu Setiaji, Ferry Wahyu Wibowo, ”Chatbot Using A Knowledge in Database: Human-to-Machine Conversation Modeling”, *7th International Conference on Intelligent Systems, Modelling, and Simulation,* 2016.

[8]Tarun Lalwani, Shashank Bharotia, Ashish Pal,Shreya Bisen, Vasundhara Rathod “Implementation of a Chatbot System using AI & NLP”, *International Journal of Innovative Research in Computer Science & Technology - IJIRCST,* Volume 6, Issue 3, May 2018.

[9] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. *ICLR.*

[10]Guruswami Hiremath, Aishwarya Hajare, Priyanka Bhosale, Rasika Nanaware, “Chatbot for education system”, *International Journal of Advance Research, Ideas and Innovations in Technology*

[11] F. Perez-Cruz, "Kullback-Leibler divergence estimation of continuous distributions," 2008 IEEE International Symposium on Information Theory, Toronto, ON, 2008, pp. 1666-1670. doi: 10.1109/ISIT.2008.4595271

[14]Diederik P. Kingma, Jimmy Lei Ba; *"Adam: A Method for Stochastic Optimization***”**, *ICLR 2015*

978-1-5386-7401-7/19/$31.00 ©2019 IEEE