iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** XI **Month of publication:** November 2021

DOI: https://doi.org/10.22214/ijraset.2021.38757

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XI Nov 2021- Available at www.ijraset.com

COVID-19 ChatBot

Satish Tirumalapudi[1], Venkata Vamsi Medidi[2]
[1, 2]*System Engineer, Micron Technologies, India, Hyderabad*

Abstract: ***Chat bots are software applications that help users to communicate with the machine and get the required result, this is where Natural Language Processing (NLP) comes into the picture. Natural language processing is based on deep learning that enables computers to acquire meaning from inputs given by the users. Natural language processing techniques can make possible the use of natural language to express ideas, thus drastically increasing accessibility. NLP engines rely on the elements of intent, utterance, entity, context, and session. Here in this project, we will be using Deep learning techniques which will be trained on the dataset which contains categories, patterns, and responses. Long Short-Term Memory (LSTM) is a Recurrent Neural Network that is capable of learning order dependence in sequence prediction problems. One of the most popular RNN approaches is LSTM to identify and control a dynamic system. We use an RNN to classify the category user's message belongs to and then will give a response from the list of responses.***

Keywords: NLP – Natural Language Processing, LSTM – Long Short Term Memory, RNN – Recurrent Neural Networks.

I. INTRODUCTION

Speech and textual information play a crucial role in communicating between humans. Chatbots have become popular due to the widespread use of messaging services and advancements of natural language. A Chatbot is a software application used to conduct an online chat conversation via text, instead of direct contact with a living human agent, these have become popular due to the widespread use of messaging services and the advancement of natural language understanding. Also known as conversational agents that mimic written human speech for purpose of human conversation or interaction with a real person. Now a days during the pandemic time it's became very difficult to know the availability of beds, medicines in different hospitals. Here Chatbot is used to give proper updated information to people instead of depending on other sources. Not only to know the information on availability of beds and medicines but also it's used to know any information on COVID-19 like vaccine availability and all other information. It can reduce the manual intervention.

II. PROPOSED SYSTEM

A. Architecture

The proposed one usually contains 5 major stages. The first stage is gathering the data related to availability of beds, medicines and other information from hospitals in json format and loading it by using json.load() method present in json package of python. In the second stage, pre-processing of the gathered data by using tokenization and lemmatization techniques. The third stage is creation of training and testing samples for the pre-processed data. The fourth step is to build a CNN modelized to predict the response for the given message to chatbot. The fifth step is to predict the response by using the model built in previous stage. The architecture for the proposed system is as shown below in Fig. 1

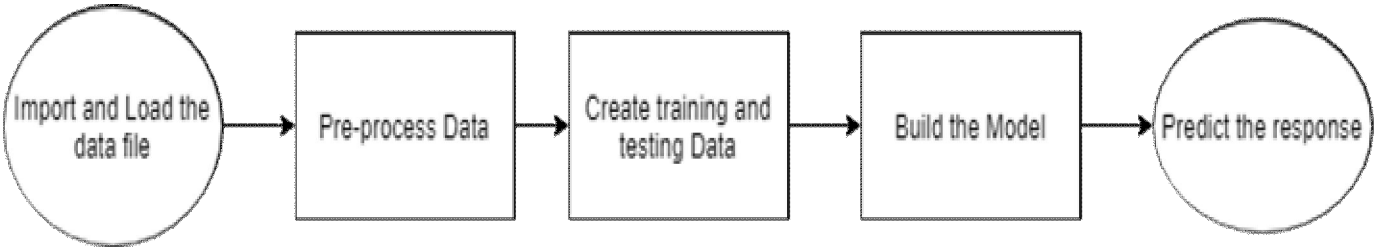


Fig. 1 Proposed architecture

III.METHODOLOGY

A. Import and Load the Data File

The proposed system will take dataset in the form of JSON. In this stage we need to gather dataset or create our own data set depends on the way want to use the chatbot. Here, we have taken a data set on hospitals from kaggle to build and use the model in chatbot.

©IJRASET: All Rights are Reserved

B. Pre-process Data

After gathering of dataset or creation of dataset next step is to preprocess the dataset. Here we have used Tokenization and Lemmatization preprocessing techniques to clean the dataset.

1) *Tokenization:* Tokenization is a fundamental task focused on text processing. Among other tasks, the segmentation process is used to identify information units, such as sentences and words. Tokenization is breaking the raw text into small chunks. Tokenization breaks the raw text into words, sentences called tokens. These tokens help in understanding the context or developing the model for the NLP. The tokenization also helps in interpreting the meaning of the text by analysing the sequence of the words. The most common way of forming tokens is based on space. There are different kinds of libraries available to perform tokenization. NLTK, Genism and Keras can be used to accomplish the task. Tokenization can be done to either separate words or sentences. If the text is split into words using some separation technique it is called word tokenization and same separation done for sentences is called sentence tokenization. The different tokenization techniques which can be available based on language and purpose of modelling. The different type of tokenization include white space tokenization, Dictionary based tokenization, Rule based, Regular expression, Penn TreeBank, Moses, Spacy and sub-word tokenization
2) *Lemmatization:* Lemmatization is one of the most common text pre-processing techniques used in Natural Language Processing (NLP) and machine learning in general. If you've already read my post about stemming of words in NLP, you'll already know that lemmatization is not that much different. Both in stemming and in lemmatization, we try to reduce a given word to its root word.

Fig. 2 Example of an Lemmatization

There are multiple techniques for lemmatization, one of them is string matching and dynamic programming called Levenshtein Distance. The second one is Trie based on popular data structure tree.

- *Levenshtein Distance:* The Levenshtein Distance is a procedure using dynamic programming to measure the smallest figure of a single character that is enunciated to edit or change between two words.

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) & if\ min(i,j) = 0 \\ min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} \end{cases}$$

- *Trie:* Trie or Prefix Tree is a tree-based data structure and exoteric for the mechanisms to store data. In our system, we narrated a Trie algorithm to implement in NLP approaches for Bangla word lemmatization. The mathematical expression of the formula is as follows:

$$Lemma(word) = min\ (distance\ (word_j, root\ words_i)$$
$$Where\ i = one\ to\ the\ number\ of\ the\ root\ word$$
$$j = one\ to\ the\ length\ of\ the\ highest\ prefix$$

C. Create Training and Testing Data

Now, we will create the training data in which we will provide the input and the output. Our input will be the pattern and output will be the class our input pattern belongs to. But the computer doesn't understand text so we will convert text into numbers by using np module in python.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XI Nov 2021- Available at www.ijraset.com

D. Build Model

We have our training data ready, now we will build a deep neural network that has 3 layers. We use the Keras sequential API for this. After training the model for 200 epochs, we achieved 100% accuracy on our model.

Create model - 3 layers. First layer 128 neurons, second layer 64 neurons and 3rd output layer contains number of neurons equal to number of intents to predict output intent with softmax. Compile model - Stochastic gradient descent with Nesterov accelerated gradient gives good results for this model.

1) *Convolutional Neural Network:* CNN is a Deep Learning which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and able to differentiate one from another. Convolutional layers consist of multiple features like detecting edges, corners, and multiple textures, making it a special tool for CNN to perform modelling. That layer slides across the image matrix and can detect its all features. This means each convolutional layer in the network can detect more complex features. As the feature expands, we need to expand the dimension of the convolutional layer. In precise mathematics, the convolution procedure can be expressed as an equation:

$$y_{i^{l+1},j^{l+1},d} = \sum_{i=0}^{H} \sum_{j=0}^{W} \sum_{d^l=0}^{D^l} f_{i,j,d^l,d} \times x^l_{i^{l+1}+i,j^{l+1}+j,d^l} \cdot$$

2) *CNN Architecture:* A CNN usually takes an order 3 tensor as its input, e.g., an image with H rows, W columns, and 3 channels (R, G, B color channels). Higher order tensor inputs, however, can be handled by CNN in a similar fashion. The input then sequentially goes through a series of processing. The abstract description of the CNN structure is provided below:

$$\boldsymbol{x}^1 \longrightarrow \boxed{\boldsymbol{w}^1} \longrightarrow \boldsymbol{x}^2 \longrightarrow \cdots \longrightarrow \boldsymbol{x}^{L-1} \longrightarrow \boxed{\boldsymbol{w}^{L-1}} \longrightarrow \boldsymbol{x}^L \longrightarrow \boxed{\boldsymbol{w}^L} \longrightarrow z$$

This CNN runs layer by layer in a forward pass. The input is x^1, usually an image (order 3 tensor). It goes through the processing in the first layer, which is the first box. We denote the parameters involved in the first layer's processing collectively as a tensor w^1. The output of the first layer is x^2, which also acts as the input to the second layer processing. This processing proceeds till all layers in the CNN has been finished, which outputs x^L. One additional layer, however, is added for backward error propagation, a method that learns good parameter values in the CNN.

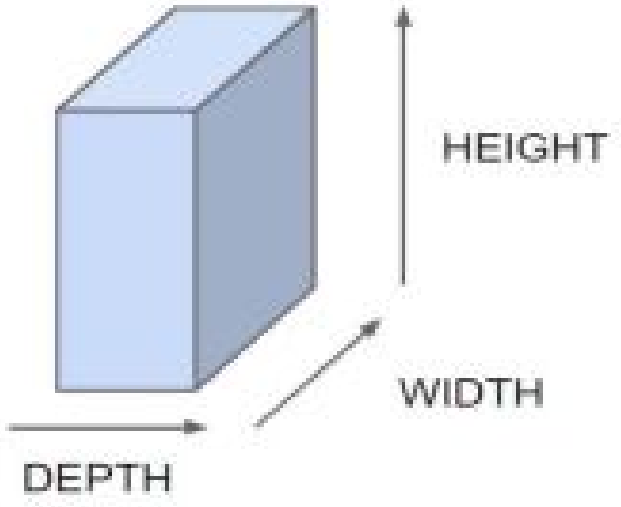


Fig. 1 3-D diagram for a convolution network

©IJRASET: All Rights are Reserved

1) *Softmax:* The softmax function is a function that turns a vector of K real values into a vector of K real values that sum to 1. The input values can be positive, negative, zero, or greater than one, but the softmax transforms them into values between 0 and 1, so that they can be interpreted as probabilities. If one of the inputs is small or negative, the softmax turns it into a small probability, and if an input is large, then it turns it into a large probability, but it will always remain between 0 and 1. It always "returns a probability distribution over the target classes in a multiclass classification problem The below diagram shows softmax o/p layer for a 5-class problem:

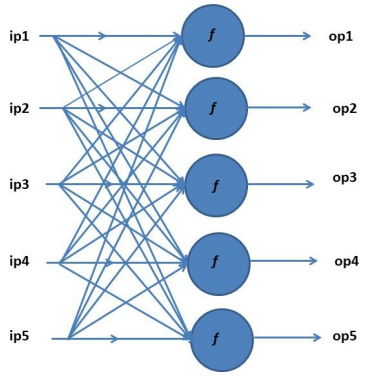


Fig. 4 Softmax o/p layer for 5-class problem

The softmax formula is as follows:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

The softmax is essential when we are training a neural network, without an activation function a neural network is a simple regression model. The softmax function provides non linearity to the neural network. The variants of softmax are Full softmax is a softmax that calculates a probability for every possible class. The other option is candidate sampling means that Softmax calculates a probability for all the positive labels but only for a random sample of negative labels.

2) *Gradient Decent:* Gradient Discent is an optimization algorithm used to find the values of parameters(coefficients) of a function(f) that minimizes a cost function(cost). Gradient descent is best used when the parameters cannot be calculated analytically (e.g. using linear algebra) and must be searched for by an optimization algorithm. Gradient descent is a way to minimize an objective function J(θ) parameterized by a model's parameters θ ∈ R d by updating the parameters in the opposite direction of the gradient of the objective function ∇θJ(θ) w.r.t. to the parameters. The learning rate η determines the size of the steps we take to reach a (local) minimum.

There are three gradient Discent Varients which differ in how much data we use to compute the gradient of the objective function.

a) *Batch Gradient Discent:* Vanilla gradient descent, aka batch gradient descent, computes the gradient of the cost function w.r.t. to the parameters θ for the entire training dataset:

$$\theta = \theta - \eta \cdot \nabla\theta J(\theta)$$

b) *Stochastic Gradient Discent:* Stochastic gradient descent (SGD) in contrast performs a parameter update for each training example $x^{(i)}$ and label $y^{(i)}$

$$\theta = \theta - \eta \cdot \nabla\theta J(\theta; x^{(i)}; y^{(i)})$$

c) *Mini batch Gradient Discent:* Mini-batch gradient descent finally takes the best of both worlds and performs an update for every mini-batch of n training examples:

$$\theta = \theta - \eta \cdot \nabla\theta J(\theta; x^{(i:i+n)}; y^{(i:i+n)})$$

©IJRASET: All Rights are Reserved

E. Predict the Response

After the model is built we can use that to predict the response for given message by user. Here we have used predict() method to predict the responses list. It will remove some of the responses based on the error threshold and the probability of becoming the response to the given message by user. We have created a UI by using Tkinter library which is shipped with tons of useful libraries for GUI. We will take the input message from the user and then use the helper functions we have created to get the response from the bot and display it on the GUI.

IV. TESTING AND RESULTS

Below is the screenshot while training the model with 200 epochs and it will give 98% **.**

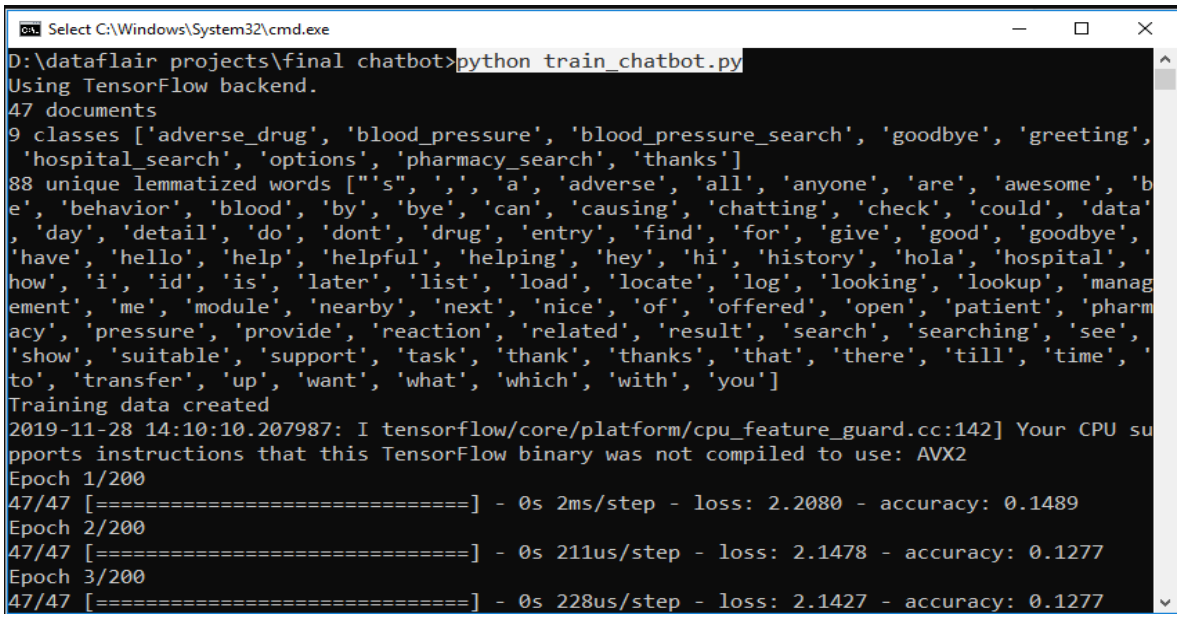


Fig. 5 Training of Dataset

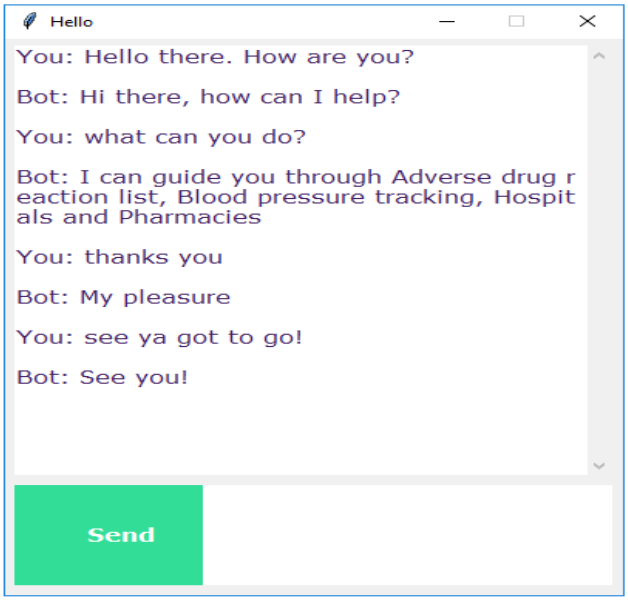


Fig. 6 Output of Chatbot

Above is the screeshot which will show the GUI of chatbot.

©IJRASET: All Rights are Reserved

V. CONCLUSIONS

The proposed chatbot, can able to help the people in this pandemic situation to know about the availability of beds, medicines and other required details without depending on other people. It is not only used for knowing hospital details, by changing the dataset we can use this chatbot for any other purposes too. we can customize the data according to business requirements and train the chatbot with great accuracy. Chatbots are used everywhere and all businesses are looking forward to implementing bot in their workflow.

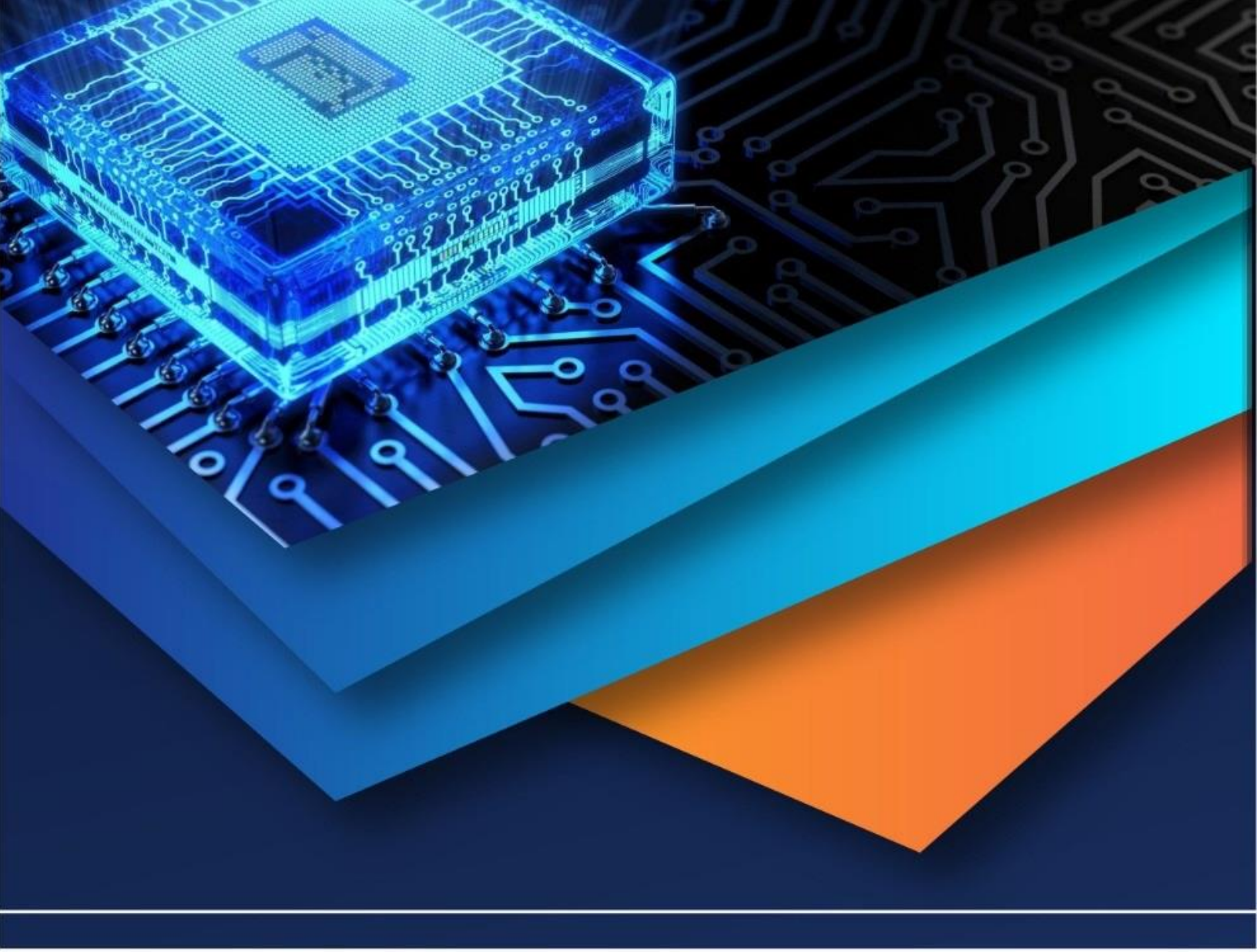
VI. ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to T Divakar Subramaniam, Senior Staff Manager Micron, India for his valuable guidance and constant support, along with her capable instructions and persistent encouragement. We are grateful to our class teacher in undergraduate Smt M. Trupthi, for his steady support and provision of every resource required for the completion of this project. We would like to take this opportunity to thank Micron Technology, India GD, for having designed an excellent learning atmosphere along with work.

REFERENCES

[1] Anupam Mondal; Monalisa Dey; Dipankar Das; Sachit Nagpal; Kevin Garda. “Chatbot: An automated conversation system for the educational:,IEEE.
[2] Biswas, D. (2020). Privacy Preserving Chatbot Conversations. 2020 IEEE Third International Conference on Artificial Intelligence and Knowledge Engineering (AIKE).
[3] Astuti, W., Putri, D. P. I., Wibawa, A. P., Salim, Y., Purnawansyah, & Ghosh, A. (2021). Predicting Frequently Asked Questions (FAQs) on the COVID-19 Chatbot using the DIET Classifier. 2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT).
[4] Lertpiya Anuruth; Chaiwachirasak Teerapat; Maharattanamalai Nattasit; Lapjaturapit Theerapat; Chalothorn Tawunrat; Tirasaroj, Nutcha; Chuangsuwanich, Ekapol (2018). [IEEE 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP) - Pattaya, Thailand (2018.11.15-2018.11.17)] 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP) - A Preliminary Study on Fundamental Thai NLP Tasks for User-generated Web Content.
[5] Shi, N., Zeng, Q., & Lee, R. (2020). Language Chatbot–The Design and Implementation of English Language Transfer Learning Agent Apps. 2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE).
[6] Handoyo, Eko; Arfan, M.; Soetrisno, Yosua Alvin Adi; Somantri, Maman; Sofwan, Aghus; Sinuraya, Enda Wista (2018). [IEEE 2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE) - Semarang (2018.9.27-2018.9.28)] 2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE) - Ticketing Chatbot Service using Serverless NLP Technology.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)