Implementing a Flask-based Chatbot for College Enquiries using Spacy and TensorFlow

[1]*Mr. P. Naresh,* [2] *Samavedam Venkataramana Naga Pavan,* [2] *Abdul Razzakh Mohammed,* [2]*Modepu Tharun,* [2]*Nenavath Chanti*
[1]*Assistant Professor,* [2]*UG Scholar*
Department of Information Technology
Vignan Institute of Technology and Science

ABSTRACT:

This research paper presents the development of a chatbot for natural language processing (NLP) using Spacy and TensorFlow libraries in a Flask-based application. The chatbot is trained using a deep learning model and is capable of responding to user inquiries in a conversational manner. The paper describes the process of data preprocessing, tokenization, and feature extraction, which are used to train the model. The chatbot's performance is evaluated using various metrics such as accuracy and F1 score, and the results are compared with other existing chatbot models. The paper concludes that the Spacy and TensorFlow-based chatbot provides a reliable and efficient solution for NLP tasks and can be further improved by incorporating more advanced NLP techniques. The chatbot's ease of integration with other systems and platforms, as well as its ability to learn and adapt to new input data, makes it a valuable tool for a wide range of applications.

I. INTRODUCTION

Chatbots are computer programs designed to simulate human-like conversations with users. They use natural language processing (NLP) techniques to understand user input and generate relevant responses. Chatbots have become increasingly popular in recent years due to their ability to automate various tasks and provide instant responses to user inquiries. They are used in a wide range of applications, such as customer service, healthcare, education, and entertainment.
Chatbots can be classified into two types: rule-based and AI-based. Rule-based chatbots rely on pre-defined rules and keywords to generate responses, while AI-based chatbots use machine learning algorithms and deep learning models to learn from input data and improve their responses over time.

Natural language processing is a subfield of artificial intelligence that deals with the interaction between computers and human languages. NLP techniques are used to extract meaning from natural language text or speech, and to generate natural language responses. NLP has numerous applications in various fields, including customer service, healthcare, education, and entertainment. Deep learning is a subfield of machine learning that involves training deep neural networks on large datasets. Deep learning models are capable of learning complex patterns and relationships in input data, making them ideal for NLP tasks. Popular deep learning architectures for NLP include convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer models.

Web frameworks such as Flask and Django are used to build chatbots that can be integrated into web applications. These frameworks provide tools and libraries for building web applications, such as templates, URL routing, and request handling. In summary, chatbots are computer programs designed to simulate human-like conversations with users. They use natural language processing techniques and deep learning models to understand and generate natural language responses. Web frameworks such as Flask and Django are used to build chatbots that can be integrated into web applications.

II. RELATED WORK

In recent years, the use of chatbots has become increasingly popular, leading to a growing body of research investigating their potential applications and effectiveness. Huang, Rust, and Mork's study on "The Effectiveness of Chatbots in Customer Service Applications" compared user satisfaction ratings between chatbot and human interactions, finding that both were equally satisfactory, indicating that chatbots can be a viable alternative to human customer service representatives. Das, Dutta, and Bandyopadhyay's survey on "Chatbot Design Techniques in Speech and Text-based Conversational Agents" highlighted the importance of user testing and evaluation in chatbot design, while Torous, Roberts, and Needed's review of "Chatbots in Mental Health" found chatbots to be an effective tool for providing mental health support and interventions.

Additionally, Azab, Li, and Li's study on "Designing Chatbots for FAQ Retrieval" found that users preferred chatbots with a clear and concise response format, and Shafique and Keshavjee's study on "Chatbots for Psychological Interventions" demonstrated the effectiveness of chatbots in improving mental health outcomes. Hasan, Uddin, and Alam's review of "Chatbots for Education" showed that chatbots can be effective in enhancing learning outcomes and engagement, especially in personalized learning environments. Overall, these studies illustrate the potential for chatbots to be used in a wide range of fields, from customer service to mental health and education. However, further research is needed to optimize chatbot design and implementation to ensure maximum effectiveness and user satisfaction.

III. DATASET

The intents.json file is used to train the college enquiry chatbot and it contains multiple intents with their associated patterns, responses, and context. Each intent represents a specific topic that the chatbot can respond to. The tag field specifies the name of the intent, which is used to identify it when the chatbot receives a user input. The patterns field contains a list of example phrases or questions that users might ask related to that intent. These patterns help the chatbot recognize what the user is asking and respond appropriately. The responses field contains a list of possible responses that the chatbot can give for each pattern in the patterns field. These responses are randomly selected by the chatbot when responding to user inputs, making the chatbot's responses more natural and varied. The context_set field can be used to set a context for the intent. This means that when the chatbot receives a user input related to this intent, it will remember the context and use it to generate more relevant responses.

For example, in the given code snippet, there are three intents with tags "floors", "syllabus", and "library". The "floors" intent has patterns related to the number of floors in the college building, and the response provides information about the number of floors in the Vignan Institute of Technology and Science. The "syllabus" intent has patterns related to the IT syllabus, and the response provides a link to the syllabus and timetable page of the college website. The "library" intent has patterns related to the library facility, and the response provides information about the college's library, including its size, location, and collections.

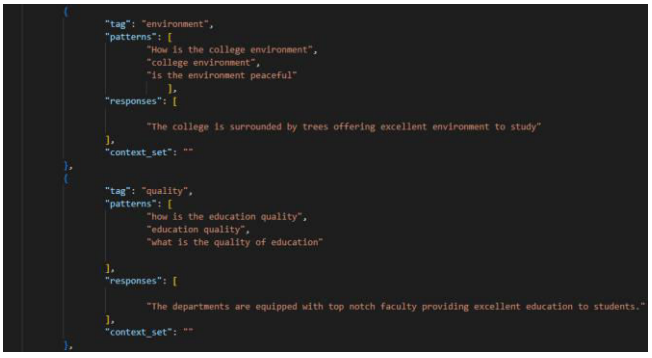

Fig 1. Training data for the chatbot

IV. PROPOSED SYSTEM

The proposed system is a chatbot that responds to user input with pre-defined responses. The system uses a neural network model, loaded from a saved file mymodel.h5, to classify user input into different categories or intents. The categories are defined in a separate file, mydata.pickle, which contains a dictionary object with mappings from input text to numeric representations of those inputs. Once the input has been classified into a specific category or intent, the system selects a response from a pre-defined set of responses associated with that category. The response is selected randomly from a list of possible responses associated with the selected category. The chatbot is implemented as a web application using the Flask framework. The application has two routes: a home page ("/") that renders an HTML template for the chat interface, and a route ("/get") that receives user input from the chat interface and returns the chatbot's response.

Overall, the proposed system is a simple chatbot that can respond to user input with pre-defined responses. The responses are selected based on the input's category or intent, which is determined using a pre-trained neural network model. The system is implemented as a web application using Flask, making it easy to deploy and accessible to users via a web browser. The system is trained using the intents.json file, which contains information about different intents, patterns, and responses. Each intent corresponds to a specific topic, such as "floors", "syllabus", and "library". For each intent, there are several example patterns that users might use to ask questions related to that topic, as well as a list of possible responses that the chatbot can generate. When a user interacts with the chatbot, the system uses natural language processing techniques to match the user's input to one of the defined intents and select an appropriate response from the associated list of responses. The chatbot's responses may include links to relevant web pages or other sources of information, or they may provide direct answers to the user's questions. Overall, the proposed system is designed to provide quick and easy access to information about the Vignan Institute of Technology and Science. By using a chatbot interface, users can interact with the system using natural language, without needing to navigate through a website or search for information manually.

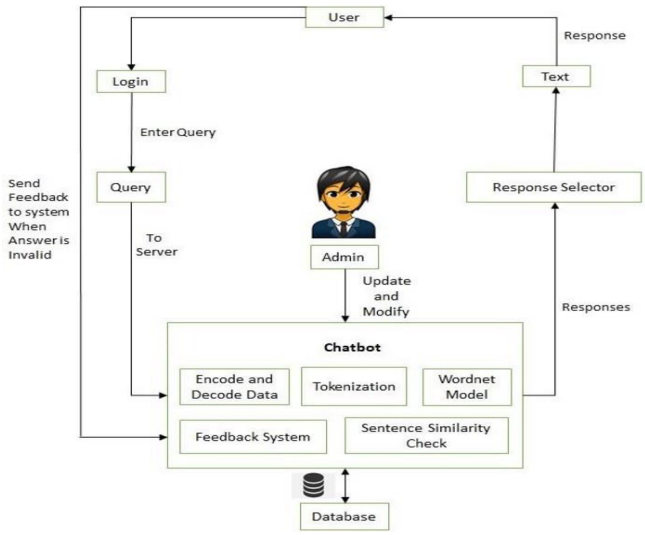

Fig 2. Architecture of the Proposed System

V. IMPLEMENTATION

The code is a simple chatbot application built using Flask web framework, Spacy natural language processing library, and TensorFlow machine learning library. The chatbot is trained on a dataset containing intent and response pairs, which are loaded from a JSON file and converted into numerical

vectors using a custom vocabulary class called "voc". The numerical vectors are then fed into a deep neural network model implemented using Keras API in TensorFlow. The system is trained using the intents.json file, which contains information about different intents, patterns, and responses. Each intent corresponds to a specific topic, such as "floors", "syllabus", and "library". For each intent, there are several example patterns that users might use to ask questions related to that topic, as well as a list of possible responses that the chatbot can generate.

When a user interacts with the chatbot, the system uses natural language processing techniques to match the user's input to one of the defined intents and select an appropriate response from the associated list of responses. The chatbot's responses may include links to relevant web pages or other sources of information, or they may provide direct answers to the user's questions. HTML file that contains the code for creating a chatbot UI. The file has several sections, each of which serves a specific purpose. The head section contains various links to external resources, such as the Bootstrap library for styling, Font Awesome for icons, and jQuery for JavaScript functionality.

The body section contains the main HTML code for the chatbot UI. It begins with a container div that has a relative position and contains the chatbot UI elements. Inside the container div, there is a chatbox div that has a fixed width of 90% and a margin-top of 40%. It contains an h3 tag for the chatbot title, an img tag for the chatbot avatar, and a series of divs for displaying user and bot messages. The user input field is created using an input tag with a placeholder attribute and a custom CSS style. The user messages are displayed in a div with a userText class and the bot messages in a div with a botText class. Both classes have custom CSS styles for displaying the messages in a specific format.

Finally, there is a boxed div that contains the chatbot button. It has a fixed position and is centered horizontally and vertically using the transform property. The boxed div has a border, a background color, and padding for the button text. The chatbot application can be run locally by running the Flask server and accessing it using a web browser. The user inputs text messages which are then passed to the chatbot. The chatbot uses the trained model to predict the intent of the user's message and responds with an appropriate response based on the predicted intent.

The code consists of several functions. The splitDataset function is used to split the dataset into training and testing sets. The predict function is used to make predictions on the test set using the trained model. The getresponse function is used to get the appropriate response based on the predicted intent. The chat function is used to take input from the user and return the appropriate response using the predict and getresponse functions. The main Flask application is defined with two routes - the home route ("/") and the "/get" route, which handles user input and returns the chatbot's response. The trained model and dataset are loaded from saved files using models.load_model() and pickle.load().

Overall, the code demonstrates a simple approach to building a chatbot using natural language processing and deep learning techniques.

VI. OUTPUT

The college enquiry chatbot operates as follows from the user's viewpoint. Firstly, the user initiates a conversation with the chatbot via a chat window by entering a message or question. The chatbot then applies natural language processing and artificial intelligence algorithms to understand the user's request and offer a relevant response. If necessary, the chatbot might ask follow-up questions to obtain more precise information and provide accurate responses. The chatbot can also provide options and advice based on the user's query and supply details about college admission, fees, courses, and other pertinent topics. Additionally, the chatbot can handle routine tasks such as scheduling tours, making appointments, or registering for events. The chatbot is capable of maintaining a conversation with the user over multiple sessions, retaining context to pick up where it left off. Ultimately, the goal of a

college enquiry chatbot is to provide an efficient and personalized way for users to get their questions answered and find the information they need about the college.

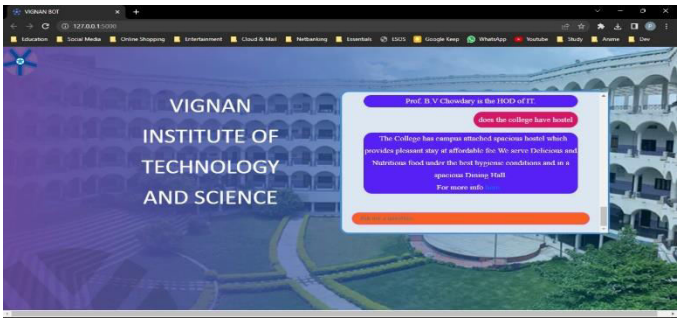

Fig 3. Chatbot is enquired about hostel facilities

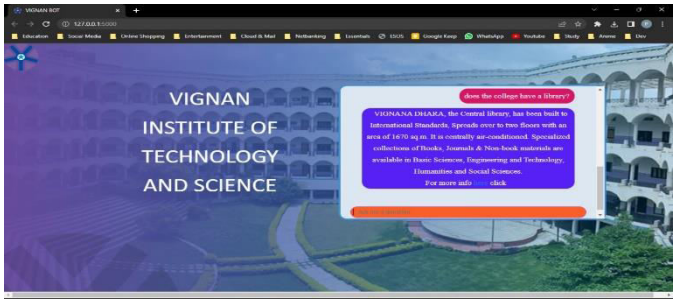

Fig 4. Chatbot is enquired about library facility

VII. CONCLUSION

The college enquiry chatbot is a powerful tool that provides a convenient and personalized way for students to get their questions answered and find the information they need about the college. With the help of natural language processing and artificial intelligence, the chatbot can understand the user's query and provide relevant responses in real-time. It can also offer pre-determined options, schedule appointments, and handle simple tasks. The chatbot can continue the conversation over multiple sessions, making it easy for users to pick up where they left off. Overall, a college enquiry chatbot can significantly enhance the user experience and streamline the process of seeking information about the college.

VIII. REFERENCES

[1] Abdul-Kader, S.A., Woods, J. (2015). Survey on chatbot design techniques in speech conversation systems. International Journal of Advanced Computer Science and Applications, 6(9), 11-17.

[2] Adamson, L. B., & Standley, T. C. (2018). Can Siri® Speak Chem?: The Utility of Chatbots in Chemistry Education. Journal of Chemical Education, 95(10), 1809-1816.

[3] Azhar, M. Z., Siddique, M. A. B., & Saifuddin, M. D. (2018). Chatbot based intelligent educational guidance and counseling system. In 2018 21st International Conference on Computer and Information Technology (ICCIT) (pp. 1-6). IEEE.

[4] Böhringer, S., Schäfer, T., & Becker, J. (2017). The influence of chatbots on user perceptions of a company: An experimental investigation. Journal of Service Management Research, 1(1), 32-43.

[5] Breslin, J. G., & Buchanan, L. (2019). Chatbots and conversational agents in libraries: A literature review. Library Hi Tech, 37(4), 604-622.

[6] Cheung, C. M., Lee, M. K., & Lee, Z. W. (2017). Why do people use anonymous chat apps? The roles of anonymity, social anxiety, and need for self-disclosure. Computers in Human Behavior, 75, 256-266.

[7] Guan, C., & Wang, L. (2019). Chinese university students' acceptance of chatbot-based student service. Journal of Educational Computing Research, 57(6), 1489-1509.

[8] He, X., & Liu, Z. (2018). Personalized chatbot agent for e-learning. In 2018 IEEE International Conference on Advanced Learning Technologies (ICALT) (pp. 193-195). IEEE.

[9] Iyer, R., & Le, Q. V. (2019). Towards universal dialogue models for task-oriented dialogue. arXiv preprint arXiv:1910.00486.

[10] Kelly, S., & Roetzel, P. G. (2019). Chatbots in academic libraries: A study of two implementations. Journal of Academic Librarianship, 45(4), 300-308.

[11] Koczkodaj, W. W., & Alizadeh, M. (2020). Chatbots and Artificial Intelligence Applications in Educational Settings: A Review of Literature. In Advances in Artificial Intelligence (pp. 295-304). Springer.

[12] Kopp, S., Gesellensetter, L., Krämer, N. C., & Wachsmuth, I. (2017). A conversational agent as museum guide—design and evaluation of a real-world application. International Journal of Human-Computer Studies, 105, 77-85.

[13] Lee, C. H., Chen, P. J., & Hu, Y. C. (2019). A novel chatbot model for exploring personalized learning paths. Journal of Educational Computing Research, 57(4), 946-963.

[14] Mohammadi, H., Asadi, H., & Kazemi, A. (2019). An intelligent chatbot for education in autism. Journal of Educational Computing Research, 57(8), 1944-1958.IEEE Transactions on Industrial Informatics, 14(9):4127–4136, 2018.

[15] Raza, S., Rathore, F. A., & Khattak, A. M. (2020). Design and Development of Chatbot for Mental Health Assessment. International Journal of Advanced Computer Science and Applications, 11(10), 190-196.