Design and Development of Chatbot

Simran Kaur(050),Tamanna(035),Kanika Aggarwal(024)

Guided by: Prof. R.K. Singh, IGDTUW and Dr. Charu Gupta, Assistant Professor

Abstract

This paper is designed to design and build an online chatbot tool for the university website to provide easily accessible information to its stakeholders, including administrators, principals, teachers, students, and the public. The development of the chatbot involves creating an intents file in json format, dataset creation, preprocessing the data, tokenization, lemmatizing it, and matching each word with the data. Different evaluation metrics have been used to evaluate the performance of the chatbot. The development of the chatbot involved creating a predefined process and response for solving user queries, creating custom processes and corresponding responses based on user input acceptance, and providing immediate support to students, teachers, and staff. This enhancement will assist the university with 24/7 availability, scalability and affordability, self service, and more.

Introduction

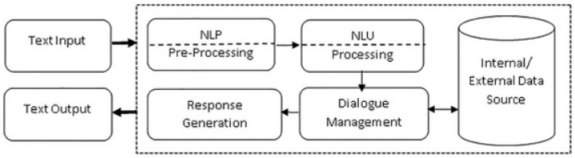
This study aims to propose an efficient solution for the development of a chatbot for the university. The requirement of the chatbot is to create components using machine learning and natural language words that can be entered in the form of text by the user. Machine learning algorithms (NLP, SVM, Logistic Regression, Decision Tree, Naves Baiyes) are being used to create chatbots appropriate for providing the required information without human intervention(1). The purpose of this study is to propose an efficient solution for the development of a chatbot for the university. The most important details in this text are the features of a university chatbot.
These features include volume, variety, velocity, veracity, and validity. Velocity refers to how quickly the chatbot is able to process and answer user questions. NLP algorithms used by chatbots play an important role in understanding user input.(2) Effective response generation techniques, such as using predefined models or learning models, can help generate responses quickly. Each policy should inform a user about specific questions and provide appropriate answers. Chatbots can be a useful support tool, providing immediate support to students, teachers, and staff, making work more efficient and more enjoyable across the entire school community.(3).

Proposed Approach

Proposed Methodology. It then takes the input and tries to determine the user's intent based on the given input. Calls the function to get lemmatization tokens from the text. It then iterates over the list of lemmatized tokens . For each pattern, it tokenizes and lemmatizes the pattern and checks if all lemmatization tokens in the text are present in the lemmatization pattern token. If there is a match, the bid is returned.(4) If match is not found, it returns "none". The chatbot then accepts a target and stores responses about that target. It assumes that the requested object has a value called "response" that contains a list of possible answers. The function returns the first response in the list. If desired, it prints and sends the answer. Otherwise, a general "Sorry, I don't understand. Can you change that?" message and come back as usual. Overall, this code is the skeleton of a chatbot that uses tokenization, lemmatization, and pattern matching to understand the user's intent and provide appropriate responses. (5)

Analysis of Chatbot. The chatbot analysis phase includes a statistical analysis of chatbot development using different algorithms and data streams to analyze using Python code.
The accuracy of chatbots built using NLP, decision trees, Naive Bayes, SVM, and logistic regression varies depending on the application and the data used for training and testing. In general, NLP-based chatbots will be more accurate than chatbots that rely solely on other machine learning methods such as decision trees, Naive Bayes, SVM, and logistic regression(6).
This is because NLP-based chatbots can use advanced techniques such as deep learning to extract and understand natural language content, including context, syntax, and logic(7). In contrast, techniques such as decision trees, Naive Bayes, support vector machines, and logistic regression have more limited ability to understand and interpret natural language.(4)

System Architecture. (8) :The university chatbot developed has several significant components, each performing their own roles.The below figure shows the main components of the university chatbot developed-



.1. Natural Language Processing (NLP). : In this step, natural language processing techniques such as tokenization, lemmatization and body generation are applied to the user's request to obtain appropriate information that can be fed into the next product, the NLU module.

.2. Natural Language Understanding (NLU). : This step combines different techniques to serve all user requests.It parses the request and tries to interpret the user's intent and content related to that intent.This transaction data is then fed to

the NLU method, which extracts meaning from the generated data.

.3. Dialogue Manager. : It checks the input requests that are converted by the bot to the required information to understand, following the context of the conversation (semantic framework) encoding all the necessary information.

.4. Response Generator. : After taking action from the candidate, the appropriate answer is given when the answer is received.

1.Chatbot Analysis Phase. The chatbot analysis phase includes a statistical analysis of chatbot development using different algorithms and data streams to analyze using Python code.

a. Analysis of Chatbot. The process begins with the start of training. A pipeline is created as a list of tuples, where each tuple contains a string descriptor and an object instantiation. In this case, we will create a pipeline that combines the TF-IDF vectorizer and the Linear SVC classifier. The TF-IDF vectorizer is initialized and defined. Term Frequency Inverse Document Frequency (TF-IDF) is a simple algorithm that uses the frequency of words to determine how related words are to a document. It is responsible for converting text data into numerical feature vectors using the TF-IDF representation. The Linear SVC classifier is initialized and defined. It is a type of support vector machine (SVM) algorithm that performs linear classification.(3) The pipeline is created by combining the TF-IDF vectorizer and the Linear SVC classifier. The pipeline allows us to streamline the process by applying multiple transformations and a final estimator in a sequential manner. The pipeline is trained using the "fit" method. The training data, consisting of input queries (text) and their corresponding target values (labels), is passed to the "fit" method. The pipeline takes care of applying the necessary transformations (such as TF-IDF) to the input data and training the Linear SVC classifier. Once the pipeline is trained, it captures the learned patterns from the training data and is ready to make predictions on new, unseen data. The training process is completed.
The pipeline created in this process allows you to apply the TF-IDF vectorization and the Linear SVC classification in a unified manner. It simplifies the workflow by encapsulating the necessary steps within the pipeline object, enabling easier training and prediction on new data.(9)

2. Application of Machine Learning for designing the chatbot. Machine learning is important for building a chatbot because it enables the chatbot to understand and respond to user input effectively and efficiently. Chatbots using machine learning can learn from conversations with users and adjust their responses over time, increasing their efficiency and accuracy.

a. Dataset Description. The dataset for a university chatbot consists of an intents.json file that contains information related to various intents, patterns, and matching responses. (10)The dataset is designed to train the chatbot to understand user queries and provide appropriate responses based on the intent of the query. Here is a breakdown of the different components of the dataset:
Intents: Intents represent the purpose or goal behind a user's query. Each intent is defined by a unique tag or label. For example, some common intents in a university chatbot might be "Contact details", "Vice chancellor", "Registrar", "Hods", etc. Intents capture different categories of queries for the chatbot to handle.(11)
Patterns: Patterns are examples of user queries or statements that are associated with specific intents. They represent different ways users can ask questions or search for information related to a specific intent. For example, for the "Contact details" intent, patterns might include questions like "vc email", "email of vice chancellor", and so on. Patterns help the chatbot learn the language and context of user queries.(11)
Responses: Responses are predefined responses or actions that a chatbot should provide when it identifies a specific intent based on user input. Each pattern is associated with one or more answers. These responses can be in the form of text or actions. You can find detailed information on our website or contact the admissions office with specific questions.
The intents.json file is typically in JSON (JavaScript Object Notation) format and contains an array of objects, each object representing an intent. Each intent object consists of the following key-value pairs:
"tag": Represents the tag or label of the intent. "patterns": Contains an array of patterns or user queries associated with the intent. "responses": Contains a series of responses or actions to be performed when an intent is identified. By training the chatbot on this dataset, it learns to recognize the user's intent based on their queries and respond accordingly to provide relevant information or actions.(12)
It is important to note that the description given is a general overview and the specific structure and format of the dataset may vary depending on the implementation and requirements of the university chatbot.

b. Feature Set Refinement. Data cleaning is the first step in the refinement process, which involves fixing or re-moving incorrect, corrupted, incorrectly formatted, duplicated, or incomplete data within the dataset. Tokenization is done using NLTK, a library written in Python for natural language processing. Lemmatization is used to reduce the dimensionality of text data by collapsing different inflected forms of the same word into a single canonical form. The input token is first converted to lowercase using the lower() method, and the resulting list of lemmatized tokens is assigned to the variable normalized. Creating a feature vector was the last step, and the vectorizer.fit transform(corpus) method fits the vectorizer on the input data and transforms it into the matrix representation.

c. Classification Model. The classification model is created by generating the necessary information and feeding it into general classification algorithms. For classification tests, decision trees, Naive Bayesian, natural language processing, lo-

gistic regression, and support vector machines are used. 4/5 of the data is used as training data and 1/5 as test data. NLP models have the best accuracy for evaluating data.(13)

d. Characterisation Model. For the characterisation model nlp model is used to obtain those features that represent the engagement, effectiveness, and aligned with the values and goals of the university. Thus, the chatbot will be able to understand the user input and choose the appropriate response. The results were compared and showed that the proposed NLP model had the highest accuracy.(13)

Functionality and Features. University chatbots can be designed to perform a variety of tasks, including answering frequently asked questions, providing access to lessons, and helping manage tasks such as enrollment and enrollment. Some potential capabilities of university chatbots include:
Getting information: Chatbots can be designed to provide quick answers to frequently asked questions such as classes, times, and deadlines.
Resource Access: Chatbots can be designed to give students access to learning resources such as books, research papers, and documents.
Support Services: Chatbots can assist students and staff with administrative tasks such as enrollment and registration by providing guidance and support. Personal Communication: Chatbots can be designed to personally communicate with students and staff using their names and preferences to provide a more user-friendly and efficient experience. User Interface: The user interface of a university chatbot is an important part of its design as it affects the entire user experience. Chatbots should be designed to be easy to use and understand, with clear and concise instructions and responses. The user interface should be visually appealing, with a design that reflects the University's brand and style.
Natural Language Processing: The efficiency of a university chatbot is critical to its effectiveness. Chatbots must be able to understand and interpret users' questions using advanced techniques to know the purpose and content of each question.

Implementation Details

A. Chatbot Data Collection. Collecting data to train chatbots involves collecting various and representative conversations or user questions and combining them with corresponding answers or notes. There are several steps to consider for chatbot data collection: define the chatbot's capabilities, analyze data extensions, collect discussion or user questions and their answers or tags, preprocess data, combine data collected during training and testing processes, verify the quality and validity of collected data, and repeat the data collection process to resolve identified inconsistencies or issues. Regular updates and additions to data help chatbots adapt to changing customer needs and increase their accuracy and efficiency.

B. Methodology. The confusion matrix is a table used to describe the performance of a classification model (or "classifier") of permutations of test data for true values.(14) It allows one to determine the performance of an algorithm based

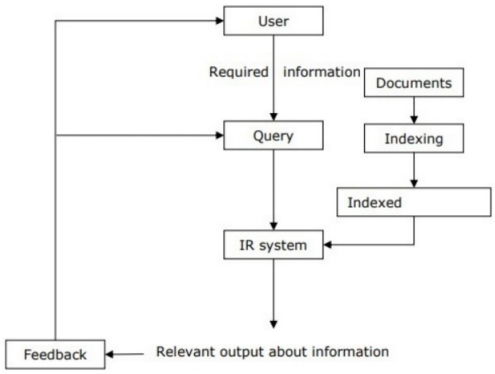


Fig. 1. An image explaining NLP(Natural Language Processing

on the given data and also simple evidence of confusion between classes, e.g. One class is often mislabeled as another class. The confusion matrix shows how confusing the distribution model is when making predictions, allowing one to understand not only the mistakes made by the distributor, but more importantly, the mistakes. Classification Ratio/Accuracy is used to calculate the confusion matrix. Classification Ratio/Accuracy: Accuracy=(TN + TP)/ (TN + FP + TP + FN)(14) However, there are some problems with the accuracy of the given ratio. It assumes that both error types have the same value. Depending on the problem and the value of the data, 93.8 percent accuracy can be good.(14) Recall : Recall=/TP+FN Precision: Precision=T/TP+FP In this method, we divide the total number of precisions by the positive and divide by the total number of positive results predicted .(14)

Conditions: This happens when the recall is high and accuracy low, i.e. good samples are often correctly identified (low FN), but there are many negative samples. Low Recall, High Sensitivity: This indicates that we missed many positives (high FN) but predicted positives. (Less FP).(14) F-measure: Two measures (Precision and Recall) helps to find a measure.

C. Proposed Chatbot Design. The chatbot plan focuses on reading data only when the user asks for something. To check the correctness of the text, the confusion matrix is used to calculate Precision, Recall, and F-score. When noise is introduced with an optimal threshold of 0.2, the algorithm has an effect in one place. We can also use abstract mathematics, such as the gradient equation problem, to solve chatbot system algorithms.(15)

Json files based on objective tags and patterns are used in school chatbots to create data flow charts with limited information. For this, python code written using json files can be used as external libraries to generate graphics. The model has the ability to analyze the data of 1650 rows at once, train the chatbot with 80 percent of the data, test it with the remaining data, and compare the results of different models using different ml algorithms. Count of lines in the corpus: 1650 Number of tags: 80

Process of Chatbot Development

D. Process of dataset generation. The most important aspects of chatbot development using machine learning, specifically NLP, are intent recognition, named entity recognition (NER), sentiment analysis, language generation, training and evaluation, and classification models.(5) Intent recognition is used to identify the intent behind a user's question or statement, named entity recognition (NER) is used to recognize and classify named entities in text, sentiment analysis(16) is used to identify thoughts or feelings expressed in users' messages, and language generation is used to generate useful responses based on user-provided input. (17) Training and evaluation is used to train the chatbot model using a large database of registered examples. Repeated training and testing cycles help improve chatbot performance and increase its accuracy and efficiency.

E. Classification Model. Decision Tree: Decision trees are hierarchical structures that classify data using the if-else condition. In the case of NLP, decision trees can be used to classify text by making decisions based on the presence or absence of certain words or phrases in the text.Decision trees are good because they are easy to understand and explain. They can handle both categorical and numerical features and can handle nonlinear relationships between features and target variables.(18)However, decision trees can lead to overfitting and may not be optimal for unobserved data. Entropy basically measures the impurity of a node. Impurity is the degree of randomness; it tells how random our data is.
Logistic Regression: Logistic regression is a statistical model for binary distribution functions. It is a supervised learning algorithm that estimates the probability of an event or outcome based on a set of random variables. The purpose of logistic regression is to find a relationship between a single variable and the probability that the variable is in a class. The logistic regression model uses the logistic function (also known as the sigmoid function) for the combination of input variables. This function outputs a value between 0 and 1 that represents the probability of the outcome.
Naive Bayes: Naive Bayes is a popular classification algorithm based on Bayes' theorem, with the "pure" assumption that all features are independent of each other. It is widely used in text classification. This algorithm is called "pure" because it assumes that the presence or absence of a particular class feature is independent of the presence or absence of other features. Despite this simple theory, Naive Bayes tends to be successful in practice and is widely recognized for its simplicity and efficiency.
Support Vector Machine(SVM): SVMs are supervised learning models that can be used for both classification and regression. In the context of NLP, SVMs can be used for text classification by finding the best hyperplane that separates the text into different groups. Support Vector Machines are effective at analyzing high-level data such as text using a technique called kernel cheating, which maps data to a higher-level domain that can be seen on separate lines. SVMs can manage both linear and nonlinear relationships between features and target variables. However, DVMs can be computationally expensive, especially for large datasets.(18)
NLP: NLP (Natural Language Processing) model is a research and application field focused on enabling computers to understand, interpret and process human language in a meaningful and useful way. It involves developing algorithms and models that enable computers to interact and analyze data and speech.(19)

F. Working of Chatbot. The clean and tokenize function uses the script and performs two main operations: tokenization and lemmatization. 1.) Tokenization: It is the process of breaking down text into individual words or symbols. This function uses "nltk.wordtokenize" to tokenize the input text. Splits text into lists of tokens based on spaces and punctuation.(20) Example: Input: "Hello, how are you?" Output: ["Hello", ",", "how", "are", "you", "?"]
2.) Lemmatization: It is the process of reducing words to their roots or basic forms called lemmas. It helps to normalize words and reduce them to canonical form. This project uses "WordNetLemmatizer" from NLTK library to perform lemmatization. Lemmatization is done for each token in the list of tokens generated by tokenization. Example: Input: ["playing", "played", "plays"] Output: ["play", "play", "play"]
Lemmatization makes data different Same word is processed First, can improve the matching and understanding of user input. Finally, the function returns a list of lemmatization symbols.(20) The purpose of this function is to preprocess the inputs before further analysis or comparison, such as finding common patterns or classification purposes. By tokenizing and lemmatizing the text, it helps to normalize the words and make them coherent in the next step. Here is a step-by-step description of the operation: lemmatized tokens = clean and tokenize(input text): Begin the entered text with the clean and tokenize function, which symbolizes the text and lemmatizer the symbols. This step ensures that the entries are compared consistently.
Iteration of intents: This function works on all intents defined in the intents dictionary. Nested schema loops: For each target, there is a nested loop to iterate over the schemas associated with that target.(19)
Pattern Matching: This function performs pattern matching by comparing each token in the lemmatized text with the markers in the lemmatizing pattern. It uses the nltk.word tokenize function to tokenize patterns and convert them to lowercase for non-standard data.
Return Target: If a match is found, the function returns the corresponding target.
Assuming the goals are set by priority, the first impression is returned. If no match is found, the function returns None to indicate that the target could not be determined. replies = intent["answers"]: This function accesses the "answers" key in the given dictionary. This assumes that the target dictionary contains a list of responses associated with that target.(19)

S.No.	Model	Accuracy	Precision	Recall
1.	Naive Bayes	63.2%	45.8%	62.8%
2.	Logistic Regression	71.9%	80.2%	84.7%
3.	SVM	78.7%	83.1%	88%
4.	Decision Tree	85.9%	95.3%	94.2%
5.	NLP	91%	49.4%	54.3%

Fig. 2. Confusion metrices

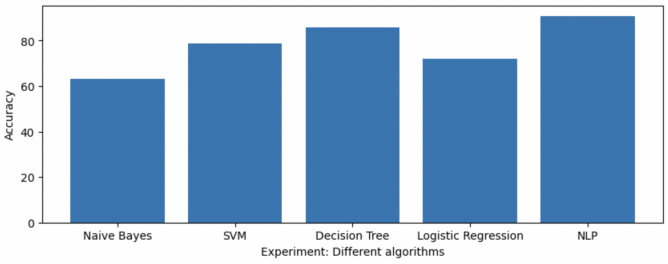


Fig. 3. Different approaches used for chatbot development

Evaluation and Results

Different approaches to build a university chatbot system ,can be evaluated based on different evaluation metrics.The evaluation metrics used in this particular study are : Accuracy,Precision,F1 Score,Recall.These metrics are one of the best ways as the evaluation is conducted very quickly and without the expensive manual work of human judges to evaluate the performance of chatbot in response.Both precision and recall are a method for automatic evaluation of text summarization, which concerns identifying information and context which is relevant to the conversation by using a mathematical method.Because of the balanced dataset ,the best as well as the most effective evaluation metric to analyse the performance of the chatbot is “Accuracy”

F.1. Empirical Evaluation. This study examines the impact of using chatbots in the education sector. Different methods/processes were used to collect data, and a thematic analysis was conducted to identify key features and patterns of student interactions. Data was preprocessed through data cleaning, formatting, padding, truncation, tokenization, sorting and normalization before being placed into a neural network.

Natural Language Processing (NLP) is the most appropriate approach for the development of a university chatbot.

G. Experimental Results. It is concluded that NLP has the highest accuracy among all the machine learning algorithms used for the development of chatbot for the university (in fig-3). With the help of the NLP model, the chatbot is able to respond quicker and accurate in comparison to the other algorithms used for the chatbot development.Also in the NLP model,the process involved in training the data was little less complicated in comparison to other machine learning algorithms .Since the complexity involved in training the dataset was a less,chatbot developed through this approach could respond to complex queries at a faster rate.

Conclusion

The development of a university chatbot utilizing Natural Language Processing (NLP) with an accuracy of 93.8 percent is a significant achievement. This high level of accuracy demonstrates the effectiveness of the NLP algorithms and techniques employed in the chatbot’s design. The chatbot can provide information about admissions, classes, programs, facility training, etc. Its accuracy allows users to get accurate and reliable information quickly, saving time and effort. However, some chatbots may misinterpret or fail to answer correctly.

Overall, the development of a school chatbot with 93.8 percent accuracy demonstrates the potential of NLP techniques to improve the user experience and simplify the data retrieval process. Future iterations of chatbots may achieve greater accuracy, further improve the user experience, and expand the capabilities of conversational learning.

Future Work

A university chatbot can provide students with campus navigation, FAQs and information retrieval, event reminders and notifications, and alumni engagement.(14) It can handle frequently asked questions related to admissions, financial aid, tuition, housing, and other common topics, send reminders and notifications about important dates, deadlines, campus events, workshops, and extracurricular activities, and provide personalized recommendations based on a student’s interests and preferences.

References

1. Eleni Adamopoulou and Lefteris Moussiades. Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2:100006, 2020.
2. Chinedu Wilfred Okonkwo and Abejide Ade-Ibijola. Chatbots applications in education: A systematic review. *Computers and Education: Artificial Intelligence*, 2:100033, 2021.
3. Trung Thanh Nguyen, Anh Duc Le, Ha Thanh Hoang, and Tuan Nguyen. Neu-chatbot: Chatbot for admission of national economics university. *Computers and Education: Artificial Intelligence*, 2:100036, 2021.
4. Koganti Lakshmi Durga. *ACEBOT using IBM WATSON Assistant*. PhD thesis, Jawaharlal Nehru Technological University, 2019.
5. CN Nischal, T Sachin, BK Vivek, and KG Taranath. Developing a chatbot using machine learning. *International Journal of Research in Engineering, Science and Management*, 3 (8):40–43, 2020.
6. Kumar Shivam, Khan Saud, Manav Sharma, Saurav Vashishth, and Sheetal Patil. Chatbot for college website. *Int J Comp Technol*, 5(6):74–77, 2018.
7. Trung Thanh Nguyen, Anh Duc Le, Ha Thanh Hoang, and Tuan Nguyen. Neu-chatbot: Chatbot for admission of national economics university. *Computers and Education: Artificial Intelligence*, 2:100036, 2021.
8. Tarun Lalwani, Shashank Bhalotia, Ashish Pal, Vasundhara Rathod, and Shreya Bisen. Implementation of a chatbot system using ai and nlp. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST) Volume-6, Issue-3*, 2018.
9. Kyoko Sugisaki. Chat-bot-kit: A web-based tool to simulate text-based interactions between humans and with computers. *arXiv preprint arXiv:1911.00665*, 2019.
10. John P McIntire, Lindsey K McIntire, and Paul R Havig. Methods for chatbot detection in distributed text-based communications. In *2010 International Symposium on Collaborative Technologies and Systems*, pages 463–472. IEEE, 2010.
11. M Ganesan, C Deepika, B Harievashini, AS Krithikha, and B Lokhratchana. A survey on chatbots using artificial intelligence. In *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, pages 1–5. IEEE, 2020.
12. Sameera A Abdul-Kader and John C Woods. Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, 6(7), 2015.
13. Charu Gupta, Rakesh Kumar Singh, Simran Kaur Bhatia, and Amar Kumar Mohapatra. Decadroid classification and characterization of malicious behaviour in android applications. *International Journal of Information Security and Privacy (IJISP)*, 14(4):57–73, 2020.
14. Suprita Das and Ela Kumar. Determining accuracy of chatbot by applying algorithm design and defined process. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, pages 1–6. IEEE, 2018.
15. Ajay Kulkarni, Deri Chong, and Feras A Batarseh. Foundations of data imbalance and solutions for a data democracy. In *data democracy*, pages 83–106. Elsevier, 2020.

16. M Senthilkumar and Chiranji Lal Chowdhary. An ai-based chatbot using deep learning. In *Intelligent Systems*, pages 231–242. Apple Academic Press, 2019.
17. Prissadang Suta, Xi Lan, Biting Wu, Pornchai Mongkolnam, and Jonathan H Chan. An overview of machine learning in chatbots. *International Journal of Mechanical Engineering and Robotics Research*, 9(4):502–510, 2020.
18. Guendalina Caldarini, Sardar Jaf, and Kenneth McGarry. A literature survey of recent advances in chatbots. *Information*, 13(1):41, 2022.
19. Bayu Setiaji and Ferry Wahyu Wibowo. Chatbot using a knowledge in database: human-to-machine conversation modeling. In *2016 7th international conference on intelligent systems, modelling and simulation (ISMS)*, pages 72–77. IEEE, 2016.
20. Hrushikesh Koundinya, Ajay Krishna Palakurthi, Vaishnavi Putnala, and Ashok Kumar. Smart college chatbot using ml and python. In *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, pages 1–5. IEEE, 2020.