Grenze International Journal of Engineering and Technology, June Issue

CampusMate: Reimagining College Guidance AI Chatbot for Streamlined University Services

Bhat Geetalaxmi Jairam[1], Nithish Kumar J[2], Olen Dale Pinto[3], Rachana R[4] and Rajat Giridhar Ghanate[5]
[1-5]Department of Information Science and Engineering, The National Institute of Engineering, Mysore, Karnataka, India
Email: bgj@nie.ac.in, 2021is_nithishkumarj_b@nie.ac.in, 2021is_olendalepinto_b@nie.ac.in, 2021israchanar_b@nie.ac.in, 2021is_rajatgiridharghanate_b@nie.ac.in

***Abstract*— Most of the students, faculty, and administrators face issues while accessing the latest and correct information about the university services like grades, course schedules, and faculty contacts. To resolve such issues, the AI-powered chatbot, CampusMate, aims to make inquiries about university-related matters easy to access and streamline and automate inquiries. By utilizing NLP and machine learning, CampusMate is able to understand the user queries and give them real-time answers, making it easier for the users. With integration with a relational database, CampusMate fetches and delivers information on academic performance, departmental details, and faculty contacts. The design of the system is scalable and extensible, which will allow it to adapt to evolving needs at universities and easily add new features. This research describes the architecture, core functionalities, and technical implementation of the chatbot and illustrates its potential in enhancing university services, improving accessibility, and reducing administrative burdens for students, faculty, and staff.**

***Index Terms*— AI-powered chatbot, Natural Language Processing (NLP), Machine Learning.**

I. Introduction

University environments are increasingly complex, with diverse stakeholders—students, faculty, and administrators—facing challenges in accessing timely, relevant information. Traditional methods like static websites and bulletin boards are inefficient, leading to frustration and delays. This creates a need for intelligent systems that can meet the demands of a fast-paced academic environment. CampusMate, an AI chatbot, would be the best solution for universities to centralize inquiries and service requests, providing an interface to automate all kinds of university-related queries and services. This integration of NLP with a strong backend system would make it easy for users to communicate conversationally, getting instant real-time support, thus saving the time and hassle of navigating complex portals or static content.

At the heart of CampusMate's functionality is its NLP engine, which uses techniques such as tokenization, stemming, and the bag-of-words approach to understand user queries.. CampusMate uses a pre-trained neural network developed with PyTorch, which is fine-tuned on datasets of academic-related queries. This way, CampusMate is always improving its accuracy and giving more relevant responses. It can also invoke certain actions, such as fetching data from a relational database or executing tasks like SGPA calculation based on predefined academic records.

CampusMate uses an SQLite3 database to store and retrieve important university data, such as course details, student records, faculty schedules, and department information.

Grenze ID: 01.GIJET.11.2.803_22

Accessing this structured database allows CampusMate to provide real-time, accurate information, such as checking course availability, viewing academic progress, or retrieving university policies.
It can perform various functions such as answering FAQs, service request management, and support in administration like course registration or exam scheduling. CampusMate has a personalized interface and changes its responses based on the user's role as either a student, faculty member, or administrator, thereby providing the appropriate, role-based support for each user.

II. Literature Work

While chatbots have shown great potential in education, challenges remain in ensuring accurate intent recognition, handling complex queries, and maintaining user privacy and data security.
Katherine Rosenbusch [1] first viewed that technology is revolutionizing higher education. Institutions can be an incubator to rethink and redesign education for the benefit of society altogether. Online, mobile, and blended learning is part of our future. An important step is to track how these models are actually enriching the learning outcomes.
M. Jang, Y. Jung, and S. Kim in [2] explores perceptions of managers. Most studies to date have taken a customer/developer-centric perspective, but adoption of chatbots is highly reliant on the willingness of managers in accepting their use. It also predicts the future of the use of chatbots in financial firms. The paper highlights South Korea's active integration of chatbots into the financial sector, which gives insights into its potential growth and role in digital transformation within the industry.
[3] This research paper is a study of the possibility of chatbots as natural language interfaces for question-answering (QA) systems. A system is proposed that converts website text into a chatbot knowledge base format. The knowledge base was used to answer questions based on it, automatically retrained using the TRE09 QA track. The results of the evaluation were promising with two-thirds of the generated answers correct.
M. Rajbabu, P. Prabhuraj, and S. Jeyabalan [4] details the design of a chatbot with the intent to process banking questions through vernacular languages. They focus on the aspect that chatbots are interactive tools of software meant to imitate human-like conversations, mostly being used in the customer support system, like a help desk, e-commerce, and business services. The authors suggest the integration of popular web services along with chatbot technology can potentially improve user experience because it is highly accessible and also secure for broader audiences, even for regional-language speakers.
Lebeuf, Storey, and Zagalsky [5] discuss the significant impact that software bots are having on the software engineering industry. Using a detailed case study, the authors demonstrate the potential of incorporating bots into the software development process, showing that they can perform simple and complex tasks. The research would focus on bot technology to enable the optimization of operations, to reduce human errors, and also to streamline practices in software engineering, thus encouraging innovation and ultimately improving overall functionality in development environments.
Architaba H Gohil [6] says that the use of AI Chatbots in education is becoming quite common with students receiving personalized 24/7 learning support. This study aims to explore the experiences of students when using Chatbots for educational purposes in Bhavnagar, and how useful these are, their difficulties, and the need for clarity in boosting the learning outcomes. However, teacher perceptions differed, with some focusing on the dynamic workability of AI for teaching support and others weighing specific concerns regarding the future of teaching with AI.
The authors, Moneerh Aleedy and Hadil Shaiba in [7] focused on developing an AI-powered chatbot for customer support using techniques such as NLP and deep learning, for example LSTM, GRU, CNN in creating automated responses to the queries for the customer. A form of sequence-to-sequence learning, where the queries are mapped into responses. The paper gives emphasis to the data preparation, model design, and evaluation metrics, such as BLEU and cosine similarity. The experiment results showed that both LSTM and GRU performed very well, but LSTM was selected as the best model for the generation of meaningful and emotional responses.
Intent Classification, also known as intent detection or intent recognition, is the ability to identify what a user might be trying to achieve through natural language input. The objective of IC is to map user queries to predefined intents, such as booking a flight, checking the weather, or seeking customer support. By accurately classifying the user's intent, the chatbot can provide relevant responses or trigger appropriate actions, enabling efficient and meaningful interactions in conversational AI systems.[8]
A major challenge in natural language processing (NLP) is achieving efficient intent classification, particularly in scenarios with limited data for each intent, commonly referred to as "few-shot" learning or low-data volume

scenarios. Traditional machine learning models often struggle in such environments due to the scarcity of labeled training data. To address this, transfer learning has become a popular approach. This involves utilizing pre-trained sentence encoders, such as BERT [9], RoBERTa [10], ALBERT [11], and XLNet [12], which have been trained on large corpora of text and can be fine-tuned on smaller datasets to improve intent classification performance in resource-constrained settings.
The below Table I, shows the comparison between various AI Chatbots. The table presents a comparative analysis of various literature surveys on AI chatbots, highlighting key findings, and challenges.

TABLE I. COMPARISON OF VARIOUS STUDY

Reference	Focus Area	Key Findings	Challenges/Limitations
Katherine Rosenbusch [1]	Technology in Higher Education	Education is transforming through online, mobile, and blended learning. Institutions need better tracking systems.	Ensuring technology actually leads to meaningful educational improvements.
M. Jang, Y. Jung, S. Kim [2]	Chatbot Adoption in Finance	Manager perceptions significantly impact chatbot adoption. South Korea is actively integrating finance chatbots.	Adoption depends on manager willingness, not just technology readiness.
Study on QA Chatbots [3]	Question-Answering Systems	Converts website text into chatbot knowledge base; 66% answer accuracy.	Needs improvement in accuracy and continuous learning capabilities.
M. Rajbabu et al. [4]	Vernacular Chatbots in Banking	Regional language banking chatbots improve accessibility and security.	Requires web service integration and improved language models.
Lebeuf, Storey, Zagalsky [5]	Software Bots in Development	Bots streamline tasks, reduce human errors, and enhance innovation.	Adoption challenges in complex software environments.
Architaba H Gohil [6]	AI Chatbots in Education	Provides 24/7 personalized learning support; student experiences vary.	Teacher concerns about AI's role and need for better outcome tracking.
Aleedy & Shaiba [7]	AI Chatbots in Customer Support	NLP with deep learning (LSTM, GRU) for automated responses, LSTM performed best.	Model selection depends on data quality; responses need improvement.
Intent Classification [8]	Intent Recognition in NLP	Maps queries to predefined intents to enhance chatbot efficiency.	Struggles with ambiguous inputs and requires robust training data.
BERT [9]	Bidirectional Transformers	Pre-trained models that process text bidirectionally for improved context understanding.	Computationally expensive and requires significant fine-tuning for specific tasks.
RoBERTa [10]	Optimized BERT Approach	Enhanced training methodology improves on BERT with better performance across NLP tasks.	Requires even more training data and computational resources than BERT.

III. METHODOLOGY

The following are the technologies and methodologies applied in chatbot development to guarantee effective performance and user satisfaction. Rule-based chatbots employ pre-scripted scripts and decision trees for basic interactions, whereas AI-driven chatbots utilize Natural Language Processing (NLP), Neural Networks and Machine Learning (ML) to dynamically adjust to user inputs.

A. Feedforward Neural Network (FNN)

A Feedforward Neural Network (FNN) , which uses a Multi-Layer Perceptron (MLP) type of artificial neural network which is used as a Classification Model where information moves in one direction-from the input layer through hidden layers to the output layer. Unlike recurrent neural networks (RNNs), FNNs do not have cycles or feedback loops.
Algorithm for Feedforward Neural Network
Step 1: Data Preprocessing
- Load dataset
- Normalize/scale features
- Split into training, validation, and test sets

Step 2: Initialize Network Parameters
- Define input_size, hidden_size, and num_classes

- Initialize weights and biases for each layer

Step 3: Forward Propagation

- Compute output of the first hidden layer:

$$H1 = ReLU\,(W1 * X + b1) \qquad (1)$$

- Compute output of the second hidden layer:

$$H2 = ReLU\,(W2 * H1 + b2) \qquad (2)$$

- Compute output of the third hidden layer:

$$H3 = ReLU\,(W3 * H2 + b3) \qquad (3)$$

- Compute final output (logits):

$$O = W4 * H3 + b4 \qquad (4)$$

Step 4: Compute Loss

- Use Cross-Entropy Loss for classification tasks:

$$Loss = -\sum(y * \log(\hat{y})) \qquad (5)$$

- Compute how far predictions are from the actual values.

Step 5: Backpropagation

- Compute gradients of the loss with respect to weights.
- Adjust weights using gradient descent or Adam optimizer.

Step 6: Model Training

- Repeat forward propagation -> loss computation -> backpropagation -> weight update for multiple epochs.

Step 7: Model Evaluation

- Use accuracy, precision, recall, F1-score to evaluate performance.
- Test on unseen data.

B. Rule Based NLP Model

A Rule-Based NLP Model is a non-machine learning approach that utilizes tokenization, stemming, and pattern matching to process text data. This model is particularly useful for tasks like course name recognition, where predefined rules can accurately extract specific keywords and phrases.

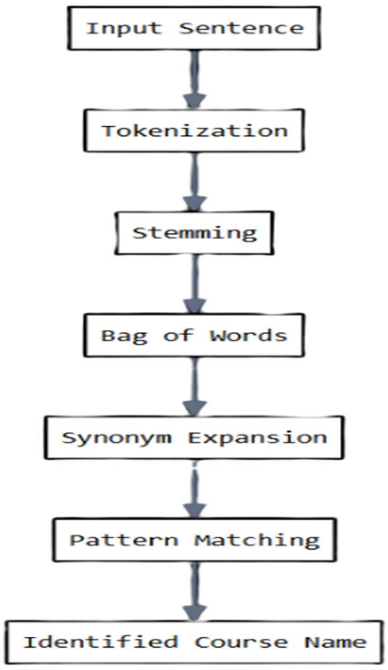

Fig. 1. Flowchart of Rule Based NLP Model

Figure 1 shows the flowchart of Course Name Extraction Natural Language Processing Pipeline. This flowchart illustrates the sequential text processing operations used to search for course names in input sentences

IV. SYSTEM ARCHITECTURE

Building the AI-powered university services chatbot, CampusMate, is based on the following major steps involving NLP, machine learning, and database management. The system will be developed to interpret the user queries and respond to them in real-time for students, faculty, and administrators. This is the methodology of building such a chatbot.

The Figure 2, illustrates the structure of a college information chatbot using AI. It supports Admin, Department, Teacher, and Student profiles with unique responsibilities. The front-end (HTML/CSS/JavaScript) is connected to a Flask/Python backend, which analyses questions via an NLP engine (NLTK/SpaCy) and a neural network (PyTorch). Storage of data is facilitated via an SQLite3 database for optimized retrieval.

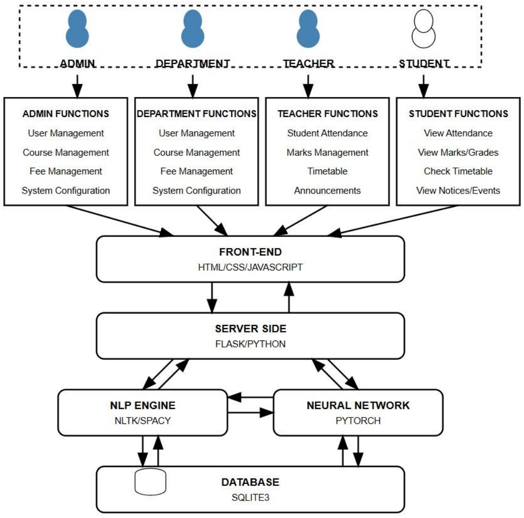

Fig. 2. System Architecture

A. Flow Diagram

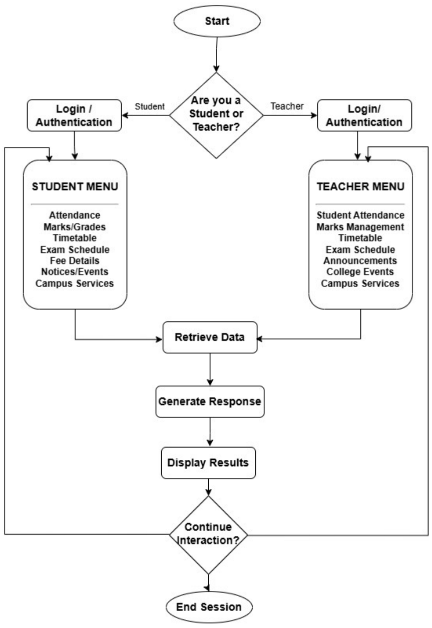

Fig. 3: Flow Diagram of the Chatbot

The flowchart shown in the Figure 3, depicts an integrated college management system that begins by identifying users to be students or teachers. After authentication, students are directed to a menu with attendance record, marks/grades, timetable, exam schedules, fee details, notices/events, and campus services, while teachers are directed to their page with student attendance tracking, marks management, timetable, exam scheduling, announcements, college events, and campus services. Both the user flows converge into a common processing stream where data is retrieved, responses are generated, and results are displayed. The system then prompts users to either continue their interaction by returning to their respective menus or to end their session, leading to a streamlined but role-based educational management experience that serves the specific requirements of both students and teachers while sharing a common operating platform.

B. Data Collection and Database Design

The data collection for CampusMate involves gathering structured information related to students, faculty, courses, and departments, which is stored in a relational database (SQLite3). The database is designed with multiple tables to efficiently organize this data. Key tables include Students (with fields like student_id, name, and department_id), Courses (course_id, course_name, credits), Faculty (faculty_id, name, email), and Departments (department_id, department_name, head_of_department). Additionally, the Enrollments and Class Schedules tables store information about student course enrollments and schedules. This well-organized database structure ensures fast and seamless retrieval of university-related information for user queries.

C. Entity Relationship Diagram

The ER diagram in Figure 4 is a university management system for the CampusMate Chatbot, showing entities and relationships. The most significant entities are ADMIN, STUDENT, FACULTY, DEPARTMENT, COURSE, CLASS_SCHEDULE, and ENROLLMENT. The ADMIN manages the system. A STUDENT is in a DEPARTMENT and is enrolled in numerous COURSES through the ENROLLMENT entity, which tracks attendance and grades. FACULTY members are given departments and can teach numerous COURSES, which are scheduled in the CLASS_SCHEDULE entity. COURSES are in a department and have attributes like course_code and credits. The CLASS_SCHEDULE shows relationships between courses, faculty, and departments and defines days_of_week, start_time, and end_time. This ER diagram accurately represents the student, faculty, course, and enrollment relationships of the university system.

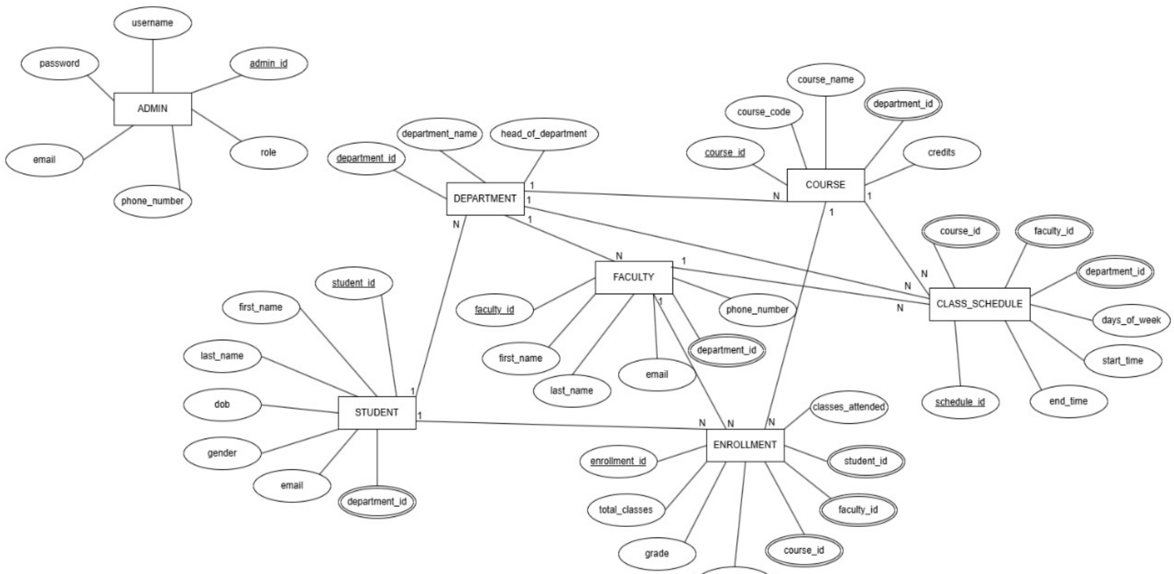

Fig. 4: ER Diagram a University Management System for Chatbot

D. Backend Development

It's built with the Flask, a lightweight web application framework, that listens for requests through the API. Users interact with CampusMate through RESTful API endpoints where the chatbot processes what the user typed, identifies the intent of the user, and performs the corresponding actions. It also interacts with SQLite3 through Object Relational Mapping (ORM) tool SQLAlchemy.

V. Results and Discussion

The system was able to dynamically execute database queries, thus ensuring fast and accurate responses. The Bag of Words model with synonym expansion via WordNet proved effective in understanding diverse user inputs, even when phrased differently. SQLite3 was integrated as the database, allowing for seamless storage and retrieval of structured university data.

A. Training Loss using Cross – Entropy Loss

For a classification problem with C classes, the Cross-Entropy Loss is given by:

$$L = -\sum_{i=1}^{C} y_i \, log(\,\hat{y}\mathrm{i}\,) \tag{6}$$

where:

- yi is the true label (one-hot encoded, meaning only one class has a value of 1, and the rest are 0).
- $\hat{y}$i is the predicted probability for class i (output from the model after applying softmax).
- The sum runs over all possible class

B. Accuracy of the ML Model

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{\text{Total Number of Predictions}} * 100 \tag{7}$$

The Accuracy of the Feedforward Neural Network Model is 98%.

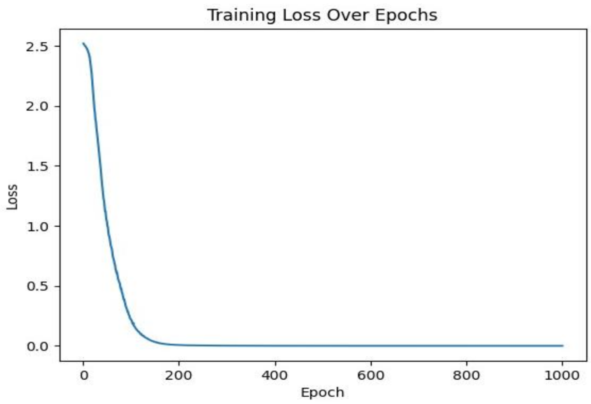

Fig. 5: Graph of Loss vs Epoch During Training Dataset (Intents)

This graph in Figure 5, illustrates the training loss of an ML model for 1000 epochs. The y-axis represents the loss, and the x-axis represents the number of epochs.

TABLE II. Testcases for CampusMate Chatbot

Test Case ID	Test Scenario	Test Steps	Expected Response	Actual Response	Status	Remarks
TC_01	Verify greeting response	User enters: 'Hi'	Bot should respond with a greeting message	Bot responded: 'Hello, thanks for visiting'	Pass	Working as expected
TC_02	Verify response for course details	User enters: 'Tell me about courses offered'	Bot should display all available courses with duration	Bot responded with complete course list	Pass	Working as expected
TC_03	Verify response for faculty details	User enters: 'Tell me about Professor Smith'	Bot should display Professor Smith details	Bot responded: 'Let me fetch the details for you.' (Static Response)	Fail	No actual details provided from database
TC_04	Verify response for invalid input	User enters: '@#$%$'	Bot should respond with: 'I am unable to understand that..'	Bot responded correctly	Pass	Working as expected
TC_05	Verify response for course query	User enters: 'Tell me more about BTECH'	Bot should provide complete information about BTECH course	Bot responded: 'We offer courses like BCA, B...' (Incomplete)	Fail	Incomplete response shown

The loss is initially high but drops quickly in the early epochs, indicating that the model is learning well. Once the model is about 200 epochs in, the loss stabilizes and nears zero, meaning that it is converging. This indicates that the model has been able to minimize the error and is well-trained with minimal overfitting.
The results of the CampusMate proposed system show that NLP and machine learning can be integrated to streamline university service delivery.
The chatbot is able to interpret and respond to a wide range of student, faculty, and administrative queries in real-time. Through thorough testing, the system was able to achieve high precision in the user intent identification - retrieving grades, course schedules, and faculty contact details-through combining pattern matching and intent classification, using a neural network.
The Table II, represents the test cases designed to verify the responses of the developed AI Chatbot. It checks whether the chatbot is providing accurate responses for different user queries based on the predefined intents. This table helps in identifying errors and improving the performance of the chatbot.

VI. Conclusion

CampusMate demonstrates how AI-powered chatbots can really simplify and automate the whole process of inquiry related to the university and serve as a better solution for students, faculty, and administration. The integration of NLP, machine learning, and relational database enables it to provide answers to queries such as grades, course information, and faculty in real time.
With scalable and extensible architecture, this tool is future-proof for academic institutions because it could adapt to any evolving needs within the university. While it's effective in completing a wide variety of tasks, future improvements may further increase accuracy and user experience by enhancing its contextual understanding.

Future Work

However, some complexities came up with ambiguous questions, where the chatbot was doing badly in contextual answers. Future work may focus on improving contextual understanding through advanced NLP models like BERT. CampusMate significantly reduces administrative workload and enhances user experience by automating routine queries.
In the future, CampusMate could then be elevated with the further incorporation of more advanced NLP models such as BERT or GPT to make it more contextual in understanding the chat or even have its ability to cope with complex, multi-turn conversations.
Incorporating voice-based interaction using Speech-to-Text APIs would also make it more accessible. Integrations with more Learning Management Systems would mean a greater scale of services to offer through CampusMate, like registration for courses and assignment tracking. Another related direction could be creating an analytics dashboard for administrators within the universities.

References

[1] Rosenbusch, Katherine. "Technology intervention: rethinking the role of education and faculty in the transformative digital environment." Advances in Developing Human Resources 22.1 (2020): 87-101
[2] M. Jang, Y. Jung, and S. Kim, “Investigating managers’ understanding of chatbots in the Korean financial industry,” Computers in Human Behavior, vol. 120, Jul. 2021, doi: 10.1016/j.chb.2021.106747
[3] Shawar, B.A. and Atwell, E., 2010. Chatbots: Can they serve as natural language interfaces to QA corpus?. In 6th International Conference Advances in Computer Science and Engineering (pp. 183-188).
[4] M. Rajbabu, P. Prabhuraj, and S. Jeyabalan, “An intelligent behaviour shown by chatbot system for banking in vernacular languages,” International Research Journal of Engineering and Technology (IRJET), pp. 1210–1212, 2019.”
[5] “C. Lebeuf, M.-A. Storey, and A. Zagalsky, “Software bots,” IEEE Software, vol. 35, no. 1, pp. 18–23, Jan. 2018, doi: 10.1109/MS.2017.4541027.”
[6] Gohil, A.H., 2025. Enhancing Student Learning With Chatbots: Evaluating Utility And Challenges Among Undergraduate Students In Bhavnagar.
[7] Aleedy, M., Shaiba, H. and Bezbradica, M., 2019. Generating and analyzing chatbot responses using natural language processing. International Journal of Advanced Computer Science and Applications, 10(9).
[8] Y.-T. Lin, A. Papangelis, S. Kim, S. Lee, D. Hazarika, M. Namazifar, D. Jin, Y. Liu, and D. Hakkani-Tur, “Selective in-context data augmentation for intent detection using pointwise v-information,” arXiv preprint arXiv:2302.05096, 2023.
[9] V. Khan and T. A. Meenai, “Pretrained natural language processing model for intent recognition (bert-ir),” Human-Centric Intelligent Systems, vol. 1, no. 3-4, pp. 66–74, 2021.

[10] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” arXiv preprint arXiv:1907.11692, 2019.
[11] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “Albert: A lite bert for self-supervised learning of language representations,” arXiv preprint arXiv:1909.11942, 2019.
[12] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” Advances in neural information processing systems, vol. 32, 2019.