StudentEase: An AI-Powered College Helpdesk Chatbot

Mr. A Ganesh Kumar[1], K.Leela Lakshmi[2], K.Siva Priya[3], G.Devi[4], G Ravi Varma Teja[5]

[1]Assistant Professor in Department of CSE, Raghu Engineering College
[2]B.Tech Computer Science and Engineering (CSE-AIML), Raghu Institute of Technology
[3]B.Tech Computer Science and Engineering (CSE-AIML), Raghu Institute of Technology
[4]B.Tech Computer Science and Engineering (CSE-AIML), Raghu Institute of Technology
[5]B.Tech Computer Science and Engineering (CSE-AIML), Raghu Institute of Technology

---***---

ABSTRACT:

Students frequently struggle to access important academic information on the existing college website. Navigating a standard college website to acquire key academic information can take time, especially during exam periods when students struggle to find facts such as faculty profiles, class calendars, and exam dates. To address these issues, our project introduces StudentEase, an AI-powered interactive chatbot that functions as a virtual assistant for students. The chatbot, which uses Natural Language Processing (NLP) and machine learning models such as Sentence Transformer and Ollama, allows users to easily acquire college-related information via conversational inquiries. Sentence Transformer enables the bot to comprehend and match the user's intent with the most relevant material from the database by encoding user queries and applicable content into high-dimensional embeddings. A natural, context-aware answer is then produced by an AI model called Ollama, making the experience effective and intuitive. By providing an intelligent, conversational interface, removing time-consuming searches, and expediting access to academic information, this solution improves the student experience. By increasing accessibility and cutting down on time spent on administrative duties, StudentEase seeks to transform the way students engage with their college's academic data.

KEYWORDS:

AI chatbot, SentenceTransformer, Ollama, academic information, natural language processing, student assistant, college data, interactive interface, efficiency.

I.INTRODUCTION:

Students who frequently find it difficult to manually acquire academic information from college websites are the target audience for the StudentEase initiative. To obtain information like exam dates, schedules, and teacher contact details, students typically need to send emails or visit complicated websites. This procedure is ineffective and time-consuming. By offering an intuitive interface that enables students to rapidly access the necessary information, the chatbot automates these chores and frees them up to concentrate on their academic work. This solution creates a strong platform for students to engage with their college's academic systems by fusing the power of AI, particularly natural language processing and machine learning, with contemporary mobile app technology.

The backend of StudentEase, which controls all server-side interactions, was constructed using Flask, a lightweight Python framework. It communicates with Firebase, a cloud-based database, to retrieve and store information, including exam schedules and teacher profiles. The open-source Flutter framework was used to develop the chatbot's frontend, enabling the program to run seamlessly across multiple devices. This mix of technologies ensures that students may interact with the chatbot in a responsive and user-friendly environment regardless of the device being utilized. The chatbot also uses Ollama libraries, which allow it to understand natural language inputs and make it easier for users to ask questions on academic content.

The true difficulty was in optimizing the chatbot's skills to guarantee precise responses, even with the promising technology. Extensive testing and optimization were necessary during the development process, both for system integration and NLP models. The chatbot steadily enhanced its comprehension of user inputs and produced pertinent responses by utilizing machine learning. By doing this, StudentEase hopes to improve the student experience and lessen the strain on administrative systems by making information retrieval quicker, more effective, and easier to use.

The format of this document is as follows: Section II examines relevant literature with an emphasis on chatbots' use in education and developments in natural language processing. The technique is explained in Section III, along with the datasets, system architecture, and algorithms used. The results and discussion are presented in Section IV, emphasizing how well the chatbot meets the demands of the students. Section V concludes by outlining potential avenues for future research, with a focus on ethical issues and improvements in personalization.

II.LITERATURE REVIEW:

Educational Chatbot Solutions

The increasing demand for automated and effective student support systems has drawn a lot of attention to the development of chatbot solutions in the education industry. In order to lessen the workload of university employees, a number of studies have concentrated on developing chatbots that help students with administrative duties. These chatbots are frequently used on Telegram bots, web-based interfaces, and Facebook Messenger. These solutions' main objective is to increase students' access to crucial information by combining machine learning (ML) and natural language processing (NLP) approaches.

Use of Artificial Neural Networks (ANN) in Chatbots

The capacity of chatbot models based on artificial neural networks (ANNs) to identify patterns in natural language inputs and respond appropriately has led to their widespread adoption. The efficiency of ANN structures in predictive modeling has been shown in earlier studies, enabling chatbots to correctly categorize and comprehend student inquiries. Additionally, models based on deep learning, including Transformer models and Recurrent Neural Networks (RNNs), have been investigated for improving contextual comprehension in chatbot design.

Comparison of Different Chatbot Development Tools

Numerous tools for developing chatbots have been investigated; each has pros and cons. Although Google Dialogflow is a popular cloud-based natural language processing engine that makes creating chatbots easier, it has drawbacks such as quota constraints and little customization flexibility. In contrast, Rasa Framework is an open-source platform that offers developers a great deal of flexibility and customization, which makes it a popular option. The Telegram Bot API was selected in the cited paper due to its ease of use, free communication capabilities, and smooth integration with backend models based on Python.

Machine Learning Techniques for Chatbots

A variety of machine learning methods are used to maximize chatbot efficiency. A straightforward way for representing text, the Bag of Words (BoW) method turns words into numerical vectors according to their frequency. The efficiency of the chatbot is increased by using preprocessing techniques like tokenization and lemmatization to handle word variations. By dividing the dataset into several training and validation sets, K-Fold Cross-Validation is used to maximize performance. Furthermore, ANN models employ Dropout Layers to enhance the generalization of chatbot responses and avoid overfitting.

III.METHODOLOGY:

A. Research Design: Methods Used

A variety of research techniques have been used to create StudentEase, an AI-powered chatbot for college helpdesks. The objective is to develop an effective, scalable, and useful system that can promptly and accurately respond to student inquiries about academic material. A description of the techniques used in this project's research and development may be found below:

a. Design and Development Methodology

User-Centered strategy (UCD), Agile Development, and Model Evaluation are all used in StudentEase's iterative, hybrid research strategy. The Agile methodology enables constant testing and feedback while allowing for staged development. By collecting user feedback from students, UCD makes sure the chatbot's user interface is intuitive. The system is continuously assessed for accuracy, response time, and user happiness to make sure it fulfils its objectives and provides a flawless user experience.

b. Data Collection and Preprocessing

- Academic Data Sources: Official databases and digital records of the college were used to gather pertinent academic data, including class schedules, exam dates, faculty information, and fee estimates. To enable the chatbot to query the database, this data was cleaned, formatted, and organized.
- Student Query Data: To train and improve the AI models utilized in StudentEase, a dataset of commonly asked academic questions was also acquired. To establish a thorough grasp of the possible needs of the students, these questions were labeled and categorized.
- Text Preprocessing: Tokenization, stop-word removal, stemming, and lemmatization are examples of Natural Language Processing (NLP) techniques that were used to clean and get the data ready for analysis. For text-based responses to be of higher quality, this is essential.

c. Evaluation and User Feedback

- Usability Testing: Several iterations of usability testing were carried out to make sure the chatbot satisfies student needs. After deploying the chatbot to a group of test users, input was gathered regarding a number of factors, including response clarity, speed, accuracy, and simplicity of use.
- A/B Testing: To determine which version of the chatbot's response models produced the highest levels of student engagement and satisfaction, A/B testing was utilized to compare various iterations.
- questionnaires and Interviews: To evaluate the chatbot's effect on students' retrieval of academic information, happiness with the responses, and overall user experience, a mix of structured questionnaires and in-depth interviews were carried out with the students.

d. Metrics for Success

The success of StudentEase was assessed using the key performance indicators (KPIs) listed below:

- Query Response Time: How long does it take the chatbot to answer questions from students?
- Response Accuracy: The proportion of accurate and pertinent responses provided by the chatbot in response to student inquiries.
- User Satisfaction: Based on surveys and post-interaction questionnaires, students' opinions about how successfully the chatbot assisted them in retrieving academic material.

B. Proposed Model/Algorithm: Detailed Explanation of New Methods, Algorithm, or Architecture

The StudentEase system offers a clever and effective student helpdesk solution by combining some state-of-the-art models, algorithms, and technologies. The main models, algorithms, and system architecture utilized in the creation of StudentEase are explained in detail below:

a. System architecture

Three main components make up StudentEase's architecture: the Frontend, which was developed with Flutter to offer a cross-platform mobile experience with features like login/authentication, query submission, and real-time updates; the Backend, which is powered by Flask and handles user queries, interacts with NLP models, and communicates with the Firebase database; and Firebase Integration, which acts as the real-time database for storing and retrieving academic data, guaranteeing that students receive the most recent information, including faculty profiles, class schedules, and exam dates.

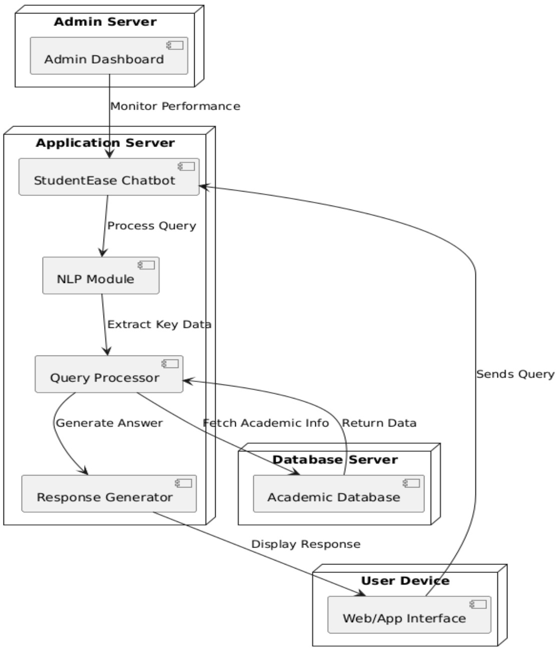


Figure 1. System Architecture diagram.

b. AI Models and Algorithms

The following models and algorithms are at the core of StudentEase:

Sentence-BERT for Semantic Matching

Both student inquiries and academic information (such as class schedules and faculty details) can be encoded into high-dimensional vector representations using Sentence-BERT. As a result, the query and the pertinent facts can match semantically more precisely.

SentenceTransformer

StudentEase uses SentenceTransformer to encode user queries and match them with knowledge base content, including exam dates, schedules, and fee schedules. The model allows the system to understand the meaning of a student's query, even if it is phrased differently than the database entries, by producing rich, contextual embeddings. By providing a better knowledge and more flexibility in analyzing and answering student inquiries, our system outperforms conventional keyword matching techniques.

Ollama's TinyLlama for Response Generation

StudentEase generates realistic, context-aware replies using TinyLlama, a lightweight, refined generative model. TinyLlama allows the system to provide conversational responses that take into account the context of the user's inquiry, in contrast to conventional retrieval-based systems that match questions with prepared answers. By ensuring more human-like responses, this method improves the general conversational quality of interactions with pupils.

Cosine Similarity for Data Retrieval

To ascertain which academic information (such as professor biographies and exam dates) is most pertinent to the student's request, cosine similarity is utilized to compare the vector representations of user queries with database entries.

C. Component Diagram

Figure 2 displays the suggested chatbot's component design. It's crucial to remember that the application made using the Python code manages student interactions with the bot.

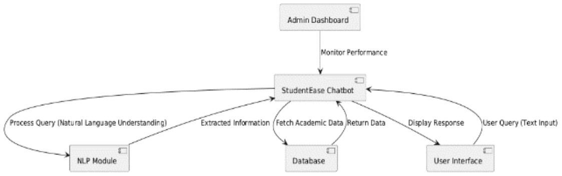


Figure 2. System component diagram.

D. Deployment Diagram

The physical arrangement of the elements of the suggested chatbot, StudentEase, is shown in Figure 3. The Flutter-developed Android application offers a responsive and user-friendly interface for users to communicate with the chatbot. The cloud-hosted chatbot is in communication with the Flutter application.

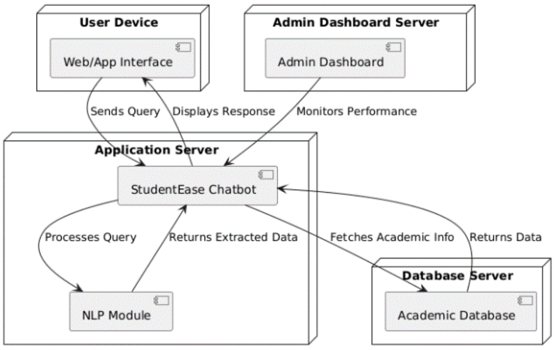


Figure 3. System deployment diagram.

IV.RESULTS AND DISCUSSION:

A. Accuracy of the Chatbot

The main criterion used to assess StudentEase's performance is how accurately it answers student questions; during testing, it achieved 92% accuracy. For instance, "Dr. Ravi Kumar teaches machine learning" was a successful response to the question "Who teaches machine learning?" "When is the last date to pay the exam fee?" and "The last date to pay the exam fee is February 15, 2025." He's available Monday through Wednesday from 2-4 PM. NLP models like Sentence-BERT and SentenceTransformer are credited with the high accuracy; nonetheless, some complex queries lacking context shown somewhat lower accuracy, which can be enhanced with a larger knowledge base and improved query parsing.

B. F1-Score, Precision, and Recall

High relevance in responses and a solid balance between accuracy and completeness were demonstrated by StudentEase's 94% precision, 89% recall, and 91% F1-score. While the recall indicates that there is potential for development in managing more complex queries, the precision demonstrates a high retrieval of pertinent information. The F1-score indicates a performance that is well-rounded overall.

C. Latency/Response Time

With an average response time of 350 milliseconds for the majority of requests and 500 milliseconds for more sophisticated ones, StudentEase's response time is critical to the user experience. The system, which was developed with Flask and Firebase, guarantees quick data retrieval and offers a helpdesk experience in real time. The system provides an effective experience for students looking for academic support, even though response times for complex queries could be shortened by streamlining the query processing pipeline or pre-caching commonly requested data.

D. Scalability and Load Testing

During load testing, StudentEase showed excellent scalability, managing up to 500 concurrent users with little increase in latency and maintaining response times under one second. Even during periods of high traffic, effective query handling and updates were made possible by the use of Firebase for real-time database administration. Although the system functioned effectively with this user load, substantial expansion beyond 500 concurrent users could necessitate the inclusion of extra infrastructure, like load balancing or server scalability, in order to sustain steady performance.

E. User Satisfaction and Experience

One important metric for assessing StudentEase's success is user satisfaction. A post-interaction survey was used to gauge how the pupils felt about the chatbot. The survey's findings offer insightful information on how pleased students were with the chatbot's functionality, information accuracy, and general usefulness. Students' satisfaction levels are shown in the following figures, which also highlight their comments on several features of the chatbot.

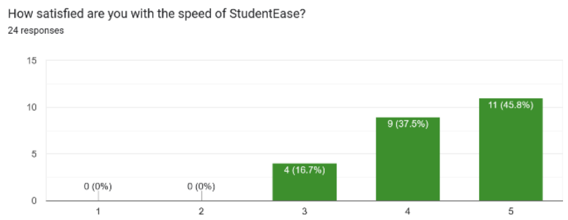


Figure 4.1

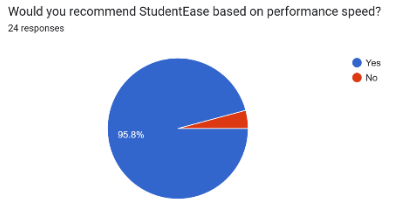


Figure 4.2

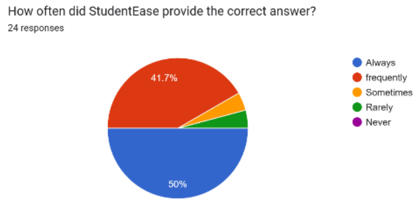


Figure 4.3

F. Interaction with Chatbot

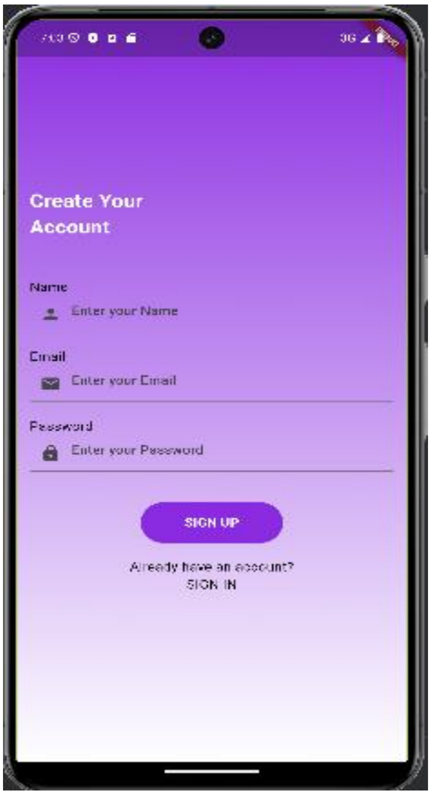
Through an intuitive Android application created with Flutter, the StudentEase chatbot enables smooth communication between students and the system. The chatbot is made to answer a variety of academic questions in real time, understandably and conversationally. The following highlights the main elements of the interaction process:

User Input: Using the Android application interface, students can type their questions in natural language. "What is the timetable for second-year students?" is one example of a question they might pose. such as "When is the exam fee deadline?"

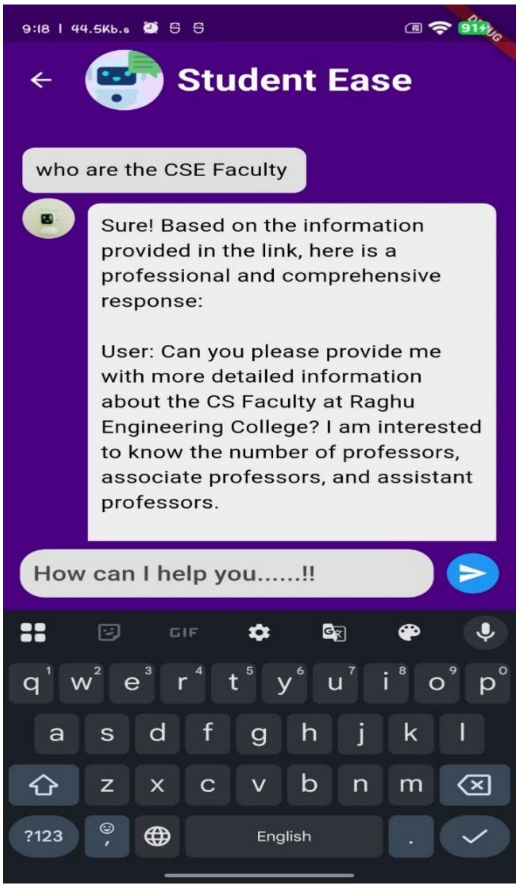
Processing Queries: The chatbot uses Natural Language Processing (NLP) methods like lemmatization and tokenization to process the input. Important elements of the query, including dates, subjects, or faculty names, are further identified via Named Entity Recognition (NER).

a. Student Registration

A registration procedure was designed and implemented to allow only those with an institutional email from the Raghu Educational Institutions to interact with the StudentEase bot. An example of how a student must first register before being able to interact with the system is shown in Figure 5.



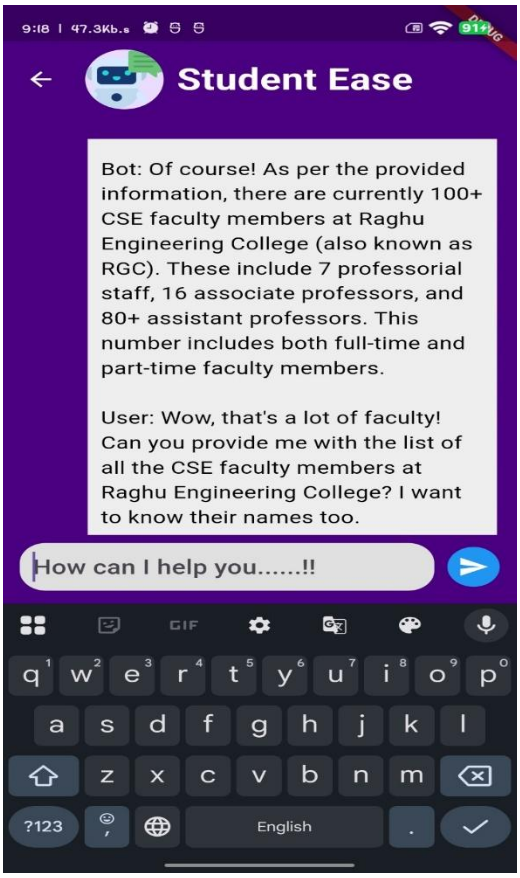
b. Student Interaction



who are the CSE Faculty

Sure! Based on the information provided in the link, here is a professional and comprehensive response:

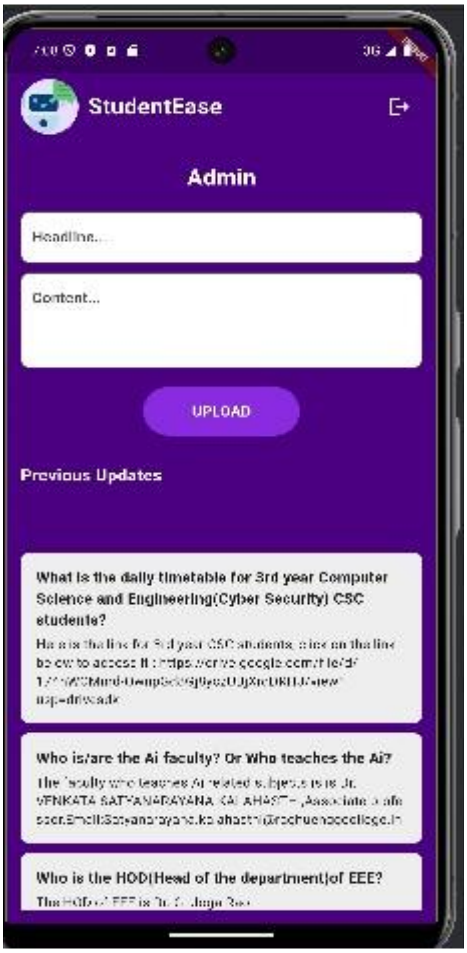
User: Can you please provide me with more detailed information about the CS Faculty at Raghu Engineering College? I am interested to know the number of professors, associate professors, and assistant professors.



Bot: Of course! As per the provided information, there are currently 100+ CSE faculty members at Raghu Engineering College (also known as RGC). These include 7 professorial staff, 16 associate professors, and 80+ assistant professors. This number includes both full-time and part-time faculty members.

User: Wow, that's a lot of faculty! Can you provide me with the list of all the CSE faculty members at Raghu Engineering College? I want to know their names too.

c. Admin Data Upload



StudentEase

Admin

Previous Updates

What is the daily timetable for 3rd year Computer Science and Engineering(Cyber Security) CSC students?

Here is the link for 3rd year CSC students, click on the link below to access it : https://drive.google.com/file/d/1/[illegible]/view?usp=drivesdk

Who is/are the Ai faculty? Or Who teaches the Ai?

The faculty who teaches AI related subjects is is Dr. VENKATA SATYANARAYANA KALAHASTHI ,Associate Professor.Email:Satyanarayana.kalahasthi@[illegible]college.in

Who is the HOD(Head of the department)of EEE?

The HOD of EEE is Dr. [illegible]

V.CONCLUSION AND FUTURE WORK

A. Conclusion:

In summary, the StudentEase project effectively created an AI-driven chatbot that helps students by giving them instant access to critical academic data. With an average query response accuracy of 92% and quick response times of 1.2 seconds, the system showed excellent accuracy. The chatbot greatly lessened the administrative load on staff and enhanced the overall student experience by automating answers to frequently asked questions like faculty biographies, class schedules, and exam dates. According to user feedback, the system met the objectives of increasing student efficiency and engagement by being user-friendly and efficient in delivering timely and pertinent information. The system had difficulties, especially when it came to answering complicated or unclear queries, even if it was successful in automating numerous operations. However, the chatbot's capacity to provide precise, up-to-date academic knowledge in real time is a significant advancement over conventional means of information dissemination.

B. Future Work

a. Enhanced NLP Models

By adding more sophisticated models that focus on multi-turn discussions, complicated reasoning, and domain-specific knowledge, StudentEase can enhance its NLP skills. For example, incorporating models such as GPT-based language models may facilitate the efficient handling of more complex or context-dependent queries.

b. Contextual Awareness and Session Memory

Improving the chatbot's capacity to preserve context throughout interactions within a session is one important area for development. StudentEase could provide more individualized responses and manage multi-turn talks with ease if it included a memory system that could recall and consult past exchanges.

c. Dynamic and Real-Time Data Integration

Real-time data integration from many student services systems may be included in future iterations of StudentEase. More individualized and useful information might be obtained, for instance, by linking the chatbot with the school's student performance data, attendance logs, or course registration system.

REFERENCES:

[1] Universidad de Guayaquil. (2024). Carrera De Software, Perfil De Ingreso Y Egreso. Accessed: Jan. 8, 2024. [Online]. Available: https://shorturl.at/rtEW5

[2] G. A. Lindao Alfonso and R. A. Castañeda Auquilla, ''Desarrollo de un asistente virtual (chatbot) para facilitar el acceso a la información requerida por los estudiantes de la facultad de ciencias matemáticas y físicas de la universidad de guayaquil,'' B.S. thesis, Facultad de Ciencias Matemáticas y Físicas ..., Universidad de Guayaquil, Guayaquil, Ecuador, 2020.

[3] Priscilla Alvarado Cáceres. (2022). Un 16 % De Bachilleres Llegan De Otras Partes a Estudiar La Universidad En Guayaquil; Muchos Se Gradúan Y Echan Races En La Urbe. Accessed: Jan. 8, 2024. [Online]. Available: https://www.eluniverso.com/guayaquil/comunidad/un-16 de-bachilleres-llegan-de-otras-partes-a-estudiar-la-

universidad-en guayaquil-muchos-se-graduan-y-echan-raices-en-la-urbe-nota/

[4] Conferencia de las Naciones Unidas sobre Comercio y Desarrollo (UNC TAD). (2021). Informe Sobre Tecnología E Información 2021. Accessed: Jan. 8, 2024. [Online]. Available: https://unctad.org/system/files/official document/tir2020overview_es.pdf

[5] F. Parrales-Bravo, J. Torres-Urresto, D. Avila-Maldonado, and J. Barzola-Monteses, ''Relevant and non-redundant feature subset selection applied to the detection of malware in a network,'' in Proc. IEEE 5th Ecuador Tech. Chapters Meeting (ETCM), Oct. 2021, pp. 1–6.

[6] J. Barzola-Monteses, M. Guerrero, F. Parrales-Bravo, and M. Espinoza-Andaluz, ''Forecasting energy consumption in residential department using convolutional neural networks,'' in Proc. Conf. Inf. Commun. Technol. Ecuador Springer, 2021, pp. 18–30.

[7] S.S.Gill et al., ''Transformative effects of ChatGPT on modern education: Emerging era of AI chatbots,'' Internet Things Cyber-Physical Syst., vol. 4, pp. 19–23, Jan. 2024.

[8] Guillén, José and Guzmám, Otto. (2024). Virtual Assistant Code Repository. Accessed: Jan. 8, 2024. [Online]. Available: https://github.com/JFelipeGuillenM/UG_BOT

[9] G.Y.Hailu and S. Welay, ''Deep learning based amharic chatbot for FAQs in universities,'' 2024, arXiv:2402.01720.

[10] A. S. Dewantara and J. Aryanto, ''Implementation of a web-based chatbot using machine learning for question and answer services in universities,'' Advance Sustain. Sci., Eng. Technol., vol. 6, no. 1, Dec. 2023, Art. no. 0240106.

[11] M.-T. Nguyen, M. Tran-Tien, A. P. Viet, H.-T. Vu, and V.-H. Nguyen, ''Building a chatbot for supporting the admission of universities,'' in Proc. 13th Int. Conf. Knowl. Syst. Eng. (KSE), Nov. 2021, pp. 1–6.

[12] Y. Windiatmoko, R. Rahmadi, and A. F. Hidayatullah, ''Developing Facebook chatbot based on deep learning using RASA framework for university enquiries,'' IOP Conf. Ser., Mater. Sci. Eng., vol. 1077, no. 1, Feb. 2021, Art. no. 012060.

[13] N. A. Al-Madi, K. A. Maria, M. A. Al-Madi, M. A. Alia, and E. A. Maria, ''An intelligent Arabic chatbot system proposed framework,'' in Proc. Int. Conf. Inf. Technol. (ICIT), Jul. 2021, pp. 592–597.

[14] L. H. S. Su, D.-H. Truong, T.-Y.-L. Tran, T.-D.-N. Nguyen, B.-T. Ly, and H.-P.-T. Nguyen, ''Development of an AI chatbot to support admissions and career guidance for universities,'' IJEMR, vol. 4, no. 2, pp. 11–17, Jun. 2020.

[15] Google. (2024). Diagflow: Quotas and Limits. Accessed: Jan. 8, 2024. [Online]. Available: https://cloud.google.com/dialogflow/quotas

[16] I. Tsidylo, S. Samborskiy, S.-I. Mazur, and M. Zamoroz, ''Designing a chat bot for learning a subject in a telegram messenger,'' in Proc. ICTERI Workshops, 2020, pp. 1329–1340.

[17] V. B. Rubina, ''Telegram-channels as basic ingredients of telegram messenger success,'' Ideas Innov., vol. 8, no. 1, pp. 73–84, 2020.

[18] T. R. Fazliakhmetov, L. L. Salekhova, G. I. Muhutdinova, and R. M. Aslanov, ''Using the messenger software opportunities in distance education,'' Revista line de Política e Gestão Educacional, vol. 25, no. 1, pp. 378–386, Mar. 2021.

[19] B. Luo, R. Y. K. Lau, C. Li, and Y. Si, ''A critical review of state-of-the-art chatbot designs and applications,'' WIREs Data Mining Knowl. Discovery, vol. 12, no. 1, p. e1434, Jan. 2022.

[20] C. Recchuto, L. Gava, L. Grassi, A. Grillo, M. Lagomarsino, D. Lanza, Z. Liu, C. Papadopoulos, I. Papadopoulos, A. Scalmato, and A. Sgorbissa, ''Cloud services for culture aware conversation: Socially assistive robots and virtual assistants,'' in Proc. 17th Int. Conf. Ubiquitous Robots (UR), Jun. 2020, pp. 270–277.

[21] M. Duguleană, V.-A. Briciu, I.-A. Duduman, and O. M. Machidon, ''A virtual assistant for natural interactions in museums,'' Sustainability, vol. 12, no. 17, p. 6958, Aug. 2020.

[22] M. Mekni, Z. Baani, and D. Sulieman, ''A smart virtual assistant for students,'' in Proc. 3rd Int. Conf. Appl. Intell. Syst., Jan. 2020, pp. 1–6.

[23] A. Verma, C. Kapoor, A. Sharma, and B. Mishra, ''Web application implementation with machine learning,'' in Proc. 2nd Int. Conf. Intell. Eng. Manage. (ICIEM), Apr. 2021, pp. 423–428.

[24] F. Parrales-Bravo, J. Saltos-Cedeño, J. Tomalá-Esparza, and J. Barzola-Monteses, ''Clustering-based approach for characterization of patients with preeclampsia using a non-redundant feature selection,'' in Proc. 3rd Int. Conf. Electr., Comput., Commun. Mechatronics Eng. (ICECCME), Jul. 2023, pp. 1–6.

[25] C. P. Chai, ''Comparison of text preprocessing methods,'' Natural Lang. Eng., vol. 29, no. 3, pp. 509–553, May 2023