

A HYBRID AI-BASED UNIVERSITY STUDENTS QUERIES CHATBOT USING NLP AND SBERT TECHNOLOGIES

Syed Muhammad Hassan Zaidi^{1*}, Saeed Ahmed², Imtiaz Husain³, Basit Hasan Qureshi⁴,
Sulaman Ahmad Naz⁵, Abdul Razaque⁶

^{1,3}Department of AI and Mathematical Sciences, Sindh Madressatul Islam University, Karachi

^{2,5,6}Department of Computing, Hamdard University, Karachi

⁴Department of Software Engineering, Sindh Madressatul Islam University, Karachi

¹m.hassan@smiu.edu.pk, ²saeed.ahmed@hamdard.edu.pk, ³imtiaz@smiu.edu.pk, ⁴basithq@smiu.edu.pk,

⁵sulaman.ahmad@hamdard.edu.pk, ⁶abdul.razaque@hamdard.edu.pk

DOI: <https://doi.org/>

Keywords

Chatbot, Natural Language Processing (NLP), Semantic BERT (SBERT), TF-IDF

Article History

Received on 27 August 2025

Accepted on 20 September 2025

Published on 10 October 2025

Copyright @Author

Corresponding Author: *

Syed Muhammad Hassan
Zaidi*

Abstract

Universities often struggle to efficiently handle student queries related to admissions, academics, and administration through traditional methods like emails, calls, or help desks. This research introduces an AI-powered chatbot designed to provide instant and accurate responses to university students using Natural Language Processing (NLP) techniques. The system combines Semantic BERT (SBERT), TF-IDF, and Cosine Similarity to enhance contextual understanding. A custom dataset of over 500 student queries was compiled through online and in-person surveys, with answers verified by university departments. The backend, developed in Python, processes queries and retrieves relevant responses, while the Streamlit-based frontend allows real-time interaction. The chatbot was deployed in a local environment and tested through unit, integration, and system-level testing. It demonstrated high accuracy, speed, and user satisfaction. Comparative results show SBERT outperformed the T5 model in terms of relevance and understanding. To efficiently handle students' queries, chatbots are a suitable consideration for future improvements to support multi-turn dialogue, multiple languages, real-time data updates, and cloud deployment for wider access. .

INTRODUCTION

In this paper we propose an AI-based university chatbot to address the growing needs for efficient and user-friendly information for university students. The inspiration came from the observation that students face in routine searching for the relevant as well as the correct responses.

Several researchers [1], [2], [3] addressed this issue to overcome the challenges like academic enrollment, administrative issues, IT concerns, co-curricular activities, student societies, and general problems.

We developed an AI-based interactive chatbot that covers all inquiries in one place to help university students with a wide range of inquiries, which will act as a virtual assistant and improve the effectiveness and availability of student services. Online Google Forms and physical surveys were used to collect data for analysis. First, we organized the queries with utmost care, and answers were obtained directly from the relevant university departments to ensure reliability.

We evaluated multiple models for accuracy, and Semantic BERT was selected as the primary model for the chatbot due to its high accuracy in understanding and retrieving relevant information. The hybrid Natural Language Processing (NLP) approach that combines Semantic BERT (Bidirectional Encoder Representations from Transformers) with TF-IDF (Term Frequency-Inverse Document Frequency) and Cosine Similarity techniques [4]. Semantic BERT uses its deep understanding of the contextual meaning behind user queries, while TF-IDF and Cosine Similarity increase the system's power to retrieve the most relevant answers from the dataset [5], [6], [7]. Further, the Text-To-Text Transfer Transformer (T5) model was used to explore possible answers during day-to-day queries. The T5 model performed less during the training phase, which is why it was not considered for the final solution and given preference to the generative approach. We aim to provide better user experience and reduce the university's load. AI-driven solution provides a student-friendly support system.

Literature Review

Early chatbots were relying on rule-based techniques such as Artificial Intelligence Markup Language (AIML) with limitations in contextual understanding and flexibility, whereas modern Artificial Intelligence (AI) based chatbots provide real-time, automated responses, which have been increasingly adopted in the educational sector for automating student support services. These chatbots significantly reduced the workload of university staff and further enhanced accessibility. Keeping these limitations in mind, present chatbots have incorporated Natural Language Processing (NLP) and Deep Learning (DL) models. The Transformer-based models, such as BERT, SBERT,

and T5, have significantly improved semantic comprehension and response relevance. Studies also suggest that hybrid approaches combining retrieval-based and generative techniques enhance chatbot performance in handling complex educational queries. In this regard, Amato [8] demonstrated the effectiveness of SBERT in a legal domain chatbot, highlighting its strong contextual understanding across domains. Gyeongmo Min and Junehee Yoo [9] applied BERT in science education, achieving good results but lacking multilingual adaptability. Anbiya [10] utilized a TF-IDF and Llama RAG hybrid for higher education helpdesks; however, scalability and contextual depth remained concerns.

Bistarelli and Cuccarini [11] emphasized closed-domain Q&A using BERT to prevent generative hallucinations but did not customize it for university-specific needs. Werkman [7] evaluated LLaMA 2 for institutional bots, noting underperformance due to domain mismatch. Peyton and Unnikrishnan [4] found SBERT superior to traditional chatbot frameworks in educational FAQs. Bilquise [12], developed a bilingual academic advisor chatbot, facing challenges in Arabic response accuracy, underscoring the need for stronger multilingual models. Chan [13] created a rule-based advisement bot, functional yet limited by its inability to handle vague or complex queries. Ranoliya [14] used Latent Semantic Analysis (LSA) in a university chatbot, but its lack of contextual strength and scalability limits modern applicability.

In addition, we illustrated the progression from static, rule-based systems to dynamic, transformer-driven architectures. The key challenges were multi-turn dialogue management, real-time database integration, and multilingual support. This underscores the evolution of research and highlights the growing need for domain-specific, intelligent,

and scalable solutions to ensure effective student support in higher education.

Table 1: Summarized Version of Literature Review

References	Year	Focus Area	Key Contribution	Research Gap
[8]	2023	Legal Domain Chatbots	Developed a legal chatbot using SBERT for legal assistance and conflict resolution.	Lacks applicability in other domains beyond the legal sector
[9]	2024	AI Chatbots in Science Education	Created a Sentence-BERT-based chatbot for high school science education in Korea.	Limited to the Korean language and lacks multilingual adaptation
[10]	2024	AI Chatbots in Higher Education	A hybrid chatbot using TF-IDF, Llama RAG to improve helpdesk efficiency.	Chatbot integration in university services is missing
[11]	2024	BERT-based Question Answering	Designed a closed-domain Q&A system using BERT, without generative model hallucinations.	Lacks fine-tuning for real-time university applications
[12]	2024	LLAMA-based University Chatbot	Explored the LLAMA 2-based chatbot for answering institute-specific queries.	Found accuracy and completeness issues, requiring further dataset expansion
[13]	2023	SBERT for Student FAQs	Compared SBERT-based chatbots with other chatbot frameworks for handling student FAQs.	Lacks testing on large-scale student queries

System Design and Methodology

This section presents a detailed account of the concept, development, and implementation of the university chatbot system. The primary objective was to design an intelligent and interactive tool capable of accurately addressing a wide range of student queries through advanced Natural Language Processing (NLP) techniques. To achieve this goal,

the development process was structured into four key phases: data collection, model architecture design, user interface development, and logical flow integration.

A. Dataset Collection and Preparation

High-quality data forms the foundation of accuracy and reliability in any research endeavor. For this study, a total of 500 queries were collected directly

from students using a combination of methods [15], [16], [17]: digital surveys administered via Google Forms and Microsoft Forms, along with in-person interactions conducted during academic and administrative sessions.

The collected queries reflected the real-world concerns of university students, covering diverse domains such as admissions procedures, course registration, fee structures, examination schedules, IT-related issues, and campus facilities. To maintain the integrity and authenticity of the dataset, several preprocessing steps were performed [18], [19], [20], [21]:

Careful review and filtering to remove redundancy.

Standardization to eliminate ambiguity

Table.2 Dataset

inconsistencies in phrasing. Verification of responses by the relevant university departments, ensuring that each question was paired with an accurate and authoritative answer.

Through this rigorous process, a clean, representative, and trustworthy corpus was constructed [22], [23], providing a reliable basis for both training and evaluation of the chatbot system.

These question-answer pairs were then organized in a structured tabular format (CSV), making them accessible for training machine learning models and retrieving responses.

CATEGORY	QUESTIONS	ANSWERS
Academic	Where can you find the academic calendar on Sindh Madressatul Islam University (SMIU) website?	Under Student section. Website link is https://www.smiu.edu.pk/
Academic	How to drop/add a course?	You can add or drop a course from your CMS portal before the deadline, after deadline you're not able to add or drop the course.
Academic	Is there need-based scholarship?	Yes, need-based scholarship is available for the needy students, each semester you can apply for this scholarship when the registrations are open for need-based scholarship.
Academic	How can you get the course outline?	You can get the course outline from HEC website or from Sindh Madressatul Islam University (SMIU) website https://www.smiu.edu.pk/ , for course material you have to contact your relevant teacher.
Academic	Why study?	Because there are a lot of students, so elevator can't fulfill their requirements, that's why the elevator is only available to Faculty and staff.
Academic	Why don't air conditioners work in winter season?	Air conditioners are off in winter season, if they are not working in summer season you can report a complaint in your department and it will be fixed.
Academic	Air condition?	Air conditioners are off in winter season, if they are not working in summer season you can report a complaint in your department and it will be fixed.
Academic	How to report a complaint to IT department?	You can report a complaint to IT department and it will be fixed.
Academic	Easy access?	You can get all the information regarding courses on your CMS.

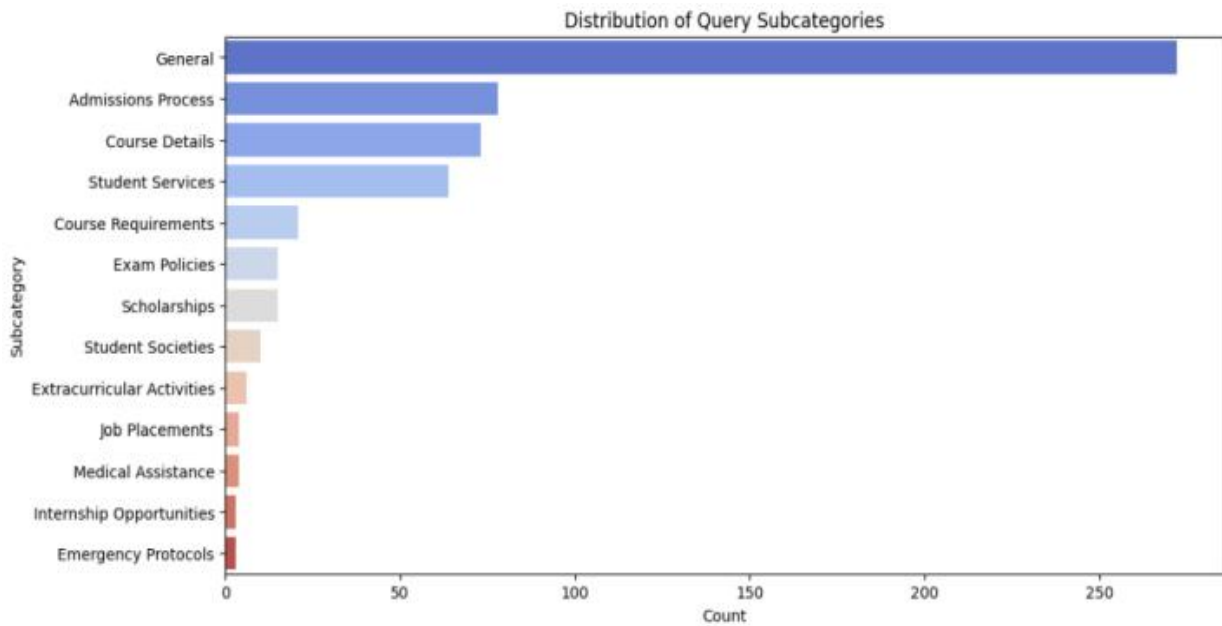


Figure 1: Distribution of Query Subcategories

Figure 1 represents the frequency of various types of student inquiries collected during the study. Each bar represents a specific category of queries, such as admissions, course registration, examination schedules, fee structures, and general campus information. The height of each bar corresponds to

the number of queries received in that category, providing a visual representation of the most common areas where students seek assistance. This distribution highlights the predominant concerns among students, enabling targeted improvements in university support services.

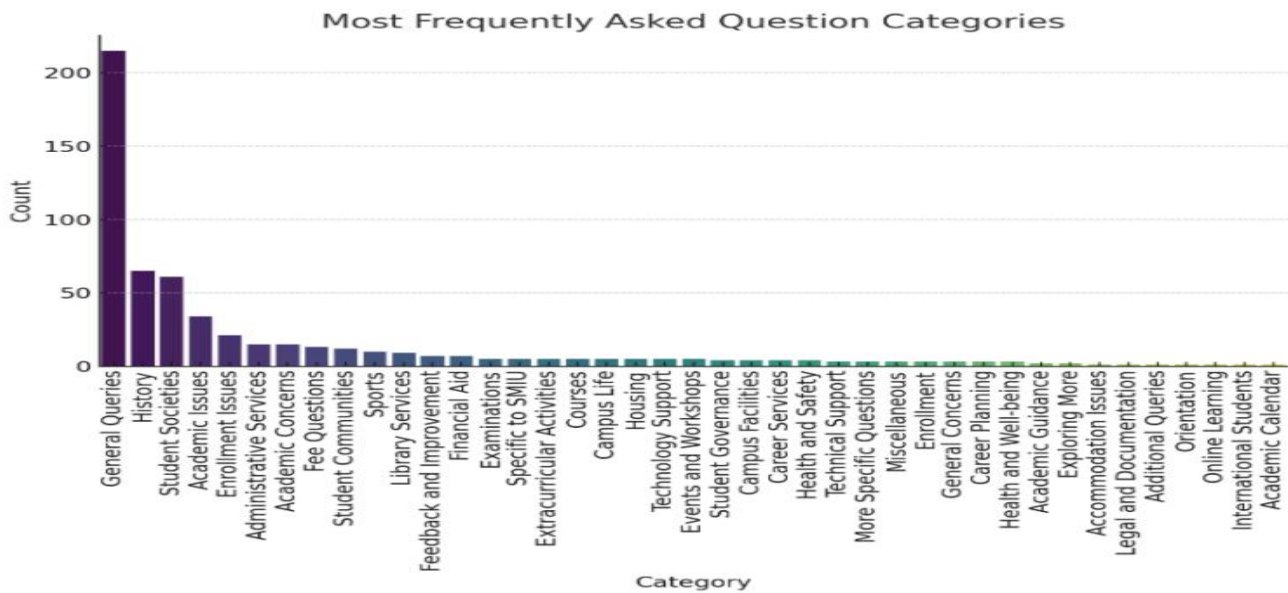


Figure.2 Most Frequently Asked Question Categories

Figure 2 displays the most common categories of student questions. Categories include topics such as admissions, course registration, fee payments, and examination details. The chart helps identify which areas generate the most inquiries, highlighting the specific informational needs of students and allowing universities to optimize their support services accordingly.

B. Model Architecture

The intelligence of the chatbot is driven by a hybrid Natural Language Processing (NLP) model that integrates multiple state-of-the-art techniques for query understanding and response matching. The architecture consists of three major components:

Semantic BERT (SBERT):

Semantic BERT (SBERT) is a transformer-based model capturing semantic meaning and contextual nuances. Student's query Q and i^{th} question in the dataset D_i , SBERT generates their respective semantic vectors such as:

$$V_Q = \text{SBERT}(Q)$$

$$V_{D_i} = \text{SBERT}(D_i)$$

Where V_Q and V_{D_i} are high-dimensional vectors that capture the contextual meaning of the queries.

Unlike conventional keyword-matching approaches, SBERT allows the system to recognize intent and conceptual similarity, even when queries are expressed in diverse linguistic forms.

TF-IDF (Term Frequency-Inverse Document Frequency):

This statistical method is employed to identify the most informative terms within a query. TF-IDF enhances the system's ability to highlight domain-specific vocabulary while reducing the influence of generic or frequently occurring terms.

$$\text{TF}(t, d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

Whereas Inverse Document Frequency (IDF) is used to measure how unique or rare a term is across all documents through the following formula.

$$\text{IDF}(t, \mathbf{D}) = \log \left(\frac{\text{Total number of documents in dataset } \mathbf{D}}{\text{Number of documents } d \text{ containing term } t} \right)$$

Then the TF-IDF Score was calculated by the following formula.

$$\text{TF-IDF}(t, d, \mathbf{D}) = \text{TF}(t, d) \times \text{IDF}(t, \mathbf{D})$$

In this process a TF-IDF vector created to represent both the input query and all stored dataset questions.

Cosine Similarity:

Cosine Similarity is the final measure used to quantify the closeness of the new query to existing queries in the knowledge base. It is calculated between the semantic vector of the student's query V_Q and the vectors of the dataset questions V_{D_i} , disclosing how much they align.

It measures the two embeddings to disclose the closeness of the new query aligns with existing queries in the knowledge base, ensuring that the most semantically relevant match is retrieved. In combination semantic understanding (SBERT), lexical weighting (TF-IDF), and vector similarity (Cosine Similarity),

$$\text{Cosine Similarity}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

Where $\mathbf{A} \cdot \mathbf{B}$ is the dot product of the vectors and $\|\mathbf{A}\|$ and $\|\mathbf{B}\|$ are the magnitudes (Euclidean norms) of the vectors.

In the chatbot's operation, the system calculates:

$$\text{Similarity}(Q, D_i) = \text{Cosine Similarity}(V_Q, V_{D_i})$$

The question D_{match} with the highest similarity score is identified, and its corresponding verified answer is retrieved and returned to the student. A score closer to 1 indicates a stronger, more relevant match. This hybrid approach, combining semantic understanding (SBERT), lexical weighting (TF-IDF), and vector similarity (Cosine Similarity), is what enables the chatbot to handle complex and contextually nuanced student queries with high accuracy. Furthermore, this hybrid model achieves robust query comprehension and precise response generation.

C. Frontend and Backend Development

To ensure accessibility and ease of use, the chatbot was developed with a modular architecture comprising a streamlined frontend and a robust backend.

Frontend Development:

Streamlit was implemented as a frontend using a Python-based framework for creating interactive web applications with minimal effort. Since it provides a clean, user-friendly interface where students can interact with the chatbot through a simple text input box, Queries are submitted directly through this interface, and responses are displayed instantly, ensuring a smooth user experience.

Backend Development:

The backend, also developed in Python, functions as the computational engine of the chatbot. Its responsibilities include:

- Loading and managing the question-answer dataset.
- Converting the student's query into an SBERT embedding.
- Applying TF-IDF vectorization for keyword weighting.

- Computing cosine similarity to identify the closest matching query.
- Retrieving and returning the most relevant answer to the frontend.

This decoupled design allows for flexibility and scalability. Enhancements such as upgrading the NLP model, integrating with larger databases, or redesigning the user interface can be incorporated without disrupting the overall system.

D. Logical Flow of the Chatbot

The operational logic of the chatbot follows a structured pipeline to ensure accurate and timely responses:

1. The student enters a query into the web interface.
2. The backend receives the query and preprocesses it by removing unnecessary characters and formatting it properly.
3. The SBERT model encodes the question into a semantic vector.
4. TF-IDF representation is computed for both the input query and all stored dataset questions.
5. Cosine similarity is calculated between the query vector and dataset vectors to find the closest match.
6. The most similar stored question is identified, and its associated answer is retrieved.
7. The response is sent back to the user via the Streamlit interface.

This logical sequence is executed in real time and optimized for speed, accuracy, and user satisfaction. The chatbot not only reduces the pressure on university staff but also ensures that students receive quick, consistent, and contextually relevant answers.

Moreover, the system was developed with flexibility in mind. The modularity of the design enables easy adaptation to other institutions or languages. Future improvements include enabling the chatbot to

handle follow-up questions in a multi-turn dialogue, integrating voice-based input and output, and connecting the chatbot to live university databases for dynamic, real-time updates.

Data Flow Diagram

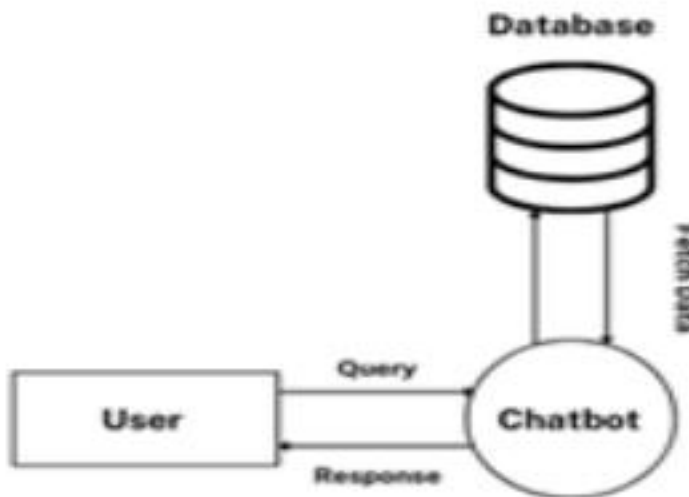


Figure 3(a): Data Flow Diagram (Level 0: Context Diagram)

Figure 3(a) provides a high-level overview of the chatbot system. It shows the basic flow of data between the user (student), the chatbot interface, the processing module, and the knowledge base.

ailed breakdown of the system’s internal components. It illustrates individual processes, including query input, semantic embedding via SBERT, TF-IDF vector generation, similarity scoring, and response retrieval.

Data Flow Diagram

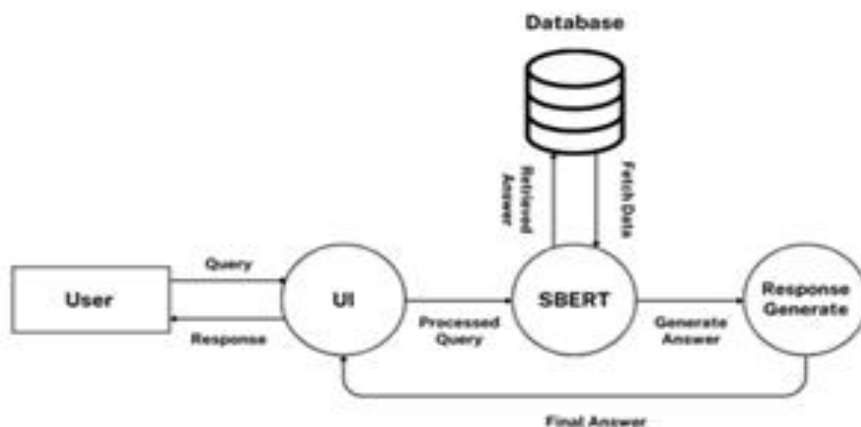


Figure 3(b): Data Flow Diagram (Level 1: Detailed Diagram)

design, the chatbot achieves its intended goal of delivering intelligent,

responsive, and scalable support to university students.

Implementation

This section explains the practical steps taken to turn the design of the chatbot into a fully working system. It describes the tools that were used, how all parts of the chatbot were connected, how the interface was built for students to use easily, and finally, how the chatbot was tested in a real-world setting.

A. Development Tools and Environment

The development of the chatbot was done using the Python programming language. Python was chosen because it is easy to use and has many useful libraries for artificial intelligence and machine learning. Some of the main tools and libraries used include:

- Streamlit: Used to build the user interface, allowing students to interact with the chatbot on a web page.
- Pandas: Used to read and manage the dataset, which contains the student questions and their verified answers.
- Scikit-learn: Used to perform the TF-IDF transformation and cosine similarity matching.
- Sentence-transformers: A library used to load the SBERT model, which helps the chatbot understand the meaning of student queries.

The chatbot was first tested in a local environment, meaning it was run on a personal computer. This helped in checking that everything worked correctly before thinking about deploying it on a cloud server for public access.

B. Integration of Components

After choosing the tools, the next step was to make sure all parts of the chatbot worked well together. The integration process involved the following steps:

- The chatbot loads the dataset file (CSV) into memory so it can access all stored questions and answers.
- The SBERT model is loaded to convert new student questions into a special format called vectors.
- At the same time, the system creates a TF-IDF matrix from all the existing questions in the dataset.

- When a student types a new question, the chatbot compares it to all stored questions using cosine similarity.
- The chatbot picks the question that is most similar and returns its answer to the student.

This integration ensures that students always get the most relevant and accurate answer based on what has already been verified and stored.

C. User Interface Features

The chatbot's interface was designed with a focus on simplicity, clarity, and ease of use, ensuring that students can interact with it effortlessly. Its core features include:

Title and Introduction clearly displayed at the top to explain the chatbot's purpose and guide first-time users. The Text Input Box is a field where students can type their questions. Whereas the send button allows users to submit their queries instantly.

The chat window displays both the student's question and the chatbot's response in a conversational format. In addition to these core functions, the interface also provides enhanced usability features, such as suggested questions. The predefined common queries that students can click on for quick access to answers.

The clear chat button enables users to reset the conversation and start fresh at any time. This user-friendly design ensures that students can access accurate information quickly, intuitively, and at any time, thereby enhancing the overall academic support experience.

D. Sample Use Case

To test the system, a simple use case was tried. A student asked the question: "How can I get my transcript?" The chatbot processed the query and searched through the dataset. It found a similar question and returned the following answer: "You can request a transcript by filling out the transcript form available on the student portal."

This example shows that the chatbot can understand what a student is asking, even if the question is not worded exactly like the one in the dataset. It provides correct, specific, and helpful answers. This proves that the chatbot is ready for use in a real

university environment and can be further improved over time.

Results and Evaluation

This section explains how well the chatbot worked after it was built and tested. The main goal was to find out if the chatbot could correctly understand questions, give useful answers, and be easy for students to use.

A. Testing Methods

The chatbot was tested in three main ways:

1. Unit Testing - Each part of the code was tested separately to make sure it worked as expected. For example, checking if the chatbot could correctly load the dataset.
2. Integration Testing - This test was done to check if different parts of the system worked together smoothly. For instance, checking if the SBERT

model, TF-IDF, and Streamlit interface worked in one flow.

System Testing - The complete system was tested like a real user would use it. Testers typed in questions to see if the chatbot gave the right answers.

B. Performance Results

The chatbot performed well during the tests. It was able to:

- Understand student questions even when asked in different ways.
- Return correct answers quickly.
- Handle different types of queries related to university life.

Most responses were accurate and helpful. It also helped reduce the need for students to visit help desks or send emails for simple questions.

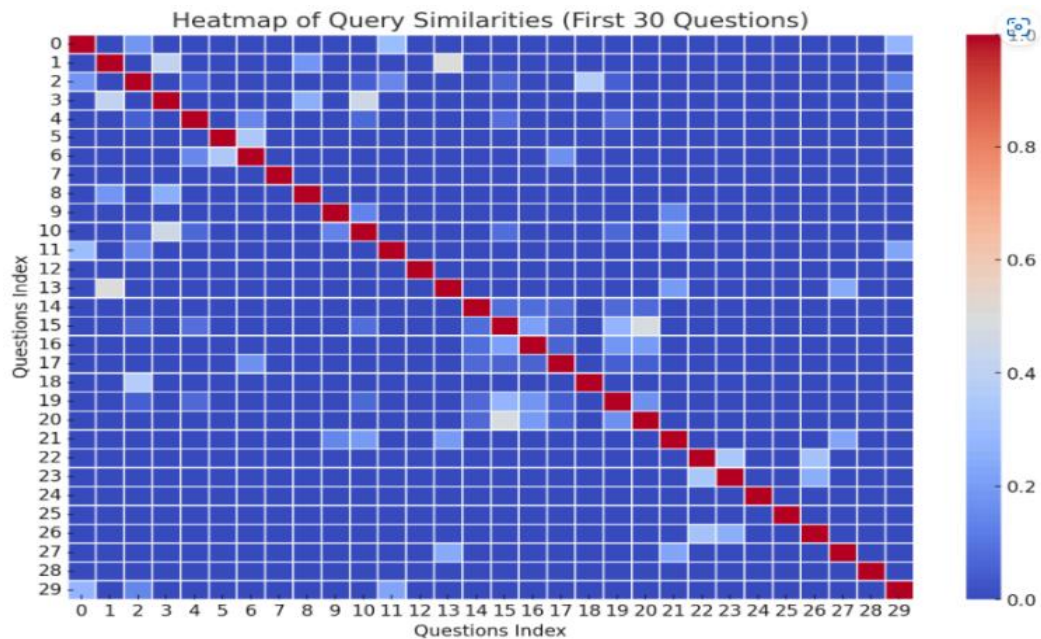


Figure.4: Heatmap of Query Similarities

The heatmap visually represents the similarity scores between different student queries. Darker regions indicate higher similarity values, suggesting overlapping or related questions. This visualization is useful for identifying clusters of queries that address similar concerns, which can be leveraged to streamline FAQ content and reduce redundancy in the dataset.

Figure 5 illustrates how cosine similarity scores are distributed when comparing new queries with existing ones in the dataset. A higher concentration of scores near 1.0 indicates strong matches and high model accuracy, while lower scores suggest less confidence in the retrieved answers. This analysis helps evaluate the effectiveness of the similarity matching mechanism.

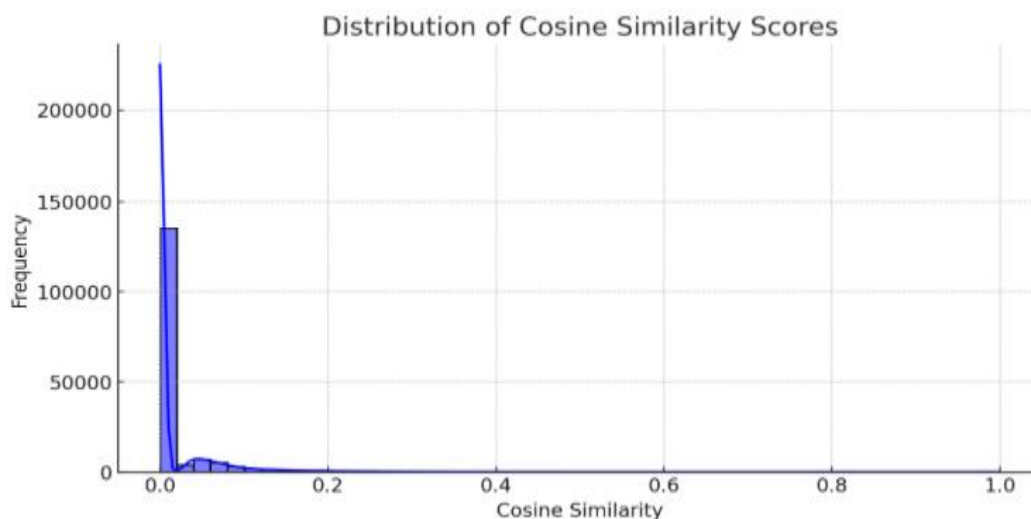


Figure. 5: Distribution of Cosine Similarity

Figure 5 illustrates how cosine similarity scores are distributed when comparing new queries with existing ones in the dataset. A higher concentration of scores near 1.0 indicates strong matches and high model accuracy, while lower scores suggest less confidence in the retrieved answers. This analysis helps evaluate the effectiveness of the similarity matching mechanism.

Figure 6 shows how the lengths of chatbot answers vary across the dataset. It measures the number of words per response and helps assess the consistency and completeness of the chatbot's replies. Most responses fall within a moderate word range, ensuring concise yet informative answers to student questions. This graph presents the distribution of the number of words in student queries. It reveals the complexity and variation in how students phrase their questions.

In figure 7, the data shows that most queries are short and direct, but there is also a significant number of longer, more detailed questions, indicating the need for strong contextual understanding in the NLP model.

C. User Experience Feedback

Some students and testers tried the chatbot and said it was easy to use. They liked the clean interface and quick answers.

The suggested questions helped them ask the right queries. However, some users pointed out that the chatbot didn't always understand very complex or vague questions.

D. Limitations Found

Although the chatbot worked well overall, it had some limits:

- It could not handle follow-up questions or continue conversations across multiple turns.
- It only worked in English.
- It did not automatically update when university rules or data changed.
- It worked only in the local environment and was not yet hosted online.

These results showed that the chatbot is effective but can still be improved in future updates.

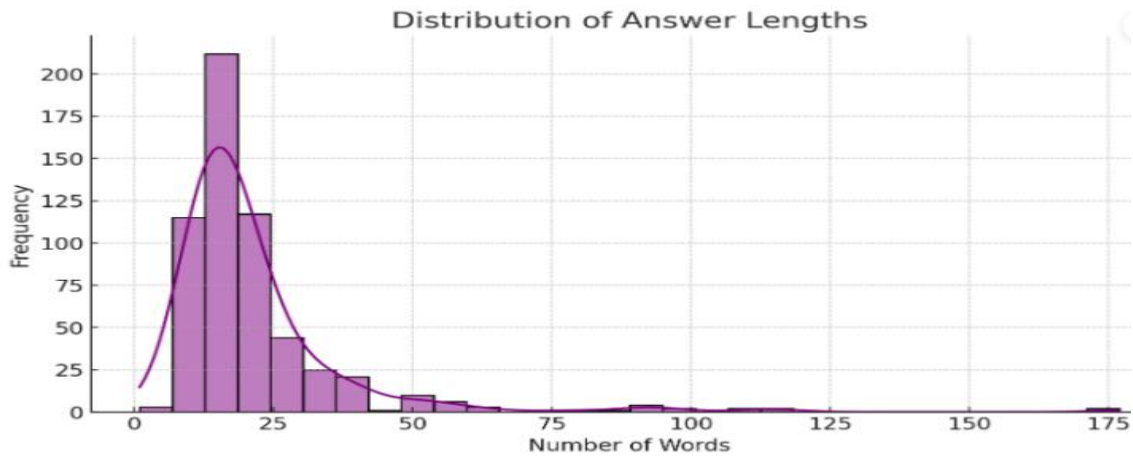


Figure. 6: Distribution of Answer Lengths

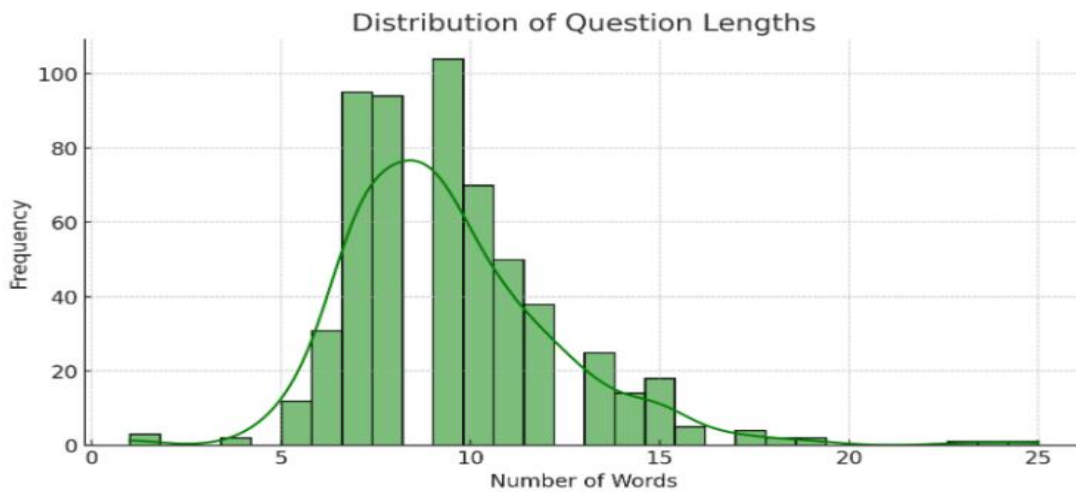


Figure. 7: Distribution of Questions Lengths

Conclusion

In conclusion, this research successfully developed an AI-powered chatbot that can help university students by answering their questions accurately and quickly. The chatbot used a mix of advanced NLP methods: SBERT, TF-IDF, and cosine similarity to understand student queries and give correct answers. The chatbot was tested thoroughly and showed good performance. Students found it easy to use and helpful for common questions. It saves time for both students and university staff by automating repetitive tasks.

However, the system also has some limitations. It currently only supports English, works offline, and cannot handle multi-turn conversations. These are areas that can be improved in the future. Planned upgrades include support for more languages, cloud deployment, and better memory to understand follow-up questions.

Overall, this project shows that a simple, well-designed chatbot can be a powerful tool in the education sector, making information more accessible and improving student satisfaction.

References

1. N. N. Khin and K. M. Soe, "University Chatbot using Artificial Intelligence Markup Language."
2. D. Ramandanis and S. Xinogalos, "Investigating the Support Provided by Chatbots to Educational Institutions and Their Students: A Systematic Literature Review," *Multimodal Technologies and Interaction*, vol. 7, no. 11. Multidisciplinary Digital Publishing Institute (MDPI), Nov. 01, 2023. doi: 10.3390/mti7110103.
3. Y. W. Chandra and S. Suyanto, "Indonesian chatbot of university admission using a question answering system based on sequence-to-sequence model," in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 367-374. doi: 10.1016/j.procs.2019.08.179.
4. M. S. Stepanov, V. G. Popov, and N. K. Fedorova, "The development of chatbot for university and college admission," *T-comm*, vol. 16, no. 10, pp. 51-56, 2022, doi: 10.36724/2072-8735-2022-16-10-51-56.
5. H. Agus Santoso et al. 2018, "Dinus Intelligent Assistance (DINA) Chatbot for University Admission Services," 2018 International Seminar on Application for Technology of Information and Communication, Semarang, Indonesia, 2018, pp. 417-423, doi: 10.1109/ISEMANTIC.2018.8549797.
6. F. Clarizia, F. Colace, M. Lombardi, F. Pascale, and D. Santaniello, "Chatbot: An education support system for students," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2018, pp. 291-302. doi: 10.1007/978-3-030-01689-0_23.
7. Loes Werkman." K. Peyton and S. Unnikrishnan, "A comparison of chatbot platforms with the state-of-the-art sentence BERT for answering online student FAQs," *Results in Engineering*, vol. 17, Mar. 2023, doi: 10.1016/j.rineng.2022.100856.
8. F. Amato, M. Fonisto, M. Giacalone, and C. Sansone, "An Intelligent Conversational Agent for the Legal Domain," *Information (Switzerland)*, vol. 14, no. 6, Jun. 2023, doi: 10.3390/info14060307.
9. G. Min and J. Yoo, "Shin, & Journal of the Korean Association for," *Science Education Journal homepage: www.koreascience.org Journal of the Korean Association for Science Education*, vol. 44, no. 3, p. 2024, 2019, doi: 10.14697/jkase.2024.44.3.231.
10. M. Anbiya, N. Islam, B. Warsito, and O. D. Nurhayati, "AI-driven chatbot implementation for enhancing customer service in higher education: a case study from universitas negeri semarang," *Journal of Theoretical and Applied Information Technology*, Vol. 31, No. 14, 2024, [online]. Available: www.jatit.org
11. S. Bistarelli and M. Cuccarini, "BERT-based question answering on close domains: Preliminary Report," 2022. [Online]. Available: <https://chat.openai.com> "DEGREE PROJECT Assessing the potential of leveraging LLaMA 2 to create an institute specific online chatbot assistant.
12. C. Chun Ho, H. L. Lee, W. K. Lo, and K. F. A. Lui, "Developing a Chatbot for College Student Programme Advisement," in *Proceedings - 2018 International Symposium on Educational Technology, ISET 2018*, Institute of Electrical and Electronics Engineers Inc., Sep. 2018, pp. 52-56. doi: 10.1109/ISET.2018.00021.
13. *International Conference on Advances in Computing, Communications and Informatics (ICACCI) : 13-16 Sept. 2017. IEEE, 2017.*
14. P. Darshan, E. Sanghavi, G. Chennakeshava, M. M. Kodabagi, B. H. Baba Fakruddin Ali and H. P. Gayathri, "AI Chatbot for Educational Institution: A Case Study," *2025 International Conference on Knowledge Engineering and Communication Systems (ICKECS)*, Chickballapur, India, 2025, pp. 1-5, doi: 10.1109/ICKECS65700.2025.11034867.
15. S. M. Hassan, A. Alghamdi, A. Hafeez, M. Hamdi, I. Hussain, and M. Alrizq, "An Effective Combination of Textures and Wavelet Features for Facial Expression Recognition", *Eng. Technol.*

Appl. Sci. Res., vol. 11, no. 3, pp. 7172–7176, Jun. 2021.

16. Hussain I, Zaidi SMH, Khan U, Ahmed A. Sentiment analysis classification of ChatGPT tweets using machine learning and deep learning algorithms. *Journal of Computing & Biomedical Informatics*. 2024;8(1).
17. Shaukat, S. S., Sheikh I. H. & Siddiqui I. A. (2005). An application of correspondence analysis, Detrended correspondence analysis and Canonical correspondence analysis to the vegetation and environment of calcareous hills around Karachi. *Int. J. Biol. Biotechnol.*, 2, 617–627.
18. Iqbal, S., Ahmed, S., Hassan, B., Abid, M. K., and Husain, I. (2024). Enhancing Lung Cancer Detection with Hybrid Machine Learning: Integrating Ant Colony Optimization. *Technical Journal, University of Engineering and Technology (UET) Taxila*. 29(3): 74-86.
19. Hassan, S. M., Ali, F., Wasi, S., Husain, I., and Ashraf, S. N. (2019). Roman-Urdu News Headline Classification with IR Models using Machine Learning Algorithms. *Indian Journal of Science and Technology*, 12(35):1-9.
20. Awan, S. A., Ali, S. A., Husain, I., Quraishi, B. H., Ashraf. S.M.A. (2021). Proficient Masked Face Recognition Method Using Deep Learning Convolution Neural Network in Covid-19 Pandemic. *International Journal of Circuits, Systems and Signal Processing*. 15(189). 7172-7176.
21. Husain, I., Khan, N.U. and Shaukat, S. S. (2015). Mathematical Techniques to Study the Spatial Pattern of *Meloidogyne javanica* in relation to soil fungi and soil characteristics in a tomato-grown field. *Research Journal of Recent Sciences*, 2277, Pp. 2502.
22. Hassan, S. M., Ali, S. A., Hassan, B., Husain, I., Rafiq, M. and Awan, S. A. (2022). Hybrid Features Binary Classification of Imbalance Stroke Patients Using Different Machine Learning Algorithms. *International Journal of Biology and Biomedical Engineering*, 16(1): 154-160.
23. Abro, M., Husain, I., Hassan, S. M., Sheikh, F., and Murtaza, G. (2025). A Predictive Model and Performance Evaluation in Mathematics for Primary Education. *Journal of Computing & Biomedical Informatics* 9(2):1-12.