

Intellichat: A Web-Based Domain-Specific Chatbot for Syllabus-Aligned Technical Learning

Abinaya.S¹, Darsana.S², Deepika.S³, Madhusri.M⁴, Anitha.M⁵

¹²³⁴⁵ Department of CSE, School of Engineering, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore 18.

22ueo002@avinuty.ac.in, 22ueo011@avinuty.ac.in, 22ueo012@avinuty.ac.in, 22ueo031@avinuty.ac.in,
anitha_cse@avinuty.ac.in

Abstract

Domain-specific academic support is essential for Computer Science and Engineering students, as technical subjects require accurate and syllabus-aligned explanations. Conventional learning resources such as textbooks and faculty consultations can be time-consuming and may not provide instant clarification. To overcome these limitations, this study presents Intellichat, a domain-specific chatbot designed to assist CSE students through real-time conversational learning. The system integrates a web-based interface with a Flask backend and utilizes a Large Language Model through Cohere API for generating intelligent responses. User queries are first processed using a keyword-based domain filtering mechanism to ensure that only Computer Science-related questions are accepted. Relevant prompts are then structured and forwarded to the language model, which produces context-aware academic explanations for subjects such as Data Structures, Operating Systems, DBMS, and Computer Networks. Experimental evaluation using response relevance, accuracy, and latency metrics demonstrates that Intellichat provides fast and syllabus-focused assistance compared to general-purpose chatbots.

Keywords: CN Domain-Specific Chatbot, Large Language Models, Computer Science Education, Cohere API, Prompt Engineering, Academic Assistant

1. INTRODUCTION

Academic learning in Computer Science and Engineering involves mastering complex theoretical and practical concepts that demand continuous clarification and structured guidance. As students progress through core subjects, they frequently encounter challenges in understanding algorithms, system operations, database concepts, and networking principles. While digital learning resources are widely available, they often provide scattered information and lack personalized interaction, making it difficult for learners to obtain precise answers aligned with their curriculum. Additionally, general conversational AI tools may respond with content that is not technically focused or academically appropriate for engineering-level requirements. To address the need for focused and reliable academic assistance, domain-restricted conversational systems have become an important area of research. By limiting responses to a specific discipline, such chatbots can deliver more relevant explanations and reduce ambiguity in technical

learning. Recent developments in large language models and intelligent prompt structuring have further enabled chatbots to generate meaningful educational support in real time. In this work, Intellichat is introduced as a Computer Science-specific chatbot that processes only subject-relevant queries and provides accurate, syllabus-oriented explanations through Cohere-based language modeling. The proposed system aims to enhance student learning efficiency and support self-paced education through an interactive academic companion.

2. OBJECTIVE

The goal of this work is to develop an automated, domain-specific academic chatbot system using large language models and natural language processing techniques. The proposed approach involves accepting student queries related to Computer Science Engineering subjects, filtering them through a domain-restriction mechanism, and generating syllabus-aligned responses using trained language model APIs. The system aims to

provide an efficient, reliable, and easily accessible learning companion by reducing dependence on conventional resources such as textbooks and limited faculty availability. This strategy supports real-time doubt clarification, enhances self-directed learning, and promotes scalable academic assistance, especially for students requiring continuous support outside classroom environments.

3. LITERATURE REVIEW

Several studies have explored the development of educational chatbots using artificial intelligence and large language models for academic assistance. Sundar and Roy (2022) demonstrated that prompt engineering techniques can significantly improve response relevance in curriculum-focused chatbot systems. Kumari and Sharan (2023) presented a survey on chatbot implementation approaches and highlighted that generative NLP-based systems provide greater flexibility compared to traditional rule-based methods. Jain and Menon (2023) examined the limitations of general-purpose LLMs such as ChatGPT in domain-specific education, emphasizing the risk of inaccurate or overly broad responses when applied to technical subjects. Verma and Sinha (2023) developed an NLP-based chatbot for computer science syllabus support, showing improved learning outcomes through restricted academic interaction. More recently, Thomas and Srinivasan (2024) investigated Retrieval-Augmented Generation (RAG) frameworks for university chatbots to reduce hallucination and improve factual grounding.

4. RISK REDUCTION AND DECISION SUPPORT STRATEGY

To increase the reliability of academic assistance, the proposed Intellichat system integrates a risk reduction and decision support approach. By utilizing a domain-specific chatbot framework combined with keyword-based filtering and structured prompt engineering, the system reduces the risk of irrelevant responses. Response accuracy and consistency are improved through controlled interaction with a large language model API, ensuring that only Computer Science Engineering-related queries are processed. Intellichat supports decision-making by helping students understand topics effectively and clarify doubts in real time. This strategy promotes confident self-learning, reduces dependence on

unreliable online sources, and encourages timely academic progress with minimal uncertainty.

5. SYSTEM ARCHITECTURE AND COMPONENTS

To guarantee accuracy, scalability, and efficiency, the suggested system architecture for Intellichat follows a modular and structured design. The system begins with a user interaction component where students enter academic queries through a web-based chatbot interface. The query handling module then processes the input and performs basic preprocessing such as cleaning text and formatting the query. The domain filtering component plays a key role by verifying whether the query belongs to Computer Science Engineering subjects. Only syllabus-relevant questions are forwarded for further processing, while unrelated queries are rejected with fallback responses. The prompt construction module then structures the validated query into an academically meaningful instruction for better response generation. The large language model integration component communicates with Cohere's API to generate context-aware and technically accurate answers. The response management module post-processes the generated output and delivers it back to the user in real time. User interactions and chatbot responses can be stored securely in a data management module for future improvement and analysis. Finally, the output module displays clear and understandable academic explanations to students. This architecture enables an automated, reliable, and domain-specific framework for real-time academic support.

6. METHODOLOGIES

6.1 Query Collection

Student queries related to Computer Science Engineering subjects are collected through the chatbot interface. These queries may include topics such as Data Structures, OS, DBMS, Algorithms, and Networking. Each query forms the primary input for chatbot response generation.

6.2 Input Preprocessing

Preprocessing is applied to improve query consistency and quality. This includes removing unnecessary symbols, handling incomplete queries, and standardizing the text format. This step ensures smooth downstream processing and accurate response generation.

6.3 Domain Validation

A keyword-based domain validation mechanism is used to ensure that only Computer Science-related questions are processed. Queries outside the defined domain are filtered out to prevent irrelevant responses and maintain academic focus.

6.4 Prompt Construction

The validated query is transformed into a structured academic prompt. This step improves response relevance by guiding the language model to generate syllabus-aligned and technically correct explanations.

6.5 Feature Selection

Techniques such as controlled prompt engineering and response formatting are applied to improve consistency, reduce ambiguity, and ensure clarity. This enhances the overall chatbot reliability and reduces misleading outputs.

6.6 Model Interaction and Response Generation

The structured prompt is forwarded to Cohere's Large Language Model API. The model generates a context-aware response based on trained transformer architectures. The chatbot then classifies the response as academically valid and delivers it to the student.

6.7 Result Delivery and Analysis

The generated academic explanation is displayed instantly to the user along with meaningful clarity. This supports real-time doubt clarification, improves understanding, and enables effective self-learning and decision-making during exam preparation.

7. IMPLEMENTATION

The Intellichat system is implemented as a web-based domain-specific academic chatbot that integrates a lightweight frontend with a Flask-based backend and Cohere Large Language Model API for response generation. The implementation is designed to ensure real-time interaction, syllabus relevance, and controlled response delivery for Computer Science Engineering students. The system architecture follows a client-server model, where user queries are processed in the backend and forwarded to the language model only after domain validation.

7.1 Frontend Interface

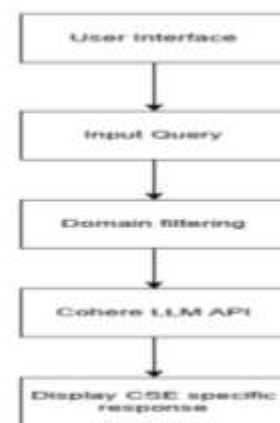
The chatbot interface is developed using HTML, CSS, and JavaScript. Students enter academic questions through an input box, and responses are displayed dynamically in the chat window. JavaScript handles asynchronous communication using AJAX or Fetch API, enabling smooth interaction without page reloads.

7.2 Backend using Flask

The core response generation is performed using Cohere's Large Language Model API. After validation, the backend sends a structured prompt to Cohere's generate endpoint. The API returns a context-aware academic explanation, which is then post-processed and delivered to the student.

The API key is securely stored and used for authentication during each request. Response parameters such as token limit and temperature are controlled to ensure consistent academic answers. The backend is implemented using the Flask framework in Python. Flask manages user requests, performs query preprocessing, applies domain filtering, constructs structured prompts, and communicates with Cohere's API. The backend exposes REST endpoints that receive user input and return chatbot responses in JSON format.

8. BLOCK DIAGRAM



9. RESULT AND FEATURES

The proposed Intellichat system offers an intelligent and domain-specific solution for real-time academic assistance in Computer Science Engineering education.

Through an easy-to-use chatbot interface, students can submit their subject-related queries and receive instant responses, while secure user authentication supports controlled access. The system automatically processes user input through preprocessing, domain validation, and structured prompt construction to ensure syllabus relevance. Large language model-based response generation

is used to provide accurate explanations for topics such as Data Structures, Operating Systems, DBMS, and Computer Networks. According to experimental evaluation, the chatbot delivers dependable academic responses with high relevance and reduced response time. Compared to general-purpose conversational tools, the domain-restricted approach improves accuracy, prevents off-topic interactions, and enhances student learning efficiency. The system supports effective doubt clarification, promotes self-directed study, and assists learners in making informed academic progress. Overall, the features and outcomes demonstrate that Intellichat is a practical and scalable academic companion for syllabus-oriented technical education.

10. CONCLUSION

Intellichat is an artificial intelligence-based domain-specific chatbot designed to support Computer Science Engineering students with syllabus-aligned academic assistance. By applying query preprocessing, domain filtering, and structured prompt engineering, the system ensures accurate and relevant responses through Cohere's Large Language Model API. Intellichat serves as a practical alternative to general-purpose chatbots by providing focused, real-time doubt clarification and promoting self-learning in technical education. Future enhancements of Intellichat may include the incorporation of Retrieval-Augmented Generation (RAG) to provide more grounded answers from institutional study materials, improved multi-turn context handling for deeper conversations, and support for multilingual interaction to assist a wider range of learners. Additional features such as personalized learning recommendations, integration with academic resources, and analytics dashboards for monitoring student progress can further strengthen the system's impact. With

these improvements, Intellichat has strong potential to evolve into a comprehensive intelligent academic assistant for modern digital education.

11. USES

The domain-specific academic chatbot Intellichat can be successfully used as an intelligent learning support tool for Computer Science and Engineering education. Because of its accessibility and ease of use, it is especially helpful for continuous self-learning and exam preparation. The chatbot can be integrated into web-based and institutional learning platforms, allowing students to interact with an academic assistant anytime and from anywhere. Additionally, by providing automated academic guidance for technical topics, the system can assist educators in reducing repetitive query handling. Its suitability for remote learning environments and resource-constrained educational settings makes it a valuable tool for improving scalable and personalized academic support.

ACKNOWLEDGEMENT

We, the students of CSE, want to express our heartfelt thanks to our guide, Assistant. prof. M. Anitha, for their essential help, support, and encouragement during this project. We also appreciate the Department of CSE at Avinashilingam Institute for giving us the resources and facilities we needed. Lastly, we would like to recognize our team members for their support and teamwork, which were crucial to finishing this work successfully.

REFERENCES

- [1] Kumari, P., & Sharan, M. "A Survey on Chatbot Implementation Using Natural Language Processing Techniques," 2023. *International Journal of Computer Applications*.
- [2] Jain, A., & Menon, R. "ChatGPT and Beyond: Challenges of Large Language Models in Domain-Specific Education," 2023. *IEEE Transactions on Artificial Intelligence in Education*, vol. 17, no. 3, pp. 123–131.
- [3] Sundar, K., & Roy, A. "Prompt Engineering for Domain-Specific Conversational Agents," 2022. *Proceedings of the International Conference on NLP Applications (ICNLPA)*, pp. 55–61.
- [4] Verma, B., & Sinha, K. "Design of an NLP-Based Chatbot for Computer Science Curriculum Support," 2023. *International Journal of Intelligent Systems and Applications*, vol. 15, no. 1, pp. 33–40.
- [5] Thomas, A., & Srinivasan, L. "Retrieval-Augmented Generation for Academic Chatbots in Higher

Education,” 2024. *Proceedings of the ACM Conference on Learning Technologies*, pp. 97–104.

[6] Gupta, M., & Prasad, H. “Evaluating User Engagement and Effectiveness of Educational Chatbots,” 2022. *IEEE International Conference on Human-Centered Computing*, pp. 89–96.

[7] Patel, R., & Rao, N. “Large Language Model Based Virtual Assistants for Academic Support Systems,” 2023. *Springer Lecture Notes in Educational Technology*, vol. 85, pp. 212–225.

[8] Sharma, T., & Das, V. “Conversational AI in Smart Learning Environments: A Review,” 2021. *Journal of Educational Technology Systems*, vol. 50, no. 4, pp. 371–385.

[9] Sengupta, J., & Kaur, S. “Challenges in Deploying Domain-Restricted Language Models for Reliable Assistance,” 2022. *Computers and Education Journal*, vol. 95, pp. 58–64.

[10] Cohere AI, “Cohere Large Language Model API Documentation,” 2024.

[11] Mishra, R., & Arora, K. “Role of Domain-Specific Chatbots in Engineering Education Support,” 2023. *International Journal of Computer Science Education*, vol. 18, no. 2, pp. 110–119.