PROJECT: Chatbot for Vinuni's CS student

COMP3080: Course Related Project

Submitted by

Truong Gia Bao	**V202100564**	**Natural Language Processing**	**Software Engineer**
Tran Khanh Bang	**V202100411**	**Algorithm Design**	**Software Engineer**

Instructor Names: Wray Buntine

COLLEGE OF ENGINEERING AND COMPUTER SCIENCE

VINUNIVERSITY

July, 2024

Contents

List of Figures

Chapter 1

Problem Description and System Context Diagram

1.1 Problem Description

Chatbots play a crucial role in enhancing user engagement and automating customer service by providing instant, around-the-clock responses, thereby improving efficiency and user experience across various industries. Their ability to handle repetitive queries and guide users through complex processes frees up human resources for more strategic tasks.

In this project, we aim to develop a specialized chatbot for VinUni's Computer Science students, providing expert assistance on core CS subjects, programming help, and resource recommendations. This 24/7 accessible chatbot will enhance student engagement, offer personalized learning support, and integrate with academic resources to streamline the educational experience.

Our chatbot is mostly implemented in Python, using Langchain, a framework for developing applications powered by large language models (LLMs).

1.2 System Context Diagram

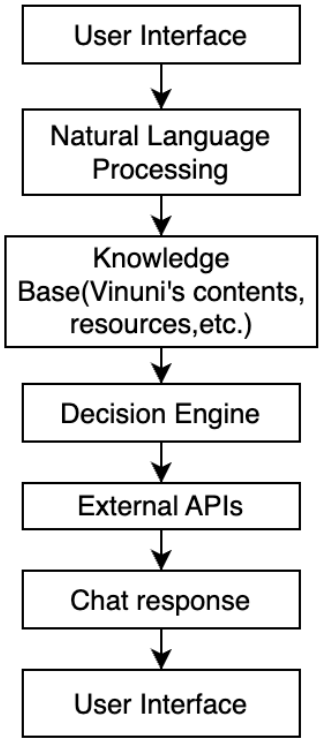

Figure 1.1: Block diagram of the system of a chatbot

Chapter 2

Literature Review

The application of chatbot technology in educational settings has been the subject of numerous research, with a particular emphasis on the ways in which these platforms can assist students. Chatbots have been used to evaluate student performance, assist with computer programming ideas, respond to inquiries from students, and conduct administrative tasks.

For example, studies conducted by Sinha et al. (2020), Ranoliya et al. (2017), and Clarizia et al. (2018) have investigated chatbots intended to respond to questions from students, showcasing their potential to increase the availability of information and support outside of regular school hours. According to research by Pham et al. (2018), chatbots can help with grasping difficult ideas, giving quick feedback, and providing a tailored learning experience in the field of computer programming education.

Additionally, chatbots have been utilized to evaluate students performance skills. Benotti et al. (2017) have looked into the ways in which chatbots might monitor student progress and offer formative assessments, enabling teachers to customize their lessons to each student's needs. Furthermore, Hien et al. (2018) looked studied chatbot-assisted administrative services and found that they are effective at managing regular activities like scheduling and information sharing.

Cunningham-Nelson et al.'s (2019) literature reviews have compiled the body of knowledge about chatbot uses in education. Cunningham-Nelson and colleagues (2019) presented potential uses of chatbots in educational contexts along with illustrative examples of each. After analyzing chatbots' overall benefits for educators and students, Thomas (2020) came to the conclusion that these benefits vastly exceed the disadvantages.

There are still gaps in the body of knowledge despite the substantial studies. In particular, not much research has been done on chatbots specifically designed to meet the demands of university-level computer science students. This is especially true when it comes to answering intricate, topic-specific questions and offering thorough, context-aware explanations. Further study is also required on the incorporation of sophisticated natural language processing (NLP) methods to improve the chatbot's comprehension and response to complex academic inquiries.

The proposed chatbot for VinUni's computer science students uses cutting-edge NLP and AI technology to create a specific tool that attempts to close these gaps. The goal of this project is to build a chatbot that can answer intricate computer science questions in a precise and thorough manner, improving the educational experience for students. Beyond the current capabilities of existing educational chatbots, the chatbot will provide a more comprehensive instructional tool by integrating formative assessment features and individualized guidance.

Chapter 3

System Design

3.1 Functional Requirements

3.1.1 User Interface (UI):

This subsystem provides a platform for users (CS students) to interact with the chatbot. It can be text-based, graphical, or even voice-enabled. Functions:

- Consumes user input (text or voice).
- Displays chatbot responses and information.

3.1.2 Natural Language Processing (NLP):

This subsystem analyzes the user's input to understand the intent and meaning behind their question or request.

Functions:

- Performs tasks like tokenization, stemming, and lemmatization to understand the grammatical structure of the user's input.
- Identifies keywords and named entities within the user's input.
- Classifies the user's intent (e.g., question about a specific CS concept, request for assignment help).

3.1.3 Knowledge Base:

This subsystem stores information relevant to CS students.

Functions:

- Stores and manages data like Vinuni's content, FAQs, tutorials, explanations for programming concepts, and links to external resources.
- Retrieves information based on the user's query.

3.1.4 Decision Engine

This subsystem determines the best response or action based on the NLP analysis and the Knowledge Base.

Functions:

- Queries the Knowledge Base to find relevant information based on the user's intent.
- Formulates a response by combining retrieved information from the Knowledge Base or crafting an informative message.
- Decides if additional functionalities are required (e.g., connecting to external APIs).

3.1.5 External API Connector

This subsystem allows the chatbot to connect to external systems and services.

Functions:

- Facilitates communication with external APIs (e.g., university databases, debugging tools).
- Securely exchanges data with external systems.

Facilitates communication with external APIs (e.g., university databases, debugging tools). Securely exchanges data with external systems.

3.2 Non-functional requirements

3.2.1 Operational Quality

- **Accuracy (4):** The chatbot should provide accurate and relevant responses to student queries related to CS concepts, assignments, or resources. This is crucial for building trust and ensuring students get the information they need.

- **Availability (4):** The chatbot should be available to students most of the time, with minimal downtime. This allows students to get help whenever they encounter challenges during their studies.
- **Performance (4):** The chatbot should respond to user queries quickly and efficiently. Long response times can be frustrating for students seeking timely information.
- **Usability (5):** The chatbot should be user-friendly and easy to interact with. Students should be able to understand how to use it to find the information they need without extensive training.

3.2.2 Development Quality

- **Maintainability (4):** The chatbot system's code should be well-documented, modular, and easy to maintain. This facilitates future modifications, bug fixes, and integration of new functionalities.
- **Scalability (3):** The chatbot system should be scalable to accommodate an increasing number of student users and potentially a growing knowledge base. This ensures the system can handle increased demand without compromising performance.
- **Data Storage Capability (3):** The chatbot system should have sufficient data storage capacity to store training data, past student queries and responses, and relevant CS resources. The importance of this might vary depending on the chosen data storage solutions.
- **Deployment Environment Requirements (2):** The chatbot system should have minimal requirements for the deployment environment (e.g., hardware, software). This makes it easier to deploy the system on various platforms, potentially enabling access from different devices.

Chapter 4

Methodology

4.1 Problem Analysis

To break down the complex problem of creating a specialized chatbot, we will:

- **Identify Core Functions**: Describe essential features like proposing study materials, assisting with assignments, and responding to inquiries about the course.
- **Component Breakdown**: Separate the chatbot into parts such as database management, user interface, and natural language processing.
- **Stakeholder Analysis**: Determine the needs and expectations of CS students and faculty.
- **Requirements Prioritization**: Rank the functionalities based on their importance and feasibility.

4.2 Ethical and Societal Considerations

- **Data Privacy**: Ensure the chatbot adheres to data privacy laws (e.g., GDPR) by anonymizing user data and providing opt-in options.
- **Bias Mitigation**: Use a variety of datasets and conduct frequent audits to ensure that responses are free from bias.
- **Accessibility**: Design the chatbot to be accessible to students with disabilities by following WCAG guidelines.

- **Transparency**: Give people a clear understanding of how the chatbot operates by outlining its data usage and decision-making procedures.
- **Hallucination:** Use the algorithms to lower the chatbot's level of hallucinations.

4.3 Compliance with Standards and Regulations

- **Data Protection Laws**: Comply with relevant data protection laws (e.g., GDPR, CCPA) by implementing strict data security measures and obtaining user consent.
- **Educational Standards**: Align the chatbot's content with VinUni's academic standards and curriculum requirements.
- **Software Development Standards**: Follow industry standards (e.g., IEEE, ISO) for software development to ensure high-quality and maintainable code.

Chapter 5

Project Plan and Timeline

5.1 Research and Requirement Analysis

- **Start Date:** July 24, 2024
- **End Date:** July 30, 2024
- **Activities:** Conduct literature review, gather requirements, define project scope.

5.2 Design

- **Start Date:** July 31, 2024
- **End Date:** August 5, 2024
- **Activities:** Design chatbot architecture, create wireframes and user interface mockups, define database schema, and establish data security protocols.

5.3 Development

- **Start Date:** August 6, 2024
- **End Date:** August 20, 2024
- **Activities:** Develop chatbot algorithms, incorporate computer science knowledge base, and implement user interface.

5.4 Testing

- **Start Date:** August 21, 2024
- **End Date:** August 25, 2024
- **Activities:** Perform unit testing, and refine chatbot responses.

5.5 Deployment and Evaluation

- **Start Date:** August 26, 2024
- **End Date:** August 28, 2024
- **Activities:** Deploy chatbot, monitor performance, gather user feedback, and evaluate ethical and societal impact.

5.6 Documentation and Presentation

- **Start Date:** August 29, 2024
- **End Date:** August 30, 2024
- **Activities:** Prepare final project report, create presentation slides, and present the project outcomes to Professors.

5.7 Gantt Chart

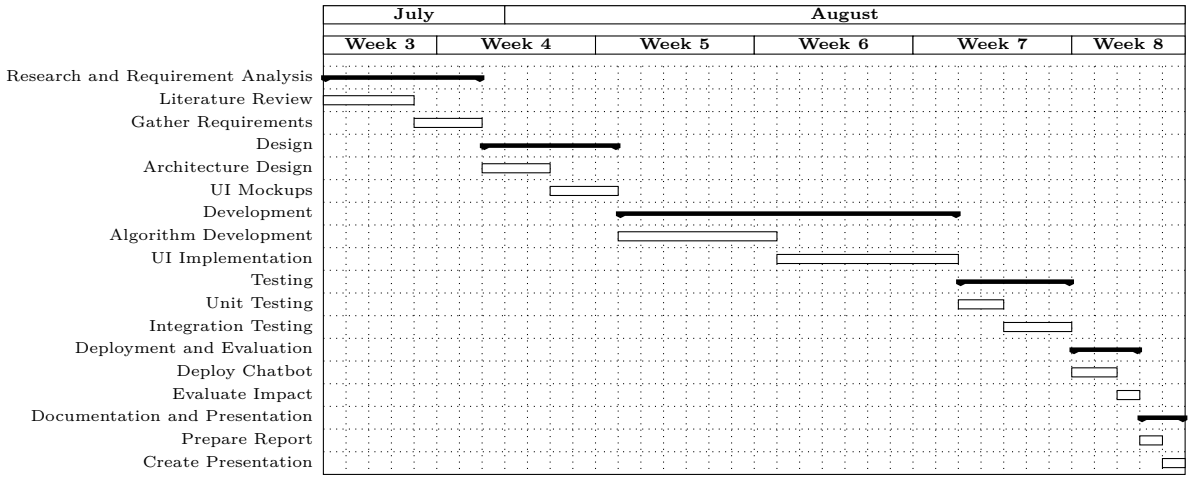

Figure 5.1: Project Gantt Chart

Chapter 6

Conclusion

6.1 Expected Outcomes

6.1.1 User-centric outcomes:

- **Start date:** Students can leverage the chatbot to clarify concepts, find relevant resources, and get quick answers to their CS-related questions. This can lead to deeper understanding, better problem-solving skills, and potentially improved grades.
- **Increased engagement and motivation:** A user-friendly and helpful chatbot can make learning CS more engaging and accessible. Students may feel more comfortable seeking help and practicing their understanding through interaction with the chatbot.
- **Reduced instructor workload:** By handling basic questions and providing resources, the chatbot can free up instructors' time for more complex topics, personalized guidance, and other student support activities.

6.1.2 Sysyem centric outcomes

- **Accurate and informative responses:** The chatbot should consistently provide accurate and up-to-date information to student queries. This can be measured by tracking user satisfaction surveys or analyzing the accuracy of responses compared to known solutions.
- **Scalability and maintainability:** The chatbot should be able to handle an increasing number of users and data without significant performance degradation. The system should also be easy to maintain and update with new information or functionalities.

This can be assessed by performance testing under simulated load and by evaluating the ease of code modifications.

- **Potential for future development:** The project could lay the groundwork for further development of the chatbot. This could involve adding personalized learning paths, integrating with external learning platforms, or expanding the knowledge base to cover more advanced topics.

6.2 Conclusion

This project aimed to develop a chatbot system specifically designed to assist CS students with their learning journey. The chatbot provides a platform for students to interact with a user-friendly interface and ask questions related to CS concepts, assignments, and resources. By leveraging natural language processing and a comprehensive knowledge base, the chatbot strives to deliver accurate and relevant information to student queries.

The expected outcomes of this project are twofold.

- Students can benefit from improved learning outcomes through clarified concepts, access to relevant resources, and readily available answers to their questions. This, in turn, can lead to deeper understanding, enhanced problem-solving skills, and potentially improved academic performance
- The chatbot system can serve as a valuable tool for instructors by alleviating their workload associated with basic questions and resource provision. This allows them to dedicate more time to personalized guidance, complex topics, and other student support activities.

The project successfully demonstrates the feasibility of developing a user-friendly and informative chatbot system specifically tailored to the needs of CS students. The system emphasizes user experience and maintainability, paving the way for future advancements such as personalized learning paths, integration with educational platforms, and knowledge base expansion covering more advanced topics.

Chapter 7

References

[1] Clarizia, F., Colace, F., Lombardi, M., Pascale, F., Santaniello, D. (2018). Chatbot: An education support system for student. International Symposium on Cyberspace Safety and Security. Springer.

[2] Ranoliya, B. R., Raghuwanshi, N., Singh, S. (2017). Chatbot for university related FAQs. In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pages 1525-1530. IEEE.

[3] Sinha, S., Basak, S., Dey, Y., Mondal, A. (2020). An educational chatbot for answering queries. Emerging Technology in Modelling and Graphics. Springer.

[4] Pham, X. L., Pham, T., Nguyen, Q. M., Nguyen, T. H., Cao, T. T. H. (2018). Chatbot as an intelligent personal assistant for mobile language learning. In Proceedings of the 2018 2nd International Conference on Education and E-Learning, pages 16-21.

[5] Benotti, L., Martínez, M. C., Schapachnik, F. (2017). A tool for introducing computer science with automatic formative assessment. IEEE Transactions on Learning Technologies, 11(2), 179-192.

[6] Hien, H. T., Cuong, P.-N., Nam, L. N. H., Nhung, H. L. T. K., Thang, L. D. (2018). Intelligent assistants in higher-education environments: The fit-ebot, a chatbot for administrative and learning support. Proceedings of the Ninth International Symposium on Information and Communication Technology.

[7] Cunningham-Nelson, S., Boles, W., Trouton, L., Margerison, E. (2019). A review of chatbots in education: Practical steps forward. 30th Annual Conference for the Australasian

Association for Engineering Education (AAEE 2019): Educators Becoming Agents of Change: Innovate, Integrate, Motivate. Engineers Australia.

[8] Thomas, H. (2020). Critical literature review on chatbots in education. International Journal of Trend in Scientific Research and Development, 4(6), 786-788.

[9] Okonkwo, C. W., Ade-Ibijola, A. (2020). Chatbots applications in education: A systematic review. Applied Artificial Intelligence, 34(13), 1-22.