

Technical University of Munich

School of Computation, Information and Technology
– Informatics –

Proposal for Bachelor's Thesis in Informatics

**Enhancing Contextual Awareness of
IRIS: Incorporating Lecture Content into
a GPT-based Educational Chatbot on the
Artemis Learning Platform**

Erweiterung des Kontextverständnisses von
IRIS: Integration von Lehrinhalten in einen
GPT-gestützten pädagogischen Chatbot
innerhalb der Artemis-Lernumgebung

Author:	Yassine Souissi
Supervisor:	Stephan Krusche
Advisor:	Patrick Bassner
Start Date:	15.11.2023
Submission Date:	15.03.2023

1 Introduction

In the dynamic realm of higher education, platforms like Coursera¹, Udacity², and Khan Academy³ have already set the stage for a global educational network, connecting students and educators like never before. However, in conventional university contexts, especially in large-scale courses such as Computer Science, the need for personalized academic support remains unmet.

Addressing this gap, the Technical University of Munich is at the forefront of educational innovation with the development of Artemis⁴ and IRIS. Artemis, designed to be flexible and scalable, accommodates extensive student cohorts with features such as live streaming, interactive quizzes, and automated exercise evaluations. [Kru21] These innovations aim to assist educators in managing the increased demands of course delivery and in sustaining the integrity of educational feedback mechanisms. [JPB22]

Adding to this transformative landscape is the introduction of IRIS, an intelligent virtual tutor. Embedded within Artemis, IRIS represents a leap towards AI-driven interactive learning. Leveraging Generative AI and Large Language Models (LLMs), IRIS is specifically tailored to aid students in solving programming tasks. This virtual tutor is not just a repository of information but an interactive tool that provides one-on-one assistance, fostering deep understanding and autonomy in solving programming exercises.

With Artemis and IRIS, the Technical University of Munich is not just responding to the challenges of modern education but redefining the very essence of learning in the digital age. These platforms symbolize a future where education is more adaptive, responsive, and effective, tailored to students' diverse and dynamic needs worldwide.

¹<https://www.coursera.org/>

²<https://www.udacity.com/>

³<https://www.khanacademy.org/>

⁴<https://artemis.ase.in.tum.de/>

2 Problem

Within the Artemis initiative, the Computer Science education sector is confronting significant challenges that undermine the student learning experience. Despite the support from teaching assistants, a dedicated chair, and tutors, there is a notable deficiency in personalized academic support. This shortfall is largely attributed to the soaring enrollments, often surpassing 1000 undergraduates and 500 postgraduates, which stretch the educators' capacity to provide prompt and individualized responses an essential element for effective learning. [SK20]

Exacerbating this situation is the widespread use of standardized lecture materials. These materials, designed for mass dissemination, fail to cater to the diverse educational needs and backgrounds of a large student body. Consequently, this one-size-fits-all approach leaves many students lacking in necessary, personalized academic assistance.

A further complication is the difficulty in providing immediate responses to student inquiries during lectures. This challenge stems not only from the educators' limited capacity to address so many questions in real time but also from students' reluctance to ask questions, possibly due to fear of judgment or the sheer volume of queries. Such impediments in timely communication hinder students' comprehension of vital Computer Science concepts, adversely affecting their academic development.

In the midst of these challenges, IRIS currently provides feedback on programming exercises and aids in the understanding of related concepts. However, IRIS's functionality is constrained by its lack of awareness of the lecture content, limiting its ability to offer comprehensive support tailored to the curriculum. As a result, while IRIS serves as an invaluable tool for certain aspects of the learning process, its potential remains partially untapped in the broader educational context.

3 Motivation

The imperative to enhance Computer Science education is highlighted by the inadequacies of traditional teaching methods in large-scale academic settings.

Traditional interactive learning methods falter under the weight of large student numbers, where personalized engagement is challenging. IRIS is designed to overcome these barriers, offering scalable and adaptive interactions that cater to individual student needs, a feat unattainable in extensive lecture formats.

The absence of adequate guidance is a recognized impediment in the learning process. [Kru21] [PAKC06] IRIS seeks to mitigate this by providing immediate, tailored assistance, enabling students to navigate the intricacies of Computer Science education effectively. This support is crucial for preventing the development of misconceptions and ensuring a robust conceptual understanding.

Embracing a student-centered learning approach, IRIS is envisioned to foster an educational environment that adapts to the diverse requirements of learners. This strategy enhances inclusivity and encourages students to actively participate in their educational journey.

Additionally, the insights gleaned from student interactions with IRIS offer educators actionable data to refine and optimize lecture content. This feedback loop is essential for the ongoing enhancement of course material and teaching strategies.

The enhancement of IRIS represents a significant advancement in addressing the prevalent challenges within Computer Science education. It promises to elevate the quality of learning experiences and prepare students to thrive in a dynamic landscape.

4 Objective

The focus of this thesis is to enhance the contextual awareness of IRIS by incorporating lecture content into the GPT-based educational chatbot on the Artemis Learning Platform. This development aims to streamline IRIS's ability to address student inquiries related to lecture content more effectively. The focus here is on leveraging the existing integration of GPT-3.5 in IRIS and enriching it with a structured, easily accessible vector database of lecture materials. [Ope23]

By embedding these materials into a vector database, IRIS will be equipped to perform more accurate and relevant searches in response to specific student queries. This process involves the creation of lecture content embeddings, ensuring that IRIS can efficiently access and utilize this information during student interactions.

The project will adhere to best practices in software engineering. The deliverables of this thesis will be a fully functional integration of the vector database with IRIS, and documentation detailing the design and implementation process. The aim is to demonstrate how this enhancement can provide more precise and contextually relevant support to students, thereby enriching their learning experience and potentially reducing the workload on human tutors.

4.1 Proposed High-Level Workflow

Proposed High-Level Workflow

Upon receiving a query, the Large Language Model (LLM) promptly processes the input, transforming the natural language question into a query embedding. This embedding will be the basis for a semantic search within the knowledge base, which is to be constructed utilizing a vector database capable of housing the pre-encoded lectures. The semantic search executed by Artemis delves into the vector database, retrieving a ranked list of results pertinent to the user's query. These results are critically analyzed by the LLM, which then crafts a coherent and contextually relevant response. [Ope23] This interaction is envisioned to be depicted in an activity diagram, as outlined in Figure 2.

In the selection of the vector database, OpenAI's recommendations will be taken under advisement, with the final choice being contingent upon a multifaceted evaluation. Criteria such as extensibility, performance, integrability, and sustainability will be paramount in this decision-making process. The goal is to ensure that the database not only meets the current func-

tional requirements but is also robust enough to support future expansions and integrations.

During the processing of student questions, a specialized prompt for the LLM will be developed to effectively extract and interpret relevant lecture content. This prompt will be tailored to understand the context of each inquiry, ensuring that the responses provided are both accurate and directly relevant to the specific content of the lectures.

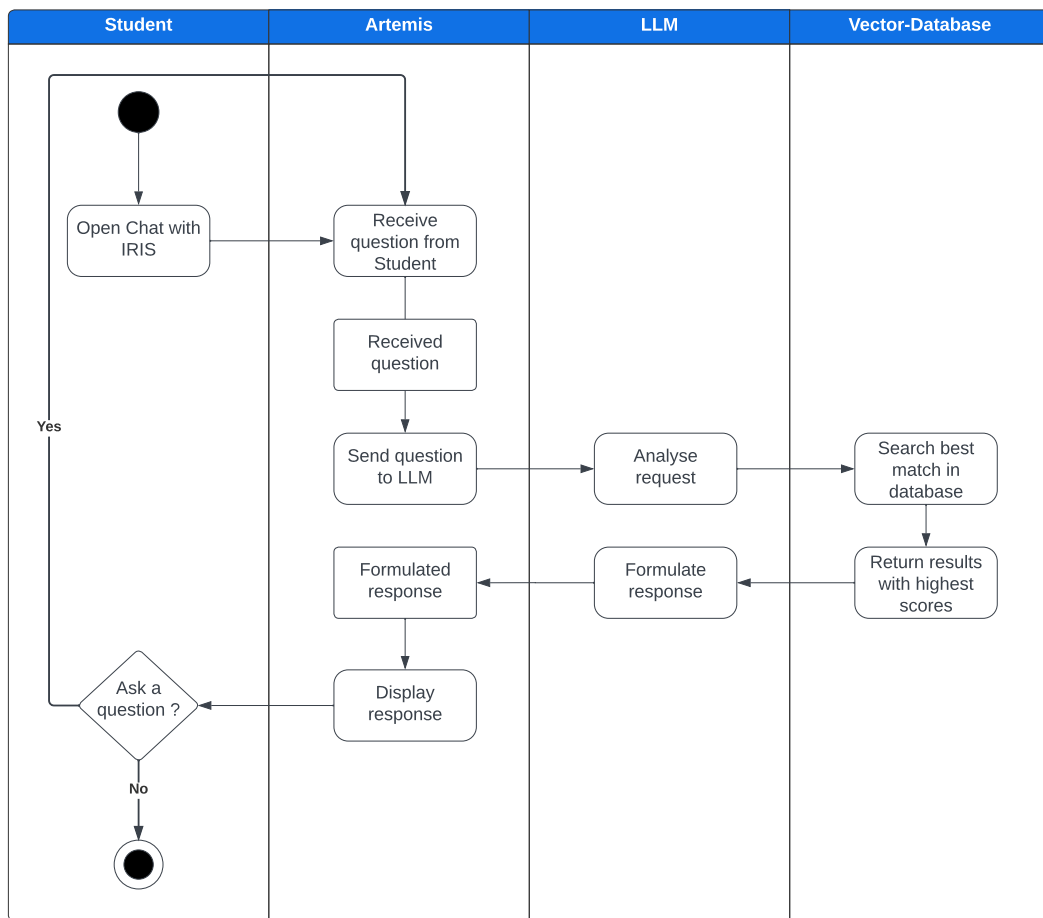


Figure 1: A detailed description of the diagram that serves as a legend.

5 Schedule

- **15.11 Thesis Kickoff**
- **Until 01.12: Initial Requirement Synthesis and Concept Validation**
 - Perform an exhaustive evaluation of the project’s prerequisites, covering both the technical framework and functional specifications.
 - Assess a range of established methodologies and Language Model paradigms for their suitability in achieving the project’s goals.
 - begin implementing A Minimal Viable Product for testing.
- **Until 15.12: Core Prototype Construction)**
 - Develop an elementary prototype in Python to showcase the essential capabilities of the Language Model with integrated long-term memory features, fulfilling the defined criteria.
- **Until 01.01: Enhanced Prototype Development**
 - Engineer an additional prototype with minimal viable functionality, utilizing Python to exhibit foundational Language Model operations coupled with extended memory functionalities, conforming to the established requirements.
- **Until 15.01: Comparative Analysis and Integration Selection**
 - Execute a comprehensive evaluation of the employed technologies and elect one for integration into the Artemis framework.
 - Begin the integration into Artemis.
- **Until 01.02: Finish the integration into Artemis and begin testing**
 - Undertake extensive validation by integrating the prototype into the Artemis ecosystem to confirm seamless incorporation and functionality.
 - begin testing the product with the Artemis team.
- **Until 15.02: Integration within Artemis**

-
- incorporate the feedback from the Artemis team
 - begin testing the product with together with the students.

- **Until 01.03: Finalization and Pre-Deployment Optimization**

- Allocate this period for rigorous debugging, rectification of any aberrant behavior, and refinement to a polished Minimum Viable Product (MVP).
- Prime the project for a seamless transition into the operational phase during the forthcoming Summer Semester 24/25, ensuring its deployment readiness.
- Compile exhaustive user manuals and documentation to ensure a smooth transition into active use.

References

- [JPB22] Bernd Bruegge Jan Philip Bernius, Stephan Krusche. *Machine learning based feedback on textual student answers in large courses*. 2022.
- [Kru21] Stephan Krusche. *Interactive learning - A Scalable and Adaptive Learning Approach for Large Courses*. 2021.
- [Ope23] OpenAI. *Gpt-4 technical report*. 2023.
- [PAKC06] John Sweller Paul A. Kirschner and Richard E. Clark. *Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching*. Routledge, 2006.
- [SK20] Lara Marie Reimer Bernd Bruegge Stephan Krusche, Nadine von Frankenberg. *An Interactive Learning Method to Engage Students in Modeling*. 2020.