SeAMK
SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Aymane El bardaoui

AI in education: Implementation of an LLM AI chatbot

Thesis
Spring 2025
Bachelor of Engineering, Automation Engineering

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Degree Program: Bachelor of Engineering, Automation Engineering

Specialization: Machine Automation

Author: Aymane El bardaoui

Title of thesis: AI in Education: Implementation of an LLM AI chatbot

Supervisor: Raine Kauppinen

Year: 2025 Number of pages: 47 Number of appendices: 0

This thesis delves into the implementation of a Large Language Model (LLM) based AI Teaching Assistant (AI-TA) tailored for particular courses. The goal is to improve the learning experience of students. The AI-TA was made using a retrieval-augmented generation architecture: the courses materials were embedded into a vector database to focus the LLMs ’responses, and a minimal User Interface was created to enable students to have easy and simple interaction with the AI-TA. The methodology followed a constructive design and evaluation approach; the AI-TA’s performance was then tested against generic AI models that are not specialized in the courses’ content.

The customized AI-TA answered correctly approximately 95% of the questions related to the courses fed to it, with only minor inaccuracies that can be noticeable in particular complex queries. The average response time was approximately 2 seconds, which meets real-time interaction requirements. In comparison with general models such as ChatGPT and Google’Gemini. The AI-TA was the best at determining answers that were closely aligned with the course material. This shows improved accuracy and relevance in subject-focused queries, indicating that the integration of courses-specific knowledge via embeddings can improve the precision of an LLM’s output in an educational environment.

The outcome suggests that combining LLMs with specific courses is an effective approach to creating AI teaching assistants. Without the need for additional tuning, the AI-TA was able to deliver accurate responses and context-aware answers. This project underlines the potential of AI-TAs in education by providing immediate aid or support to students. Considerations for ethical concerns such as data privacy and bias mitigation are discussed to ensure responsible integration of AI in learning environments.

[1] Keywords: AI in Education, Large Language Model; Teaching Assistant; Chatbot; Personalized Learning

TABLE OF CONTENTS

Pictures, Figures and Tables

Terms and Abbreviations

AI	Artificial Intelligence
AI-TA	AI Teaching Assistant. It is a term used to refer to AI systems that provide educational and teaching aid using AI.
LLM	Large Language Model. It is a type of AI model that makes use of the transformer architecture to generate textual information in a coherent manner.
ITS	Intelligent Tutoring Systems. ITS refers to computer systems that imitate human tutors and provide instruction and feedback to students.
ChatGPT	Chat Generative Pre-Trained Transformer is AI language model made by OpenAI. It can understand and generate human-like text-based user input.
Gemini	Developed by Google deep mind, it is designed to generate text images audio and video enabling advanced reasoning and interaction between multiple data types.
API	Application Programming Interface, a set of rules that allows different software applications to communicate with each other.

1 INTRODUCTION

1.1 Background

With the advancements in the field of Artificial Intelligence, many practical implementations of AI have been introduced in a multitude of fields for automating and improving the productivity of workers and employees. Such AI tools have also been integrated into the educational realm and show promise in improving the learning experience. To provide an enhancement in the learning of the students, AI-powered teaching assistants (AI-TAs) are being developed to provide personalized learning. Current existing AI tools in the market have been trained on a generic corpus of information, which makes it so that their responses are, at times, not adequately tailored for the proper understanding of the student.

The focus of this thesis is to see whether custom-trained AI-TA materials pertaining to a specific course provide more accurate responses than generic AI models. The study aims to evaluate whether these custom AI-TAs can provide responses better aligned with the course content, and lead to a better understanding of the course content by the student.

1.2 Aim of the thesis

The primary aim of the thesis is to develop and evaluate an LLM-based AI system to be a Teaching Assistant and provide aid to students' queries regarding a specific course that the AI Teaching Assistant (AI-TA) has been trained on. Key objectives of the thesis are:

- Requirement Analysis to determine essential features for the AI-TA.
- Development of a custom LLM-based AI-TA trained on course material of a specific course.
- Evaluation of the performance of the AI-TA based on decided metrics such as accuracy of content and user satisfaction.

Through addressing these objectives, the study aims to aid in the progression of research into the development of AI-driven tools for learning by demonstrating the potential benefits of custom AI-TA tools and their viability against generic AI models.

1.3 Methods

The research employs the constructive research approach, which entails the development and testing of novel approaches through practical implementation (Lukka, 2003). This study's primary aim is the development of a customized Large Language Model (LLM)-based AI-TA for a specific course and evaluating its performance to see if it provides adequate and improved responses when compared to generic AI solutions.

The research process has been divided into four phases. The first phase focuses on the literature review of AI applications in the realm of education, examining its benefits and limitations. The second phase focuses on the requirement analysis to specify all the essential features to be included in the AI-TA and selecting a specific course for the model implementation. The primary focus of the third phase is on the designing, development of the model, and testing of the customized AI-TA, to ensure that it is consistent with the specified requirements. The final phase is concerned with evaluating the responses of the custom AI model and comparing its performance with generic AI models based on metrics such as accuracy and user satisfaction.

This research makes use of a methodological approach to the development of the AI-TA, which consists of the following phases:

- Phase 1: Literature Review – A detailed and in-depth analysis of the current implementations of the AI-TAs, their viability, uses, benefits, and potential drawbacks, along with the limitations of the generic AI model compared to customized AI-TAs.

- Phase 2: Requirements Analysis – Identification of the key components required for the development of the AI-TA, and defining the evaluation criteria to determine the performance of the custom model in comparison with the generic AI options.

- Phase 3: Design, Implementation, and Testing – This phase is concerned with the development and training of the AI-TA system on the selected course and performing tests to ensure the proper functionality of the system.

- Phase 4: Evaluation and Recommendations – The bespoke AI-TA system is evaluated by performing a comparison of the responses of the custom AI-TA with the generic AI models, and the performance is evaluated based on metrics such as the accuracy of the provided information.

Data for the study was sourced from SeAMK's internal course materials and open educational resources, ensuring that the AI model is trained on real-world academic content.

1.4 Structure of the Thesis

The thesis consists of multiple sections, detailing the theoretical and practical real-world use cases of AI in education and AI-based Teaching Assistants (AI-TAs). The theoretical section concentrates on the roles and implications of AI in the realm of education, its benefits and limitations, and the potential risks posed by AI. The practical section is mainly concerned with the development of the customized AI-TA for the specific course, and its testing to ensure proper functioning.

Chapter 2 provides an overview of AI in education, its role, applications, benefits, and challenges. Chapter 3 focuses on the requirement analysis for the AI-TAs, to specify the essential features, and detail the course selection and evaluation metrics. Chapter 4 is primarily concerned with the design, implementation, and testing of the testing of the pilot implementation of the AI-TA, detailing the methodologies used for building and refining the system. Chapter 5 presents the evaluation results, and a comparison of the customized AI-TA with the generic pre-existing models, based on metrics such as accuracy and student engagement. Chapter 6 is focused on ethical concerns like data privacy and biases introduced by AI in the context of education. Chapter 7 concludes by providing a summary of key findings and recommendations for future AI research pertaining to education.

2 AI in Education

2.1 Introduction to AI in Education

The adoption of AI has seen an unprecedented rise in the educational realm, which has had a major impact on how students and teachers interact with learning and teaching respectively (Su & Yang, 2023). The integration of AI into learning endeavors has led to the development of AI-powered Teaching Assistants (AI-TAs), which support students by providing individualized learning experiences, and support teachers by automating repetitive administrative tasks. AI-guided educational experiences, therefore, have the potential to revolutionize the quality of education obtained by students (Lee et al., 2023).

Integrating AI into the educational realm provides major benefits. AI can be used to automate many routine processes such as grading and attendance, and other such repetitive tasks (Torre-López et al., 2023). Additionally, Generative AI models provide real-time responses to queries of students regarding a wide range of subjects and even possess the ability to adapt their responses based on the student's current level of understanding regarding the concept. This makes such AI models an indispensable tool for grasping new concepts with a much more gradual learning curve. However, just like any other technology, the benefits of such technologies also invite certain risks, such as data privacy, and potential algorithmic biases.

2.2 Applications of AI in Education

AI has a multitude of use cases in education (see Figure 1), especially in the realm of learning optimization. The ability of LLMs to imitate and thoroughly understand human speech makes them ideal for simulating a tutor for providing an environment for students to learn concepts in a guided fashion (Lin et al., 2023). The ability of such models to understand context makes them very good at generating responses tailored to the student's current level of understanding (Jian, 2023, p.16). Based on the student's prior experience with learning material, the AI can recommend materials that would be most appropriate for aiding in the student's understanding. This can be particularly helpful in large classrooms where it may not be possible to provide personalized instruction and guidance to all students.

Intelligent tutoring systems (ITSs) can gauge the level of understanding of a student based on the questions asked, and, based on the level of proficiency the student has on the topic at hand, the AI provides the appropriate amount of detail. Through having an awareness of context, these ITS can infer the content that the student is having difficulty with and provide help regarding the concept that is deemed difficult.

A popular use case for AI in learning is in the form of chatbots, which can answer the queries of students regarding general information, provide guidance, and provide aid in understanding certain information in the course material (Tlili et al., 2023). Such AI systems are always available, which allows the students to have easy access to information, positively impacting the level of understanding regarding the course content.

Figure 1. AI Application in Education.

2.3 Advantages of AI in Education

A use case for AI-powered tools exists for all entities involved in the realm of education. AI Teaching Assistants (AI-TA) can play a significant role in aiding students in comprehending complex concepts in a multitude of fields, by providing them with customized explanations to their queries based on their current level of proficiency regarding the concepts at hand. Such systems would provide a higher level explanation to students not familiar with certain concepts and provide much more detailed examples and scenarios to students that are more proficient. This allows the students to gradually grasp the concepts strongly, positively aiding

in their understanding. Creating a bespoke gradual learning curve for each student makes such AI tools a very useful tool for aiding understanding of concepts and allows for a large-scale improvement in student comprehension.

Along with benefiting the students, AI tools can also be developed to automate repetitive tasks necessary to be performed by the teachers, so that the teachers can have more time engaging with the students. Such strong capabilities make it clear that AI has a significant potential to enhance the learning capabilities of students (Xie, 2023).

2.4 Testing AI in Education

When building an AI educational tool, It is very important to verify that it functions correctly without minimum flaws, and it meets the performance expectations of the user. Various types of testing have been performed, such as functional testing that evaluates if the AI can understand different types of students questions and provides correct answers based on the course materials. Additionally, the speed and performance were tested by measuring the time the AI-TA takes to reply and how much computer power it uses. Also, the AI was tested with off topic questions outside the course content to see if it provides a polite answer indicates that it cannot answer, which is better than providing incorrect or irrelevant information. This type of testing shows that AI-TA can handle the queries without misleading the student with incorrect information. These testing approaches ensure no matter what the condition is the AI-TA is always going to be accurate, efficient and reliable.

2.5 Ethical Considerations of AI in Education

While AI systems like the one tailored in this project offer significant benefits in education it is important to talk about their ethical concerns. It is crucial to address these issues to ensure that the implementation of an AI Teaching Assistant is done responsibly and does not harm students or the learning environment. We are going to discuss four major ethical aspects relevant to AI in education that are data privacy, bias, transparency, and fairness in grading. These issues have been highlighted by researchers as key challenges that must be managed as AI becomes more integrated into classrooms (Eden et al., 2024).

Each of these considerations is explained below in the context of the AI-TA system and similar tools used in education.

- Data Privacy and Security

The privacy of the students is a huge concern when talking about AI in education. AI teaching assistants can handle very sensitive data about the students, such as their identities, academic records, and performance metrics. In our case scenario, the system of implanting this AI-TA uses course materials and questions that might be asked by the students. It is important to protect such data, as it may reveal what the students are struggling with. If the AI-TA were to be integrated into the Moodle learning platform, it would likely log interactions (questions asked, answers given) – this data could be valuable for improving the system but must be safeguarded.

One of the risks is that the data collected by AI systems could be exposed and used for some unintended purposes. Educational institutions have legal and ethical obligations to uphold student privacy. For example, in the United States, FERPA (Family Educational Rights and Privacy Act) regulations are strict on how student records can be shared, and these kinds of laws exist globally (Future of Privacy Forum, 2024). All student conversations with the AI-TA should be considered part of the educational record and should be protected. This is by implementing robust encryption for data at rest in transit, along with controls so that only authorized people or students themselves can view their data.

Another aspect is letting users know about data usage. Schools and universities should be transparent about what kind of data AI such as AI-TA collects and whether that data is shared and used for further training of the AI or other purposes (Eden et al., 2024). In our case, we did not use any student data to train the model (since our focus was on course content). But if we suppose that the system was learning from student queries over time, this becomes a question of consent. Students should ideally have the option to opt out of having their questions stored or used for further improvements of the model.

Moreover, keeping the AI-TA's base on a secure school server is preferable to sending data to third-party cloud services. In our case, we used third-party LLM API, which means student questions that might include parts of assignments or personal data for example are sent to an external server (model provider). This could be a privacy weakness. In a production

environment, one might solve this by choosing an LLM deployment that can run locally at the institution or by making sure that the third-party service complies with strict data privacy agreements (for example: not storing or sharing the queries provided by the student). Techniques like anonymization or not sending any student-identifiable information in prompts can also help.

Data security is closely related to privacy. Any AI-TA could be exploited if it wasn't securely designed. For example, an attacker might try prompt injections to make the AI reveal sensitive information that it shouldn't. Therefore, measures like input validation and sandboxing the AI's capabilities so that it can not access anything beyond its scope should be put in place.
In conclusion, student data privacy and security are very important and it requires technical safeguards, policy decisions, and transparency with the users. Failing to do so can erase trust and even put students at risk, as their sensitive and personal data can be exposed (Eden et al., 2024). Responsible implementation of educational AI must treat privacy protection as a top priority.

- Bias and Inclusivity

AI systems are only as fair as the data and algorithms that we implement in them. Bias in AI can affect in many ways. In an educational context, this can have very serious implications for equity. A teaching assistant AI can, for example, favor certain examples that align with the majority culture represented in its training data, and alienate students from minority backgrounds. In the context of our AI-TA, course content provided by SeAMK and web links is the model's base knowledge if the AI data contains biases that reflect historical and social biases (which it doesn't) the AI's outputs could reflect or even amplify those biases (Barshay, 2024). For example, if asked to provide examples, a biased AI might consistently choose ones that feature male scientists (if the training data underrepresents female scientists) creating a stereotype.

The bias can also be seen in how the AI interprets student questions. It might misunderstand and then provide inappropriate responses to dialects or phrasing used by a group of students which leads to unequal help: some students might get more accurate responses than others just because of linguistic bias. So, it is very important to make sure that the AI is inclusive In understanding diverse student inputs.

Also, there is documented evidence of bias concerning AI-assisted grading (which is an area related to AI-TA). Recent research by ETS showed that an AI scored essays that have been written by Asian American students were lower on average, demonstrating a potential racial bias in automated grading (Barshay, 2024). While our AI-TA does have the feature of grading, this example illustrates the risk that AI behavior might accidentally disadvantage certain groups. If the AI was used in any evaluative capacity (even giving feedback on student answers), one must be aware that it doesn't favor a particular writing style or background that could correlate with a demographic.

To fix this, we have to start with the data. Our AI-TA's specialized knowledge came from neutral course materials. That is a plus, as it focuses the AI on what has been covered in the class. However, the LLM itself is a broad model. One mitigation is to keep the role of the AI supplementary it should not be the sole answer of the truth. If a student ever feels that the answer is not right (off or biased) they should double-check other sources or ask the teacher. In addition to that developers should test the AI-TA with multiple sets of questions and scenarios to see if any biased content is generated. In our test scenario, we did not face overt bias, perhaps because the domain was limited.

Another approach to that also includes bias detection and correction mechanisms. Researchers are exploring some techniques to identify biased responses and fix them (Ferrara, 2023). For example, if the AI-TA was built in a classroom with students who have different languages from each other we need to make sure that the AI responds well to all supported languages, not privileging English over others. Testing the AI for bias should be part of the deployment of any educational AI. In short, while AI provides personalized assistance, it must still be monitored to ensure that it gives answers equally to all students. Teachers can review AI responses periodically for fairness and correctness (Silvestrone & Rubman, 2024).

- Transparency and Explainability

In an educational context transparency means the ability to inspect how the AI works and the practice of making users aware of its nature and limitations. AI models often resemble a black box, they take an input and provide an output and it's not always clear how or why.

From the student’s point of view, it should be clear that the AI is an automated system, not a real human, and that the answers are generated based on certain sources. In our case, we could improve the transparency by indicating which slides from the pdf files or links the answer was generated from. This feature would greatly increase trust, the student can verify the answer against the course material. Many modern AI solutions prefer explainable AI, where the system provides justification or references for its outputs (Luckin & Cukurova, 2022). We do not have that feature however, integrating it to show the source text that the AI used to answer the question would align with this principle.

Then another aspect also is being transparent about the limitations of the AI-TA. It can occasionally make mistakes or not know the answer, and this is what every student must know. In our case scenario, the AI-TA sometimes had to say it did not know the answer or that information communicating this is better than the AI pretending to know the answer. The AI-TA should be a complement learning tool that does not replace lecture books or teachers. This manages expectations and encourages students to use it as a study aid rather than an authoritative source on its own.

For learners and administrators, transparency means they should know what data the AI has been trained on and how it processes the queries. This goes along with the matter of privacy which means that being open about how the system works can help in building trust among stakeholders (Eden et al., 2024). If the AI system gives a disputed answer by following some certain path (for example the AI said this because these notes suggest X and the question was formulated in Y way") that would be important for resolving this issue. Thus, the "black box" nature of AI is a known worry in evaluations which means it can be hard to override an AI decision if you don't know how it arrived there (Luckin & Cukurova, 2022). Hence, the valuable option is designing the AI-TA with some level of clarity (even as simple as showing the top retried chunks to the teacher.

One more side is transparency in terms of usage: making sure that the implementation is done ethically by informing the students when they ask a question like " Who else sees the questions I ask the AI ". We must inform the students and obtain their consent if needed but ideally, they deserve a transparent answer from no one beyond maybe the system maintainers, and even those only in aggregate.

In summary, transparency and explainability should be incorporated into educational AI which aids in building trust and allows the technology to be used in an accountable way. By providing sources of the answers, clarifying AI's identity, and explaining its processes. We can mention that AI might not be always right, and it might provide misleading information in the background. For our AI-TA, the next step could be to add the feature of displaying how the AI found the answer giving both students and teachers confidence in its responses.

- Fairness in Grading and Feedback

One of the roles of AI in education is grading and providing feedback. While in our case the AI-TA primarily answers questions, it won't do any harm to imagine it or any similar AI systems being used to grade quizzes or student assignments automatically or to give feedback on students' written answers. Making fairness is a critical ethical issue. Because when a human gives grades, for all their potential inconsistency, they can be at least held accountable and explain their judgment. But in the case of AI when it starts giving grades, how do we know and make sure these grades are fair, unbiased, and reflective of a student's true performance?

Studies show that AI grading systems can introduce unfairness. As mentioned before, if the training data for grading has biases, certain groups can be automatically underscored (Barshay, 2024). Another fairness problem is consistency: ironically, AI might be too consistent compared to human graders, not giving the benefit of partial credit in some cases a human teacher might recognize an " outside the box " correct answer. We saw in Chapter 2 that one benefit of AI is supposed to be consistent grading, but fairness is not just about that it's also about understanding the context (Silvestrone & Rubman, 2024). A fair AI system should accommodate diverse correct answers and be flexible when it comes to students' creativity, something AIs struggle with if they are not carefully designed.

If the AI was expected to give feedback on student answers, measures should be in place such as:

Rubric-based evaluation: This can align the AIs checking with human grading standards by ensuring that the AIs check against a teacher-provided rubric for example. Human oversight is also important AI should not be the sole and only decider, especially at a high grading level. Teachers could review AI-generated grades or feedback before they are finalized. This

"human in the loop" approach is good for catching any obvious mistakes the AI can potentially make (Silvestrone & Rubman, 2024).

Appeal mechanism: Students should have a way to make an appeal and ask for clarification on grades given by AI. If the student asks "Why?" the AI or teacher should be able to justify it. This takes us back to transparency; without it achieving fairness might be a long-term goal.

Another aspect of fairness is accessibility. The accessibility of the AI-TA to all students in the class must be considered. Schools must consider the divide, there are students with better internet access for instance, or devices that use AI more. Therefore, to keep things fair, schools may need to provide time in computer labs or integrate AI-TA usage into the course in a structured way so that everyone can benefit from it equally. It should provide aid in learning uniformly and not only for those who seek it out the most.

Finally, fairness goes beyond that to how the AI-TA itself is evaluated. We should continuously monitor its performance across different student demographics and the kinds of problems it faces. For example, the AI might perform better in math courses than in essay courses), in this case, we need to tweak the system or limit its usage to appropriate context.

In summary, AI can make grading very efficient when it is done correctly and it requires careful design and oversight. In the case of our AI-TA, we kept its role as an assistant rather than an evaluator, perhaps to avoid complexities of grading fairness as mentioned in this pilot. Should the project move towards grading, the principles mentioned above must guide its development. Ethical use of AI in grading means using it to support teachers, not using it to replace their judgment, while always keeping the student's interest and equity at the forefront.

2.6 Future of AI in Education

The future of AI looks very promising, and with the significant strides in Large Language Models (LLMs), it is apparent that the technology will continue to become more sophisticated and improve in quality therefore becoming even more exceptional at providing aid in the educational sector. As such, AI models become more capable, they will become more context-aware and hence be able to provide a much more personalized experience for learning.

As such technologies improve, a potential possibility is their inclusion into the classroom so that the LLMs can be used to provide aid to students in an engaging and personalized way.

3 Requirements Analysis for AI Teaching Assistants

3.1 Defining the Scope of an AI Teaching Assistant

The AI-powered Teaching Assistant (AI-TA) created for this project intends to provide an interactive and personalized learning experience to the students, by aiding in understanding concepts related to the course upon which the AI-TA is trained. The AI-TA is meant to provide answers to the queries of students with information that aligns with the information present in the chosen course material used for the training. The AI system will be an LLM that will provide answers to the student ensuring that the provided output aligns with the curriculum of the course.

The primary functions of the AI-TA are:

- Course-Specific Question Answering: The response to the queries of students regarding the course material will be structured and organized.

- Context Aware Assistance: The answers generated will align with the course material based on the student queries, and retain knowledge of the student's competence regarding the topic, providing a custom learning experience.

- Real-time Feedback: The AI-TA can provide real-time feedback and guidance based on student responses, assisting the students in solidifying their concepts.

Furthermore, the scope of AI-TA excludes grading or formal assessment. This helps avoid ethical risks ensuring that all evaluations are done by humans, which maintains fairness and accountability.

3.2 Course selection criteria

Certain aspects must be considered when deciding whether the AI-TA system would be effective for a specific course. To determine whether a specific course would be ideal for the AI-TA to be trained upon, the following criteria are to be considered:

- Textual Learning Material: Since the AI model is a Large Language Model, it performs well with textual information, therefore the most appropriate course would consist primarily of text. Due to this, courses dealing with other modes of information such as images would not be an appropriate choice.

- Conceptual Depth with Clear Answers: Courses consisting mainly of clear-cut, specific, and to-the-point answers to questions would be ideal for the AI-TA, as compared to courses that tend to be subjective and deal with open-ended questions.

- Structured Curriculum: The curriculum of the course should be clearly presented to allow for understandability when the AI-TA is trained on the data corpus pertaining to the curriculum.

3.3 Evaluation Metrics for AI Performance

The developed AI-TA system can be evaluated based on a set of predetermined metrics. Such metrics can be used to ascertain whether the AI-TA performs adequately. The metrics to be considered are as follows:

- Response Accuracy: The ability of the AI-TA to provide correct and relevant information to the query inputted by the student. This can be measured by comparing the response generated to the course-provided information regarding the query.

- Student Satisfaction: Direct feedback from students using the AI-TA. Surveys can be used to obtain feedback regarding the AI-TA, to determine the user experience with AI-TA.

By gauging the above-mentioned metrics, a good understanding of the viability and efficacy of the AI-TA system can be gauged.

4 Implementation of the AI Teaching Assistant

This chapter is about the development of the custom AI-TA. It describes all the technologies that have been used alongside with the tools, it also describes the preparation of the courses' data, the system architecture, and the design of the AI pilot user interface. The key components of this implementation to ensure the AI's responses remain focused on the course content include the embedding of the course materials into a vector database, integration with a Large Language Model (LLM) via API, and prompt engineering. Research shows that virtual machine assistance can provide personalized support, immediate feedback, and adaptive learning experiences (Audras et al., 2022). The following sections explain the process of implementation in detail.

4.1 System Architecture Overview

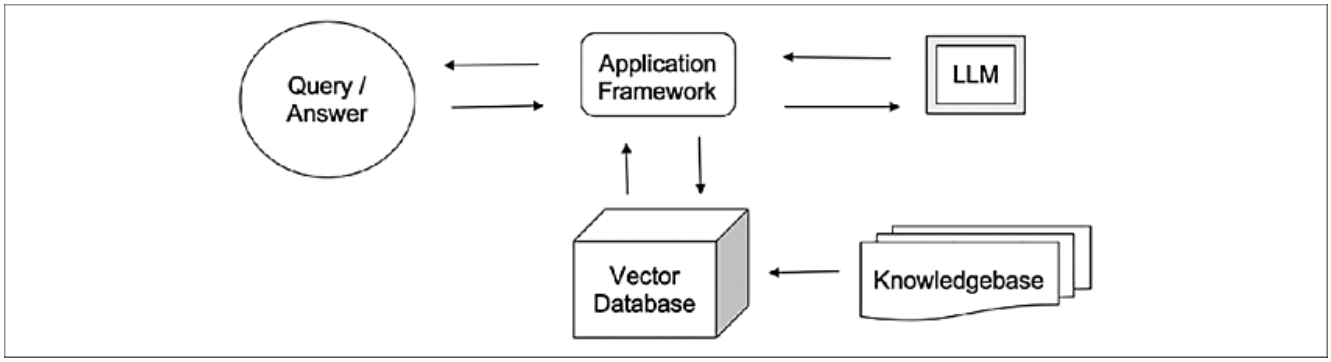

Figure 2. Retriever-augmented generation (RAG) architecture model. (Rahman & Hossain, 2024)

The system architecture follows a Retrieval-Augmented Generation (RAG) approach (see Figure 2), alongside integrating a vector-based knowledge retrieval system with an LLM to provide accurate responses. This consists of two primary phases:

1. Offline integration phase: A large corpus of course-related data (such as PDF slides from Moodle, and URL links.). is split into smaller chunks for better processing and high efficiency.
2. Online Query phase: When the student submits a question, it is first converted into embedding by using the same embedding model. Then the system performs a semantic search in the vector database and tries to retrieve the most relevant passages based on similarity. The retrieved passages are passed to the LLM as a context ensuring that the model's response stays focused on the provided knowledge.

Then finally, the LLM processes the retrieved information and provides a summarized answer, which is delivered to the student through a chat interface.

The key components of the Architecture include an embedding Model and vector database which converts course materials into numerical representations and saves them in an indexed knowledge base for fast retrieval.

Another key component is LLM summarization response generation: the previously retrieved information is used to finally produce well-structured and accurate answers that are related to each course.

User interface: It allows the students to ask questions and receive real-time answers. This architecture ensures accurate, focused responses which reduces issues like "LLM hallucinations" by grounding the answers in just the retrieved course materials provided. Superior performance has been demonstrated when using retrieval-augmented generation architecture in specialized areas (Feridooni et al., 2024).

4.2 Installing the Required Libraries

```
!pip install --no-cache-dir langchain langchain-huggingface sentence-transformers

!pip install --no-cache-dir langchain langchain-community unstructured pdfminer.six
```

Figure 3. Installation of Required Libraries.

Before making the chatbot, it is important to install the necessary libraries which make tasks involved in data retrieval, text processing, vector storage, and AI response more easy. The installation of libraries is done by using Python's package manager, ensuring that all required modules are available for use. The commands shown in Figure 3 ensure that the required packages are installed in the environment. The first command installs multiple Python packages: Langchain, Langchain-HuggingFace, and sentence- transformers. To prevent caching the packages during the installation process we are going to use --no-cache-dir which ensures that the late version of each package is successfully installed. Langchain is a framework that serves the purpose of developing applications with language models, which helps integrate natural language processing (NLP) tools. The Langchain-HuggingFace

package is just a specific integration of Langchain with Hugging Face, it allows the usage of Hugging Face pre-trained models in Langchain applications. Sentence-transformes is a package for working with sentence embeddings and doing tasks like semantic search, clustering and similarity matching all by encoding sentences into high dimensional vector representations. A similar combination of packages is often used in NLP and machine learning tasks that involve understanding, processing and comparing text data.

The second command installs different sets of Python packages: Langchain,Langchain-community,unstructured and pdfminer.six with again using the --no cache-dir to avoid caching. Langchain already have been explained above. The Langchain-community package is more of an extended version of the Langchain library, which includes more features and tools making it more flexible for multiple NLP and machine-learning projects. For processing and handling of unstructured data we use unstructured package which could be useful in extracting data from various sources like texts, images,pdfs or any documents. pdfminer.six is a library used for extracting text from PDF documents. Which is useful for working with PDF files in NLP applications. It is good to keep in mind that this set of tools is particularly useful for projects that involve working with large-scale unstructured data like PDFs and require text extraction and processing capabilities.

4.3 Generating Text Embeddings

Each chunk was converted into a high-dimensional vector representation using the HuggingFaceEmbeddings from the Langchain_HuggingFace library to start the semantic search of the text chunks (see Figure 4). The Pre-trained HuggingFace model has been used to generate embeddings for documents and student queries. With this model, we can produce a high dimensional vector for any given text input, and it captures its semantic meaning and makes efficient comparisons during the search (Reimers & Gurevych, 2019). This process involves both the embedding of the query by the student and the document text, which are represented in vector space for similarity comparison.

```python
from langchain_huggingface import HuggingFaceEmbeddings

# Initialize the embedding model
embeddings = HuggingFaceEmbeddings()

# Embed a query
vector = embeddings.embed_query("hi")

# Show first 5 vector values
vector[:5]
```

Figure 4. Generating Text Embeddings Using Hugging Face Models.

The process of all the text chunks have passed through the embedding model to produce set of embedding vectors was executed offline for building the knowledge base. Each vector was stored linking back to the original chunk of text. The system can perform efficient similarity comparisons between the student's question and the stored content, that's by representing the textual knowledge in vector form. The embeddings are computed quickly, as the HuggingFace Model is designed for understanding what the text means, which ensures fast retrieval of relevant content when questions are made. This approach is highly effective, and it eliminates the need for additional training, benefiting from the power of the pre-trained model for high-quality embeddings.

4.4 Data collection and preparation

A very important first step was collecting and preparing the course material for semantic search. The selected course content was collected from multiple sources, including lecture slides and PDF documents from Moodle (see Figure 6), and relevant web pages (see Figure 5). Python scripts were developed to load data from URLs and PDFs into text form, and HTTP requests were sent to the course-related web pages to extract the text content from there and irrelevant parts like navigation menus were ignored. For PDFs, a PDF parsing library was used to extract the text from each file of the pdfs. Collecting relevant and clean course data is crucial since the knowledge base content directly determines the quality of the assistant's answers (Feridooni et al., 2024).

```
import os
from langchain_community.document_loaders import PyPDFLoader, UnstructuredURLLoader, UnstructuredFileLoader

# Load data from URLs
urls = [
    # Data Structures and Algorithms Links
    "https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/arrays",
    "https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/collections",
    "https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1?view=net-7.0",
    "https://learn.microsoft.com/en-us/dotnet/api/system.diagnostics.stopwatch?view=net-7.0",
    "https://learn.microsoft.com/fi-fi/troubleshoot/developer/visualstudio/csharp/language-compilers/use-icomparable-icomparer",
    "https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic.queue-1?view=net-7.0",
    "https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic.dictionary-2?view=net-9.0",
    "https://learn.microsoft.com/en-us/dotnet/api/system.collections.stack?view=net-7.0",

    # Basic of Programming 1 Links
    "https://programming-22.mooc.fi/part-1/1-getting-started",
    "https://programming-22.mooc.fi/part-1/2-information-from-the-user",
    "https://programming-22.mooc.fi/part-1/3-more-about-variables",
    "https://programming-22.mooc.fi/part-1/4-arithmetic-operations",
    "https://programming-22.mooc.fi/part-1/5-conditional-statements",
    "https://programming-22.mooc.fi/part-2/3-combining-conditions",
    "https://programming-22.mooc.fi/part-2/4-simple-loops",
    "https://programming-22.mooc.fi/part-3/1-loops-with-conditions",
    "https://programming-22.mooc.fi/part-3/4-defining-functions",
    "https://programming-22.mooc.fi/part-2/1-programming-terminology"
]

url_loader = UnstructuredURLLoader(urls=urls)
url_data = url_loader.load()
```

Figure 5. Loading and Extracting Text from Online URLs.

```
# Load data from PDFs


# Define folder path containing PDFs and Word files
folder_path = "/content/sample_data/"

# List all files in the folder
files = os.listdir(folder_path)

# Filter for PDFs and Word documents
pdf_files = [os.path.join(folder_path, file) for file in files if file.endswith(".pdf")]
word_files = [os.path.join(folder_path, file) for file in files if file.endswith(".docx")]

pdf_data = []
for pdf in pdf_files:
    loader = PyPDFLoader(pdf)
    pdf_data.extend(loader.load())



# Combine all extracted data
data = url_data + pdf_data
```

Figure 6. Loading and Extracting Text from PDFs.

Once all the course texts were gathered, the data was cleaned. Issues like newline artifacts or extraneous whitespace have been removed for consistency. The combined corpus of data was split into manageable chunks to be handled (see Figure 7). The implementation used a repetitive character text splitter from LangChain (as shown in the provided code), which divides the text into overlapping parts of a few hundred words each, ensuring that each part represents a piece of information (like one paragraph or subsection) This preprocessing step allows the system to provide the most relevant chunk without processing the whole document.

In total, the course's data resulted in a collection of hundreds of text chunks. The preprocessing also included storing the source of each chunk which can be called metadata to maintain context. By leveraging HuggingFace embeddings for semantic search, each text chunk was converted into a high-dimensional vector representation, enabling efficient and accurate search operations

```
from langchain.text_splitter import RecursiveCharacterTextSplitter

# split data
text_splitter = RecursiveCharacterTextSplitter(chunk_size=1000)
docs = text_splitter.split_documents(data)


print("Total number of documents: ",len(docs))
```

Figure 7. Splitting Large Documents into Meaningful Chunks.

4.5 Vector database setup

The next step was to index them in a vector database. The system uses Chroma DB which is an open source vector store (see Figure 8). Chroma was chosen for being simple and it has a local deployment capability that allowed the project to run without needing an external database service. The purpose of Chroma vector database is to store the embedding vectors along with their corresponding text chunks together.

Each embedding vector (representing a chunk of course content) was inserted into Chroma database. Chroma automatically takes care of organizing these vectors in a way that enables efficient similarity search. In practice, this means the student gives a new query vector, and the database can quickly find the nearest similar vectors.

```
!pip install --no-cache-dir langchain langchain-chroma chromadb
```

Figure 8. Vector Database (ChromaDB) Installation

Once the indexing is finished, the system has a knowledge base of the courses: effectively a semantic index of all important points in the course material. With this, the offline phase of building the AI-TA's specialized knowledge is done. The performance of the vector database was tested after that to ensure that relevant chunks were retrieved. Notably, the retrieval is configured to return the top k results (in this implementation, k = 5 chunks) that are most like the query (see Figure 9). Using multiple chunks, the AI assistant can draw several pieces of information to formulate its answer this is in case the answer is spread across different parts of the course material.

```
from langchain_chroma import Chroma
vectorstore = Chroma.from_documents(documents=docs, embedding = HuggingFaceEmbeddings())
```

```
retriever = vectorstore.as_retriever(search_type="similarity", search_kwargs={"k": 5})
```

Figure 9. Store Embeddings in Chroma Database

Therefore, the use of vector databases in this architecture is good for best practices for knowledge-intensive question-answering systems. Allowing the system to augment the LLM with non-parametric memory (Lewis et al., 2020) in their work about retrieval-augmented generation (RAG). Basically, instead of just relying on the LLM alone, the AI-TA can look up the information from the course via the vector store, improving accuracy and transparency.

4.6 LLM Integration and Model Deployment

```
!pip install groq
```

```
from groq import Groq
from langchain.prompts import PromptTemplate
from langchain_core.output_parsers import StrOutputParser
from langchain_core.runnables import RunnablePassthrough

# Initialize the Groq
client = Groq(
    api_key="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"
)

# LLM function
def llm( question):
    """Simulate the HuggingFacePipeline using Groq API."""
    prompt = f"Question: {question}"
    response = client.chat.completions.create(
        messages=[
            {"role": "user", "content": prompt}
        ],
        model="mistral-saba-24b",
    )

    return response.choices[0].message.content
```

Figure 10. Install and Setup Groq (LLM) and define LLM function.

Now with the course data indexted, the next major component is the LLM that will generate the answers. A cloud-based Large language model was implemented using an API. The project was made using Groq platform’s LLM API, this last provides access to advanced

open-source language models with low reaction time. With Groq API I was able to develop a powerful model without needing to host it locally it provides an endpoint where queries can be sent and answers returned similar to GPT-4 API.

For setting up the LLM, the implementation got API credentials first and installed the necessary python libraries (for example langchain-huggingface integration for compatibility with Hugging Face model) (see Figure 10). This model used via Groq is the same in style as GPT transformer model. It is capable of understanding and generating detailed answers. Once it has been configured, a test query was sent to the API (for example question about the course) to confirm that the model's response is correct.

The integration is simply functioning in a way that a prompt containing the student s question from the retrieved context (as described in the next section) is made and returns the LLMs completion. With this function, the rest of the system can call simply get answer prompt and receive the model's answer by abstracting the details of the API call. The use of an API means the model itself is deployed on remote servers optimized for inference, which ensures that responses are fast and returned in a couple of seconds on average. This meets the requirement for a teaching assistant pilot. Because in a more advanced offline setting, one could replace the API with a locally fine-tuned model, but that would require significant computing resources. On the other hand, by using Groq’s cloud model, the project achieved strong performance.

It is important to note that the chosen LLM was not additionally fine-tuned on the specific course data, instead, the retrieval mechanism is relied upon to supply course-specific facts. The LLM used is a general-purpose model with broad knowledge, which the system constrains by feeding it relevant parts from the course. This approach was simpler than fine-tuning and provided accurate answers to course questions.

4.7 Prompt Engineering for Contextual Q&A

```
prompt = PromptTemplate(
    input_variables=["context", "question"],
    template=prompt_template,
)


llm_chain = prompt | llm | StrOutputParser()


rag_chain = {"context": retriever, "question": RunnablePassthrough()} | llm_chain
```

Figure 11. Define Prompt Template and how the chatbot should respond.

To make sure that the LLM provides answers that are relevant to the course and correct, a carefully designed prompt template was employed (see Figure 11). Instead of directly asking the LLM the raw user question, the system constructs a structured prompt that contains context from the course material. The prompt template was structured as follows:

- Context introduction: the model is given brief context in order to guide it to use the course content. An example of such prompts is: “Use the following course content to answer the question. If the content does not contain the answer, respond that the information is not available.”

- Injected Context: The top retrieved chunks of text from the vector database are inserted here. They are prefaced by a label like “Course content:” followed by the text of the chunks. If multiple chunks are included, they are separated by points or delineators.

- Question: the user (the student) asks a question this last is appended at the end, for example: "Question: {user’s query}.”

- Answer: Optionally, a prompt like “Answer:” is added to signal the model to begin generating the answer.

This structured approach ensures that the AI-TA is concentrated in a factual context that is drawn from the course materials, significantly improving the accuracy of the response. By using extracts that are authoritative course material, the model is more likely to be able to generate answers that are correct and directly relevant.

The prompt also included instructions to maintain a helpful, academic tone in line with how a human teaching assistant might respond. This approach was designed to ensure the style of answers remained appropriate for an educational setting, friendly, yes, but formal and precise.

4.8 User Interface Design and Integration

The user interface was the last component of the implementation (see Figure 12), it allows the end user (the student) to interact with the AI-TA simply and easily. A simple minimal chat-style web interface was developed to meet this requirement. This interface was implemented using Python web framework as a lightweight web application (using Gradio library).

The UI consists of a text input box where the student can ask his/her question, a submit button, and an area that displays the conversation. The back-end receives the question, runs through the process mentioned before of embedding, retrieval, and LLM prompt construction then obtains the answer (see Figure 13).

This last is sent after that back and displayed in the chat interface as the AI-TA response. The conversation display allows the user to ask follow-up questions in the same context, though each query is handled independently (the system does not yet maintain long conversation memory).

Special care was taken to ensure that the UI is easy to use and mirrors a typical chat with a human teaching assistant. The AI's responses are displayed clearly, perhaps with a markdown for any lists or emphasis if present in the answer. The interface was kept uncluttered, focusing on the Q&A interaction. This design choice aligns with findings that a friendly user interface is important for keeping student engagement with educational chatbots (Tsivitanidou & Ioannou, 2021). This design aims at making the AI-TA accessible and easy to use for all users.

```
!pip install gradio
```

```
import gradio as gr

def get_response(query):
    ans_each = rag_chain.invoke(query)
    return ans_each

iface = gr.Interface(
    fn=get_response,
    inputs=gr.Textbox(placeholder="Ask something..."),
    outputs="text",
    title=" Chatbot",
    description="Enter your query and get a response from the from your data.")

iface.launch()
```

Figure 12. Developing Chatbot UI

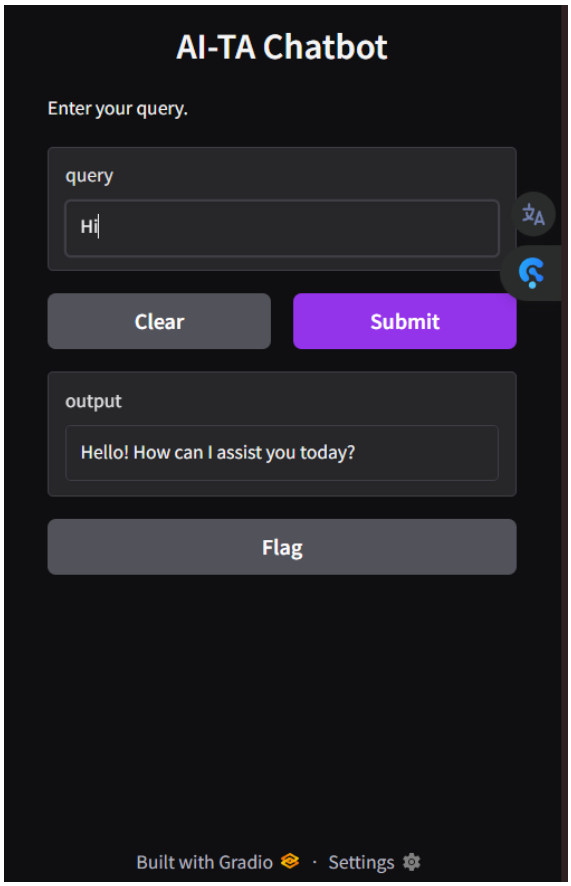

Figure 13. The AI-TA Chatbot UI

5 Testing and Evaluation

After building the AI-TA pilot, it goes without saying that the next step would be testing therefore, a series of tests were conducted to evaluate its performance and reliability. The evaluation covered the functionality of the AI-TA, speed, and comparisons with existing general AI chatbots. The goal of testing was to verify that the AI TA meets the requirements defined in earlier chapters namely that it answers students' questions accurately and efficiently and that it provides an improvement over generic AI models in course-specific contexts. This chapter outlines the testing procedures and presents the results using key metrics such as accuracy of responses and average response time. A comparative analysis with OpenAI'ChatGPT, Google'Gemini is also mentioned to contextualize the AI-TA's performance. Google's Gemini is also provided to contextualize the AI-TA's performance.

5.1 Testing Methodology

The first tests were about whether the AI-TA could correctly understand and answer questions about the course material (see Figure 14 and 15). A set of questions was prepared that covers various topics from the course's data (including both straightforward factual queries and slightly more complex conceptual questions). Each question was posed to the AI-TA through the user inter-face. Then the answers generated by the pilot were checked against the course materials for correctness and completeness.

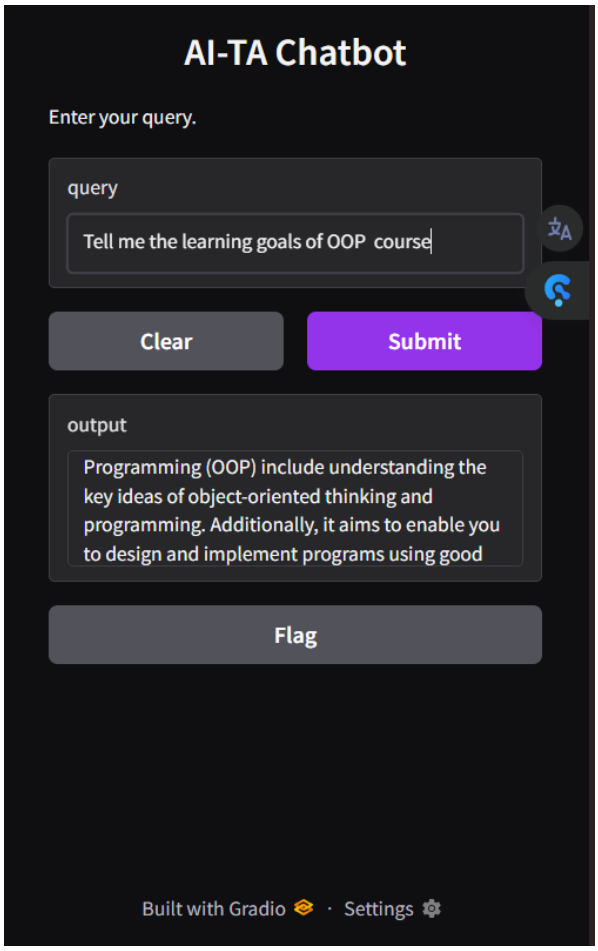

Figure 14. Functional testing.

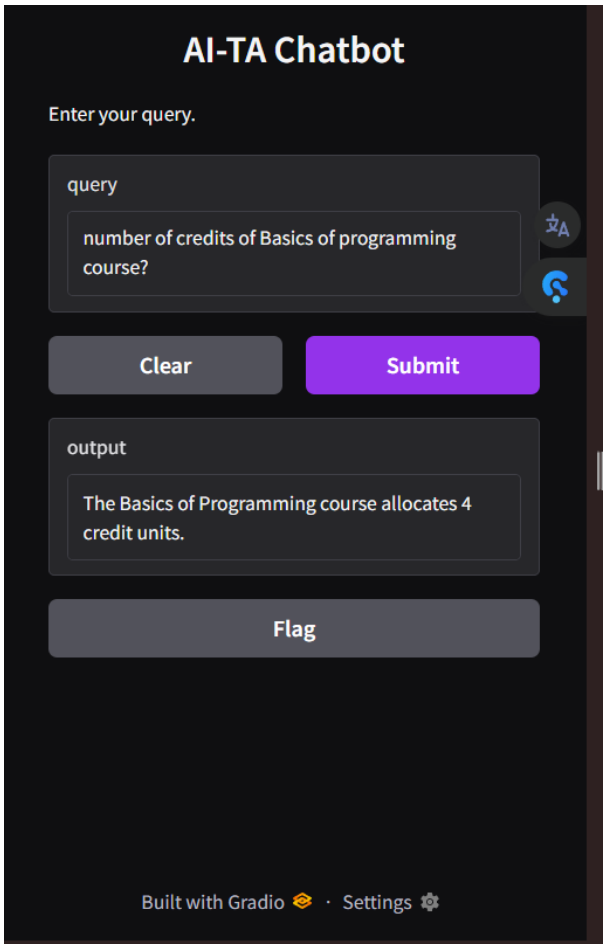

Figure 15. More functional testing

Next, the speed was evaluated by measuring the response time for each query from the moment of submitting the question to the time the answer was displayed (see Figure 16). In an interactive education chatbot, latency is very important; students should receive answers promptly to maintain a natural flow of learning. The AI-TA was timed over many queries of different lengths and complexity. Additionally, system usage (CPU/memory) was monitored to make sure that the solution was efficient. The performance testing process also involved deliberately sending longer, more complex questions to see if the processing time increased significantly or if the system struggled. Acceptable performance was determined when typical questions got average responses in only a few seconds.

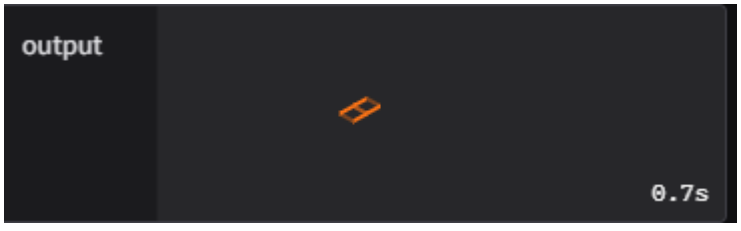

Figure 16. Performance Testing.

The AI-TA has also been tested with questions unrelated to the course, that were outside of the scope or were phrased vaguely (see Figure 17). This, without doubt, included queries that the AI-TA was not trained on (for example, a question from a different subject) and incomplete or oddly worded questions from the course domain. The purpose of this testing was to see how the system handles questions when it does not have clear answers in its knowledge base. Naturally, the AI-TA should respond with something like (“I’m sorry, I don’t have information on that topic.”) rather than giving an incorrect answer or a misleading one. Testing these unusual queries helps to observe whether the AI-TA will refrain from guessing beyond the provided data content. It was observed, for instance, that when asked a question unrelated to the course (like an off-topic personal question), the AI-TA politely stated it was outside its scope, which is a desirable behavior.

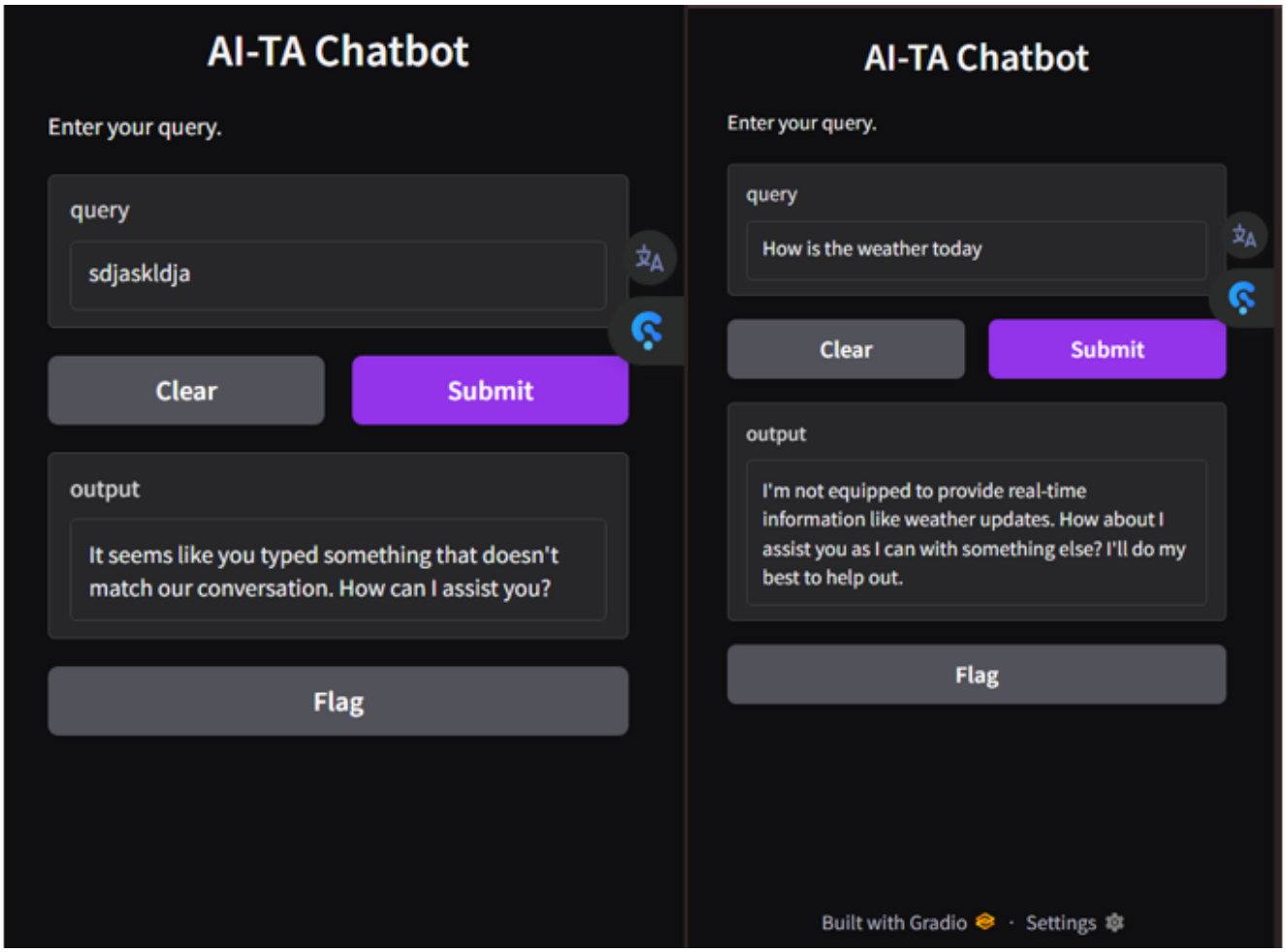

Figure 17. Edge Cases Testing.

5.2 Results and Performance Analysis

In terms of accuracy, the AI-TA's performance results were very promising. The system answered an estimated 95% of the questions related to the course correctly. Out of all the queries posed that had answers in the course materials, most of AI-TA's answers were factually correct and they aligned with the reference materials. Only a small percentage (around 5%) of responses had minor errors. These errors typically occurred in responses to particularly complex questions that required synthesizing multiple parts of the course content. Even then, the AI-TA's answers were often on the right track but could occasionally miss some details.

Regarding the response speed of the AI-TA, it has met the design goals. The average time of the answers was approximately 2 seconds or less for most questions. Simple questions were answered in under 2 seconds, while longer ones were about 3 seconds (for very detailed questions). This is an acceptable level of performance for an interactive assistant; the system does not keep the user waiting unnecessarily and it feels more responsive. The performance profiling showed that the embedding lookup in the vector database is very fast (millisecond level), and the majority of the 1-3 seconds response time comes from the LLM API processing. Nonetheless, this is near real-time and did not degrade even when questions were longer. The system did not show a significant slow-down for longer or more complex queries, which indicates that the chosen LLM and infrastructure can handle the prompt no matter what size they are. For instance, a question that included a long student query was still answered in roughly 2 seconds, similar to shorter questions, demonstrating consistent performance.

5.3 Comparison with Generic AI Models

To fully understand the advantages of the custom AI-TA, a comparison between it and general AI models has been done (ChatGPT by OpenAI and Gemini by Google). These are the benchmarks used because ChatGPT is well-known as a general AI assistant, and Gemini is a similar model introduced by Google. Neither of these models was given any specific training on the course material (interaction with them was in their default state). Table 1 summarizes the comparison from several key dimensions.

Table 1. Comparison of the AI-TA with general AI models (ChatGPT and Gemini) on key performance metrics and features.

Feature	AI Teaching Assistant	ChatGPT	Gemini
Accuracy	✅ High	✅ High	✅ High
Response Time	⚡ Fast (<2s)	⚡ Fast	⚡ Fast
Subject -Focused	🎯 Yes (Focused)	❌ No (Generic)	❌ No (Generic)
Uses Predefined Data?	✅ Yes	❌ No	❌ No
Handle Unusual Queries?	✅ Yes	✅ Yes	✅ Yes

As we notice, in terms of accuracy all three systems were rated as high for general queries. Chat GPT and Gemini naturally have been trained on massive datasets, they have strong overall knowledge and thus can answer correctly many questions. However, the custom AI-TA has an edge for the specific domain of the course. During testing, we noticed that ChatGPT could answer some course-related questions correctly, but it often gives answers that were not exactly aligned with the course materials or used from different methodology than the instructor. The AI-TA, on the other hand, provides answers that match with course's expected answers (because it draws directly from the course materials). Gemini showed a similar pattern to ChatGPT because it is also a general model. Thus, for academic content, the AI-TAs training makes it the most accurate and reliable in context.

By looking at the speed of the answers, all models are fast. The AI-TA has about 2 second average comparable to ChatGPT's performance via its API (ChatGPT responds in 1-3s for short prompts) and there is no major difference between it and Gemini; Any AI of these can deliver an answer in a few seconds, which is suitable for live interaction. This is very important that the custom AI stays as fast as the highly optimized popular models.

The biggest difference is in the subject focus part. The AI-TA is course-specific. The AI-TA was practically engineered to be an expert on the specific course content, and it uses a fixed number of documents from that course. While ChatGPT and Gemini are generalists. Unless

they have been manually provided with context in each prompt, they do not have a mechanism to limit their knowledge to particular subjects. In summary, the table reflects that the AI-TA is less likely to introduce outside information that the student has in their Moodle page from the course contents which reduces potential confusion. General AI might sometimes give correct answers that are not covered by the instructor/teacher, which could confuse students. The AI-TA avoids that by sticking to the course materials.

The use of predefined data is another key difference. The AI-TA also uses a predefined set of course data (it was fed the materials from the course, pdf files, and links.) ChatGPT and Gemini don't have that specific data unless it has been provided in their training also (which is for specific course notes are very unlikely). They mainly rely on their general knowledge. This highlights why the AI-TA can be seen as a tailored tutor, supposing the student asking a question will know exactly the references the student expects the answer to be from which the other AIs lack by default.

When it comes to handling unusual questions or off-topic ones, the three systems are capable to an extent. If asked a general knowledge question or something from a different subject the AI-TA will still produce an answer using LLM (though its vector database won't have relevant information for the off-topic queries) in our case, the test shows that AI-TA handled such queries gracefully, often stating that the information being looked for is not covered. ChatGPT and Gemini, built as general AIs, naturally handle a wide range of queries as well. Thus, all are marked “Yes” in the table for unusual queries, though we note that AI-TA is optimized for its domain and might explicitly indicate when a query is outside its domain, which can be seen as a feature (preventing it from giving possibly unreliable answers on topics it wasn’t intended for).

In conclusion, this comparative analysis shows that AI-TA performed well with top AI chatbots in general capabilities like speed and basic accuracy while offering alignment to specific course content. The exchange is that ChatGPT and Gemini have vast knowledge that could be useful if the student asks something relevant and tangential whereas the AI-TA has depth in one curriculum. This comparison concludes that a tailored AI teaching assistant can provide more relevant and course-aligned help than a one-size-fits-all model, without sacrificing the quality of AI understanding and generation. In educational settings, that specialization is often critical.

5.4 Ethical considerations in Pilot implementation

The pilot addressed four important ethical aspects. Privacy: Training the AI-TA relied on course materials alone, but query processing was via a third-party LLM which exposes student questions to external servers. We can avoid this by self-hosting. Bias: relying on official course materials, helped reduce clear bias, but regular reviews are still necessary to ensure the AI gives fair answers to different types of students. Transparency: The AI-TA didn’t include references to its sources, so adding citations (like slide or page numbers) would help users check the accuracy of its answers. Still, it was honest when it didn’t know something which builds trust. Fairness: The AI-TA was used only for tutoring, and grading was left to teachers. If grading is added later, it should include clear rubrics, teacher supervision, and a way for students to appeal. The project took a careful approach, showing the benefits of AI in learning while keeping ethical risks low.

6 Conclusions and Recommendations

6.1 Summary of Findings

The role of this thesis was to explore the role of AI in education through the development of a custom Large Language Model-based Teaching Assistant pilot for specific courses. The project included requirement analysis, implementation of the AI-TA system, evaluation of its performance, and reflection on the ethical side. The results of this implementation were encouraging. We justified that an LLM, when fed with course-specific material via embeddings and vector database, can effectively answer student questions with a high degree of accuracy and relevance. The AI-TA provided approximately 95% correct answers in testing, indicating that focusing the model on specific content successfully improves the alignment and usefulness compared to general models.

When it came to performance, the system achieved a rather fast real-time interaction speed, and answers were generated in just a couple of seconds. This responsiveness is critical for keeping a seamless learning experience and showcases that even complex AIs (with retrieval and external API calls) can be optimized for practical use. This addresses the thesis' aim of providing helpful, immediate support to students.

Comparative analysis with popular AI chatbots highlighted a key point: while ChatGPT and Gemini are powerful, a tailored AI-TA holds an advantage in an educational setting.

Specifically, by focusing on provided courses, the AI-TA was able to deliver answers that were better aligned with what students expected to know in those courses. This confirms that context specificity has an advantage in educational AI applications. While the generic AI models remain invaluable when it comes to open-domain queries, for targeted learning, a specialized AI assistant is more effective and trustworthy.

The project also covers ethical concerns of AI in education, underlining that thoughtful policy and safeguards must accompany any technological innovation in this domain. Data privacy must be protected; bias in AI answers must be identified and mitigated; the operations of the AI should be transparent to users; and if AI were to do evaluation tasks, it would have to be done fairly. These considerations are not afterthoughts, they are all integral parts and play a very important role when deploying an AI teaching assistant in the real world.

The AI-TA developed in this project was a controlled prototype, further development requires close attention to these factors to ensure it beneficially serves both students and teachers.

In conclusion, the research showed that AI, and LLM-based models in particular, have a significant role to play in the future of education. A well-and carefully designed AI Teaching Assistant can improve personalized learning by providing instant support to the students and can relieve instructors of repetitive work and potentially can also improve students' outcomes by reinforcing understanding. However, this relies on combining technical excellence with ethical responsibility when designing. This thesis has provided the basis by demonstrating technical feasibility and discussing key considerations for implementation in an academic context.

6.2 Recommendations and Future Work

Through the experience gained from implementing this pilot, multiple recommendations can be made for further improvements and wider deployment of an AI-TA:

First, improving the knowledge base by adding more courses in the future would allow students to gain knowledge about different aspects that help them solve their challenges. Even though the retrieval-augmented worked well, exploring a custom fine-tuned model could be valuable. Fine tuning the LLM on specific courses (if dataset can be built) might improve the fluency and accuracy of answers. Also, a rich user interface could be developed to include features for students to see references or ask follow-up questions in context, such as keeping a conversational thread. Currently, each question is independent and adding context memory would make it more like a human tutor. For real world use, the AI-TA should be integrated into existing real-world platforms such as Moodle. For example, this integration can log usage data in a privacy-compliant way for each teacher to see which topic has more generated questions, and to highlight areas of the course that are challenging. Any deployment of an AI-TA needs to be accompanied by clear policies. Students and faculty should be educated about when and how to use the AI-TA. For example, policies might prohibit using it during classroom exams or assignments where independent work is required. As usage grows, we need to deploy a vector database and LLM on cloud infrastructure that can handle many queries at once (in a class of 100 students, for example). Additionally, research and evaluation are crucial. We must know the effect of the AI-TA on the outcome of

students. Does having access to an AI-TA improve quiz scores, assignment quality, and exam performance? Does it increase student engagement with materials? These are questions that future work can aim to answer though controlled studies.

We must know the effect of the AI-TA on the outcome of students. Future research should investigate the impact of access to an AI-TA on quiz scores, assignment quality, and exam performance, as well as its influence on student engagement with course materials. These are aspects that future work can aim to investigate through controlled studies.

In closing, this thesis demonstrates a step forward toward the future of classrooms from the implementation of an AI teaching assistant. Educational institutions can responsibly scale up the use of AI, benefit from it more, and provide support, immediate feedback to students, and enhance their learning by following the recommendations stated above. The goal is a harmonious collaboration between human educators and AI assistants, complementing each other to provide students with a rich, supportive learning environment. The work presented here provides a roadmap and a proof-of-concept toward that vision, laying the groundwork for subsequent innovations in the role of AI in education.

BIBLIOGRAPHY

Audras, D., Zhao, A., Isgar, C., & Tang, Y. (2022). Virtual teaching assistants: A survey of a novel teaching technology. *International Journal of Chinese Education, 11*(2). https://doi.org/10.1177/2212585X221121674

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Agarwal, S. (2020). Language models are few-shot learners*. arXiv preprint arXiv:2005.14165.* https://arxiv.org/pdf/2005.14165.pdf

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). *Language models are few-shot learners. arXiv preprint arXiv:2005.14165.* https://arxiv.org/pdf/2005.14165.pdf

de la Torre-López, J., Ramírez, A., & Romero, J. R. (2023). Artificial intelligence to automate the systematic review of scientific literature. *Computing, 105*(10), 2171-2194. https://link.springer.com/content/pdf/10.1007/s00607-023-01181-x.pdf

Eden, C. A., Chisom, O. N., & Adeniyi, I. S. (2024). Integrating AI in education: Opportunities, challenges, and ethical considerations. *Magna Scientia Advanced Research and Reviews, 10*(2), 6–13. https://text2fa.ir/wp-content/uploads/Text2fa.ir-Integrating-AI-in-education-Opportunities-challenges-and.pdf

Gomes, Z., Dueck, A., Wheatcroft, M., & Szalay, D. (2024). Development of a vascular surgery-specific artificial intelligence chat interface using retrieval-augmented generation: VASC.AI, a specialized vascular surgery chatbot. *JVS–Vascular Insights, 2*, 100137. https://doi.org/10.1016/j.jvsvi.2024.100137

Harry, A. (2023). Role of AI in Education. *Interdisciplinary Journal & Humanity (INJURITY), 2*(3). https://radensa.ru/wp-content/uploads/2024/05/Role_of_AI_in_Education.pdf

Jian, M. J. K. O. (2023). Personalized learning through AI. *Advances in Engineering Innovation, 5*, 16–19. https://doi.org/10.54254/2977-3903/5/2023039

Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data, 7*(3), 535-547. https://doi.org/10.1109/TBDATA.2019.2921572

Lee, C., Myung, J., Han, J., Jin, J., & Oh, A. (2023). Learning from teaching assistants to program with subgoals: Exploring the potential for AI teaching assistants. *arXiv preprint arXiv:2309.10419.* https://arxiv.org/pdf/2309.10419

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *arXiv preprint arXiv:2005.11401*. https://arxiv.org/pdf/2005.11401.pdf

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *arXiv preprint arXiv:2005.11401.* https://arxiv.org/pdf/2005.11401.pdf

Lin, Z. (2023). Why and how to embrace AI such as ChatGPT in your academic life. *Royal Society Open Science, 10*(8), 230658. https://royalsocietypublishing.org/doi/pdf/10.1098/rsos.230658

Lukka, K. (2003). The constructive research approach. In L. Ojala & O.-P. Hilmola (Eds.), *Case study research in logistics* (pp. 83–101). Publications of the Turku School of Economics and Business Administration, Series B 1/2003.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *arXiv preprint arXiv:1908*.10084. https://arxiv.org/pdf/1908.10084.pdf

Su, J., & Yang, W. (2023). Unlocking the power of ChatGPT: A framework for applying generative AI in education. *ECNU Review of Education, 6*(3), 355-366. https://doi.org/10.1177/20965311231168423

Tlili, A., Shehata, B., Adarkwah, M. A., Bozkurt, A., Hickey, D. T., Huang, R., & Agyemang, B. (2023). What if the devil is my guardian angel: ChatGPT as a case study of using chatbots in education. *Smart Learning Environments, 10*(1), 15. https://link.springer.com/content/pdf/10.1186/s40561-023-00237-x.pdf

Tsivitanidou, O., & Ioannou, A. (2021). Envisioned pedagogical uses of chatbots in higher education and perceived benefits and challenges. In *Proceedings of the 8th International Conference on Learning and Collaboration Technologies (LCT 2021)* (pp. 230–250). Springer. https://doi.org/10.1007/978-3-030-77943-6_15

Vatsal, S., & Dubey, H. (2024). A survey of prompt engineering methods in large language models for different NLP tasks. arXiv preprint arXiv:2407.12994. https://arxiv.org/pdf/2407.12994.pdf

Villegas-Ch, W., García-Ortiz, J., & Sánchez-Viteri, S. (2024). Personalization of learning: Machine learning models for adapting educational content to individual learning styles. *IEEE Access, 12*, 10659866. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10659866

Xie, Y., Wu, S., & Chakravarty, S. (2023). AI meets AI: Artificial intelligence and academic integrity—A survey on mitigating AI-assisted cheating in computing education. In *Proceedings of the 24th Annual Conference on Information Technology Education (SIGITE '23)* (pp. 79–83). ACM. https://doi.org/10.1145/3585059.3611449