OST Eastern Switzerland University of Applied Sciences

Thesis & Project Report

Bachelor Thesis

AI as a Teacher's Assistant

Authors: Dominik Castelberg
Linus Flury
Supervisor: Prof. Dr. Mitra Purandare

OST, JUNE 2024

1 Abstract

1.1 Introduction

The digital transformation of higher education has led to a significant shift in teaching methods, with blended learning emerging as a favored approach. In this paradigm, traditional face-to-face teaching is combined with online learning, resulting in the well-known inverted classroom. In this setting, students begin with self-study, supported by multimedia materials, followed by interactive face-to-face sessions.

However, it was shown that students may struggle with self-study, highlighting the need for AI-supported learning assistants. These assistants should provide students with personalized guidance, feedback, and assessment, adapting to their individual needs and learning styles. By leveraging AI, primarily large language models (LLMs), these assistants can help students progress towards competency in an efficient and effective manner.

1.2 Approach

Three state-of-the-art LLMs (GPT-3.5, GPT-4 and Mixtral-8x7B) are evaluated on three subtasks based on a lecture script:

- Generating Questions
- Evaluating Answers
- Providing Feedback on the Current Study Level

The models are fine-tuned and the quality of their outputs were compared to their non-fine-tuned equivalents. A prototype application for a chat bot that supports multiple languages, model selection from a graphical user interface and an approach that combines chat history and RAG is built. Model access is wrapped under an abstracted class, allowing extensibility and enabling rapid integration of new models. Administrators assign documents and system prompts to individual chat bots, giving them granular control over their behavior and available information. The documents get embedded with a locally hosted Multilingual-E5-base instance.

1.3 Conclusion

Our work has shown that while there is potential in using LLMs in such a way, as the output quality of certain models were deemed usable however due to privacy concerns and scaling an open-source LLM would need to be used. Mixtral-8x7B fits such a criterion but is not reliable enough with our configuration to be used with students.

The prototype serves as a blueprint for history-aware and RAG-enabled chat bot applications. When provided with a locally hosted Mixtral-8x7B instance, the entire RAG process can be done locally. This gives its users complete control over the use of their data and ensures that classified information can be used to enhance the provisioned chat bots.
In its current form, the prototype is primarily limited by its resource consumption and therefore scalability concerns. But with a more refined deployment strategy, horizontal scaling is possible with minor adjustments.

2 Executive Summary

Introduction The digital transformation has shown itself to be highly disruptive to the higher education sector. With visual media, communication technologies and well-crafted digital experiences being more accessible than ever, we are currently witnessing the emergence of blended learning as a way to keep students engaged in the digital age. A favored approach is the inverted classroom, the combination of online learning and a following session of face-to-face teaching.
While this paradigm shift has received praise and shows promising results, students appear to struggle with self-studying. They attend classes while being woefully under-prepared, indicating that their self-assessment skills are lacking.

Project Goals This project aims to leverage the by now firmly established machine learning technology of LLMs to support students during their learning journey. We are conducting research into the viability of an LLM-backed chat bot acting as an on-demand teacher's assistant, capable of generating questions and assessing the knowledge level of a student it is interacting with.

Methodology This project is examining multiple methods to turn an LLM into a domain expert. We are validating the viability of widely used LLMs for this task and enhance their assessment abilities with fine-tuning and the usage of retrieval augmented generation. The outputs of those enhanced models are compared and the best methods to achieve our goals get evaluated based on those insights. Finally, we develop a prototype that can demonstrate the real-life application of our research.

Results We succeeded at creating a prototype that is able to provide in-depth and accurate feedback to students trying to gauge their readiness for their next in-person lecture. Surprisingly, our fine-tuned models were outperformed by their out-of-the-box counterparts, which warrants further investigation. RAG based approaches have shown themselves as being both comparatively easy to implement and incredibly powerful, providing strong results very quickly. Our prototype is built in a way that emphasizes extensibility, ensures data privacy and we have taken future-proofing measures to ensure that adaptation to scalability concerns will only be a relatively minor change in the future.

Recommendations Overall, we are very optimistic about the viability of LLMs as virtual teacher's assistants and encourage further investigation and experimentation in this direction. Especially RAG has proven itself as a highly efficient approach to specialize a model in specific fields of expertise, allowing the rapid creation of numerous chat bots with differing specializations while maintaining a high degree of specificity in their questioning of the student.

Contents

1 Abstract 2
1.1 Introduction 2
1.2 Approach 2
1.3 Conclusion 2

2 Executive Summary 3

3 Introduction 8
3.1 Background 8
3.2 Goals 8
3.3 Subproblems 9
3.4 Deliverables 9
3.5 Confidentiality 9

4 Research 10
4.1 Experimentations with RAG 10

5 Evaluation Process 12
5.1 Model Selection 12
5.1.1 Model Requirements 12
5.1.2 Further adjustments during the project 14
5.2 Model Deployment 15
5.2.1 Minimal requirements 15
5.3 Finetuning 16
5.3.1 Introduction 16
5.3.2 Process 16
5.4 Use Cases 21
5.4.1 Use Case 1: Question Generation 21
5.4.2 Use Case 2: Answer Evaluation 22
5.4.3 Use Case 3: Comprehension Evaluation 22
5.5 Evaluation Criteria (Use Cases) 23
5.5.1 Use Case 1: Question Generation 23
5.5.2 Use Case 2: Answer Evaluation 23
5.5.3 Use Case 3: Comprehension Evaluation 24
5.5.4 Overview 24
5.6 Prompting 25
5.6.1 Measures 26
5.6.2 System Prompts 28
5.6.3 User Prompts 28

5.6.4 Reasoning User Prompts . . . 29
5.7 Evaluation Environment . . . 30

6 Evaluation Results 31
6.1 Numerical Results . . . 31
6.2 Individual Numerical Results . . . 32
6.2.1 Score GPT 4 . . . 32
6.2.2 Score GPT 3 . . . 33
6.2.3 Score GPT 3.5 Finetuned . . . 33
6.2.4 Score Mixtral . . . 34
6.2.5 Score GPT 3.5 Finetuned . . . 34
6.3 Textresults . . . 35
6.4 Insights . . . 35
6.4.1 General Performance . . . 35
6.4.2 Finetuning vs. Prompting . . . 35
6.5 Mixtral vs. GPT-4 . . . 36
6.6 Token Usage & Cost . . . 36
6.7 Image Handling . . . 38

7 Prototype Application 40
7.1 Requirements . . . 40
7.2 Architecture Documentation Style . . . 40
7.2.1 Final Prototype Architecture . . . 40
7.2.2 Architecture Decision Records . . . 41
7.3 Final Prototype Architecture . . . 42
7.3.1 C4 Container Diagram . . . 42
7.3.2 C4 Component Diagrams . . . 43
7.3.3 Web API Logical Layers: Overview . . . 45
7.3.4 Web API Logical Layers: Controllers . . . 46
7.3.5 Web API Logical Layers: Services . . . 47
7.3.6 Web API Logical Layers: Data . . . 49
7.3.7 Application Database: Entity Relationship Model . . . 50
7.3.8 C4 Dynamic Diagrams . . . 51
7.4 Architecture Decision Records (ADRs) . . . 53
7.4.1 ADR001: Initial Application Data Model . . . 53
7.4.2 ADR002: Initial C4 Container Diagram . . . 54
7.4.3 ADR003: Initial C4 API Component Diagram . . . 55
7.4.4 ADR004: Initial C4 Web UI Container Diagram . . . 56
7.4.5 ADR005: Permission System Declared Out of Scope . . . 57
7.4.6 ADR006: Deployment Tools . . . 58

7.4.7 ADR007: Language Selection for API . . . 59
7.4.8 ADR008: Usage of LangChain . . . 60
7.4.9 ADR009: Usage of FastAPI . . . 61
7.4.10 ADR10: Usage of SQLModel . . . 62
7.4.11 ADR011: Usage of Angular . . . 63
7.4.12 ADR012: Usage of a local database . . . 64
7.4.13 ADR013: Usage of PostgreSQL . . . 65
7.4.14 ADR014: Usage of Min.IO . . . 66
7.4.15 ADR015: Introduction of LearningItemContainer . . . 67
7.4.16 ADR016: (REVERT) Introduction of LearningItemContainer . . . 68
7.4.17 ADR017: Introduce Model Wrapper . . . 69
7.4.18 ADR18: Introduce DocumentLoaders . . . 70
7.4.19 ADR19: Implement Services and Repositories as collections of static methods . . . 71
7.4.20 ADR20: Use of Multilingual-E5-base as the embedding model . . . 72
7.4.21 ADR021: Final API Architecture . . . 73
7.4.22 ADR022: Web UI Architecture . . . 74
7.4.23 ADR023: Final Application Data ERD . . . 75
7.4.24 ADR024: Final Container Diagram . . . 76
7.5 Deployment & Usage Guide . . . 77
7.5.1 Configuration . . . 77
7.5.2 Starting the Application . . . 78
7.5.3 Configuring a Chat Bot . . . 79
7.5.4 Using the Chat Bot . . . 80
7.5.5 Troubleshooting . . . 80
7.6 Limitations . . . 81
7.6.1 Known Bugs . . . 81
7.6.2 Potential Improvements . . . 81

8 Development Environment **83**
8.1 Prototype Development Environment . . . 83
8.2 Model Evaluation Development Environment . . . 84

9 Discussion **85**
9.1 Interpretation . . . 85
9.2 Answers to the initial questions . . . 86
9.2.1 Necessary Information . . . 86
9.2.2 Feedback . . . 86
9.2.3 Existing Products . . . 87
9.2.4 Existing Research . . . 87
9.2.5 Other applications . . . 88

9.2.6 Relevant Metrics 88
9.3 Applications 88
9.4 Limitations 89
9.4.1 Models 89
9.4.2 Prototype 89
9.5 Recommendations 90
9.6 Reflection: Dominik Castelberg 91
9.7 Reflection: Linus Flury 92
9.8 Acknowledgements 93

10 Project Plan **94**
10.1 Project Structure 94
10.1.1 Project Members 94
10.1.2 Project Meetings 94
10.1.3 Development Roles 94
10.2 Long Term Plan 95
10.3 Milestones 95
10.4 Tools 96
10.5 Time Recording 98

11 Attachments **99**
11.1 Evaluation Text Results 99
11.1.1 Use Case 1: Question Generation 99
11.1.2 Use Case 2: Answer Evaluation 105
11.1.3 Use Case 3: Question Generation 113

3 Introduction

3.1 Background

This project lays the groundwork for a potential AI based application to assist students in preparation for a lecture where the preceding studying of lecture material is necessary.

An evaluation from the Centre for Higher Education at OST (ZHB)[1] on the "Inverted Classroom", where students are provided with lecture scripts and learning videos ahead of the lecture which they are expected to study ahead of the lecture so that the lecture itself can be used for further discussions to deepen the understanding of the subject, showed that students might be under prepared for the lectures.

This lead to the idea of using an AI tool to let students prepare interactively ahead of the lecture which might increase the knowledge level of students before the lecture and benefit discussion quality during lectures.

3.2 Goals

The goal of this project is to provide an understanding of the challenges of implementing such an AI tool.

The following questions are to be answered:

1. What information is needed for the system to work, for example, does it suffice for the teacher to just provide the Lesson Script or does the teacher need to provide predefined questions, do these questions need an associated gold standard solution or can system figure out a correct answer based on the Script?
2. How can the system evaluate what is the student's comprehension of a certain topic and generate/look up new questions based on this information?
3. Is the system be able to give comprehensive feedback to the answers provided by the student? If the answer is not complete or wrong the system should not just tell that the answer is wrong but instead give the correct answer / full answer.
4. What are the pre-existing products/services (including sub-problems)?
5. What research was already done in this field?
6. What input from the teacher is needed for the system to work?
7. What are the current limitations of such a system?
8. Is such a system scalable to other application -> Different subjects, creative classes, and visual questions?
9. What are relevant metrics to evaluate the systems?
10. Does it work on the provided script?

---

[1] https://www.ost.ch/de/forschung-und-dienstleistungen/interdisziplinaere-themen/zhb-zentrum-fuer-hochschulbildung

3.3 Subproblems

Furthermore a prototype for the following subproblems should be created:

- Creating questions from a lesson script or predefined topic.
- Evaluating the answers on correctness and providing interactive feedback.
- Evaluating the student's comprehension of a certain topic and pose appropriate follow up questions.

To assess the challenges and limitations of such a system prototype should be built where the above question can be tested. The main goal is to get an understanding of the feasibility of such a system not to have a fully developed customer-ready product.

3.4 Deliverables

The expected deliverables are:

- Prototype Systems: Complete source code and research
- Comprehensive documentation of all aspects of the developed systems including approaches that were tried but were not implemented in the final prototype system.
- Recommendation on the feasibility of such a system with the current state of technology and what are potential challenges that could not have been address in this thesis.

All deliverables except for the source code and the complete collection of model evaluation Jupyter notebooks are part of this document. The complete source code and Jupyter notebooks are handed in separately in a ZIP-file.

3.5 Confidentiality

The entire project should be managed with a confidentiality level of internal. However, Prof. David Kobler has granted permission for the provided lecture script to be used with OpenAI's GPT services specifically for the purposes of evaluation and finetuning.

4 Research

4.1 Experimentations with RAG

Retrieval-Augmented Generation (RAG) is a process that was introduced by Meta[2] in 2020[8]. It is a powerful tool to provide large language models with additional contextual information during inference, allowing them to work with data they were not originally trained on.

A use case of interest for us is the provision of additional context, such as a lecture script, to the model being prompted. The model in question does not need to be fine-tuned for this process to lead to impressive results. Consider the following example of a brief conversation with a Learning Assistant based on GPT-4 that has been given additional context about the political rights of a Swiss citizen with RAG:

AI Assistant

```
Hello! I'm glad to see you're preparing for your 'Political Rights' lecture.
Let's start with some questions to assess your understanding.
Firstly, explain the role and significance of the Landsgemeinde institution
in the context of political rights.
```

Student

```
A Landsgemeinde is part of the voting process in Switzerland. It allows every
citizen to act on their right to vote.
```

AI Assistant

```
That's a good start, but let's delve a bit deeper. Please elaborate on the
principle of correct information for voters and how it is implemented in
Switzerland.
```

Here, the AI Assistant is referencing two concepts mentioned in the lecture scripts about political rights in Switzerland[3]: The **Landsgemeinde** and the **principle of correct information for voters**.

---

[2] https://www.meta.com

[3] Provided as a seperate document

Meanwhile, the same model with an identical system and user prompt, but no additional context provided with RAG shows this behavior:

AI Assistant

> `Hello! I'm here to help you prepare for your upcoming lecture on 'Political Rights'. Let's start by assessing your understanding of the topic. Please explain the concept of political rights and its significance in a democratic society.`

Student

> `Political rights are the rights a person has when interacting with a political system. One of those rights may be the right to vote, which would make it highly relevant to a democratic society.`

AI Assistant

> `That's a good start. Now, let's delve a bit deeper. Please describe the different types of political rights and how they are protected in a democratic society.`

While the client behaves in a similar way, as the behavior of the AI Assistant is dictated by its system prompt. However, the questions asked to the student are a lot less specific. This implies that the RAG approach has successfully specialized the model to be heavily influenced by the content of the additional context it was provided with RAG.

In addition, it should be noted that the embedded lecture script provided to the AI Assistant was written in German. We can conclude that this specific combination of embedding model (Multilingual-E5-base) and decoder model (GPT-4) can generalize information received to be used in other languages.

5 Evaluation Process

5.1 Model Selection

The selection of appropriate models to further evaluate is immensely important as with the right selection many time intensive evaluations can be omitted. For this selection, criteria were defined as "Must have" and "Should have". Further optional criteria were not introduced as selection criteria as the goal of the project was check the feasibility of using LLMs for the given tasks. If a model has extra features which are optional to the tasks but does not fulfill the "Must have" criteria, it will not be evaluated.

5.1.1 Model Requirements

"Must have"-Criteria:

- **"Chat bot":** The model must be able to interact with users in a back and forth conversation
- **Context:** The model must be able to hold a context during a session with a user
- **Specializable:** The model must be specializable to a topic, through RAG or fine-tuning.
- **Hosting:** The model must either be available in a cloud environment that is available to OST (e.g. Azure) or available to be locally deployed in a docker container.
- **Confidentiality:** The model must be able to be finetuned without breaching the confidentiality rules set on this project (see: Confidentiality 3.5).
- **Pricing:** The model must have a pricing range appropriate for a bachelor thesis

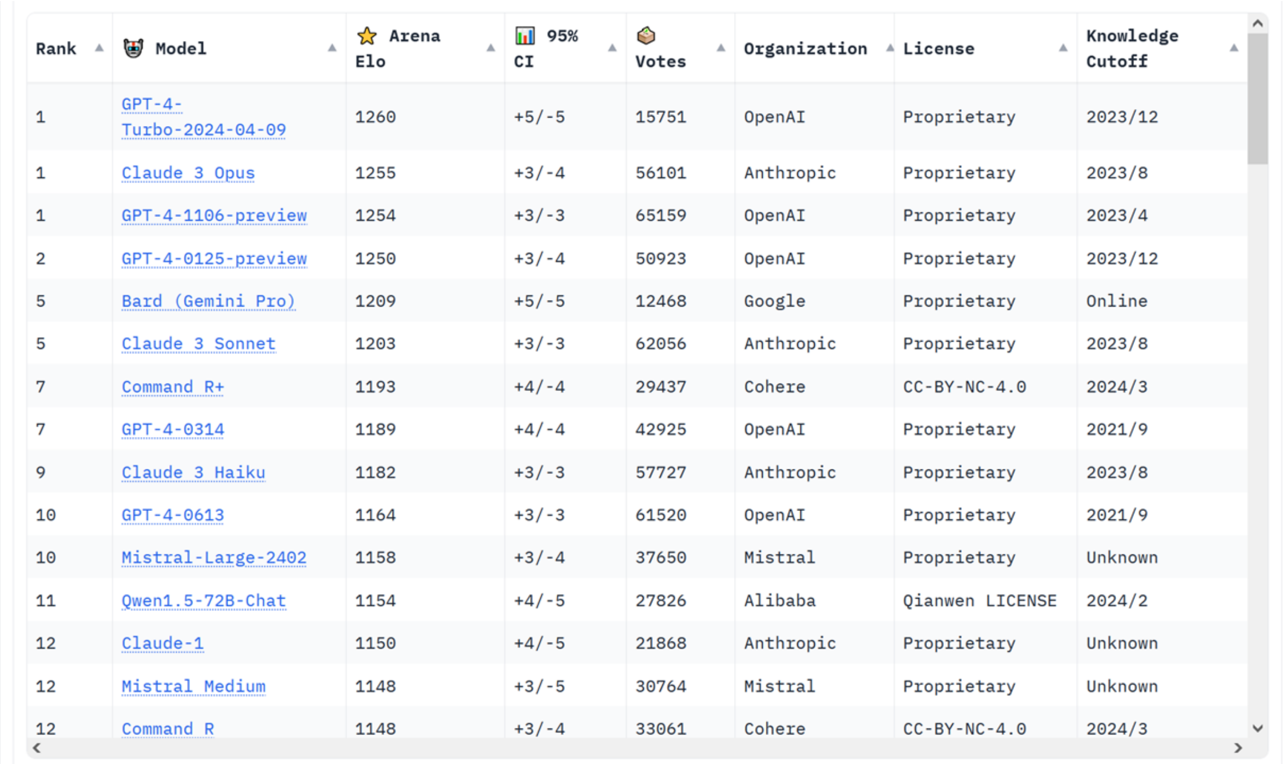
"Should have"-Criteria:

- **API:** The model should have an API which is compatible with langchain (see Tools 10.4).
- **PDF-Parsing:** The model should be able to parse PDF text files.
- **Resources:** The model must be able to produce responses within 3 seconds on the provided infrastructure.

Additionally the given tasks requires a comparison between open- and closed-source models and further comparisons between different styles of implementation of those models.
For this reason the following categories of models must be used:
Model Categories:

- **Closed Source**
  - Closed Source out of the box
  - Closed Source with finetuning
  - Closed Source with RAG
- **Open Source**
  - Open Source out of the box
  - Open Source with finetuning
  - Open Source with RAG
- **Expert/Composite**

In order to find candidate models, the community voted arena leader board on Huggingface.co[4] (see figure 1), in which the chat output of anonymous LLMs is rated against the output of another anonymous model, creating "duels" and a ELO-style leader board, was considered as well as previous experiences of the team members.



Rank	🤖 Model	⭐ Arena Elo	📊 95% CI	🗳️ Votes	Organization	License	Knowledge Cutoff
1	GPT-4-Turbo-2024-04-09	1260	+5/-5	15751	OpenAI	Proprietary	2023/12
1	Claude 3 Opus	1255	+3/-4	56101	Anthropic	Proprietary	2023/8
1	GPT-4-1106-preview	1254	+3/-3	65159	OpenAI	Proprietary	2023/4
2	GPT-4-0125-preview	1250	+3/-4	50923	OpenAI	Proprietary	2023/12
5	Bard (Gemini Pro)	1209	+5/-5	12468	Google	Proprietary	Online
5	Claude 3 Sonnet	1203	+3/-3	62056	Anthropic	Proprietary	2023/8
7	Command R+	1193	+4/-4	29437	Cohere	CC-BY-NC-4.0	2024/3
7	GPT-4-0314	1189	+4/-4	42925	OpenAI	Proprietary	2021/9
9	Claude 3 Haiku	1182	+3/-3	57727	Anthropic	Proprietary	2023/8
10	GPT-4-0613	1164	+3/-3	61520	OpenAI	Proprietary	2021/9
10	Mistral-Large-2402	1158	+3/-4	37650	Mistral	Proprietary	Unknown
11	Qwen1.5-72B-Chat	1154	+4/-5	27826	Alibaba	Qianwen LICENSE	2024/2
12	Claude-1	1150	+4/-5	21868	Anthropic	Proprietary	Unknown
12	Mistral Medium	1148	+3/-5	30764	Mistral	Proprietary	Unknown
12	Command R	1148	+3/-4	33061	Cohere	CC-BY-NC-4.0	2024/3

Figure 1: The lmsys leaderboard on 14.04.2024.

---

[4] https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard

Out of this the following models were selected:

- **GPT-4 Turbo** by OpenAI, Closed Source
- **GPT-3.5 Turbo** by OpenAI, Closed Source
- **Mixtral-8x7B-Instruct-v0.1** by Mistral AI, Open Source under Apache 2.0 license

GPT-4 was selected as the current leader on lmsys' arena leader board and because of the performance of GPT-4 during the project team's previous "Studienarbeit". However, finetuning of GPT-4 is currently only available as an experimental feature which the project team did apply for.
In order to have a secured comparison between a models output quality with and without finetuning, GPT-3.5 Turbo was included additionally.
Mixtral 8x7b is the Open Source version of Mistral AI's Mistral LLM, is trained on inputs of western European languages including German, and was selected as the highest scoring Open Source LLM on the lmsys arena leader board.

Notable mentions for popular LLMs that were not selected include Llama2 by Meta and Gemini by Google DeepMind. Gemini was not considered as an option to GPT-4, as at the start of the project finetuning was not available[5].
Llama2 was not considered as an alternative to Mixtral based on the output quality during the previous project which is also mirrored in the lmsys arena leader board ranking (70 billion parameter model reaching the third spot of open source models, while the 13 billion parameter version does not reach the top 10).

5.1.2 Further adjustments during the project

During preparation work for the evaluation work the following adjustments were made:

1. Accessing document content without any form of RAG is very token intensive as the document would need to be provided to the model in its entirety for each session. Either an implementation of RAG on the models side or on the callers side is necessary. For Mixtral the RAG implementation used in the prototype was used, for GPT Assistants[6] were used alongside Vector Stores and File-Search[7] .
2. In order to use OpenAI's Assistans with finetuned models the model GPT-3.5-turbo-0125 had to be used. Other versions were not compatible.
3. GPT-4 finetuning was at the project time in an experimental access program[8]. The project team tried to request access however were not deemed eligible users. If this were to pursued further, OST would likely need to apply.
4. GPT-4-turbo was used to evaluate GPT-4, as File Search was not available to GPT4.

---

[5] https://cloud.google.com/vertex-ai/generative-ai/docs/multimodal/multimodal-faqs?hl=en#is_fine_tuning_available_for_gemini
[6] https://platform.openai.com/docs/assistants/how-it-works
[7] https://platform.openai.com/docs/assistants/tools/file-search/
[8] https://platform.openai.com/docs/guides/fine-tuning/which-models-can-be-fine-tuned

5.2 Model Deployment

Mixtral models can be deployed through a docker container on CUDA (>=8.0)[9] enabled devices. Firstly a request to access to the model[10] with a huggingface account. A write token on huggingface then needs to be created[11]. For this the following command can be used[12] (either declare the write token as an environment variable or replace '$HF_TOKEN' in the command.

```
docker run --gpus all \
-e HF_TOKEN=$HF_TOKEN -p 8000:8000 \
ghcr.io/mistralai/mistral-src/vllm:latest \
--host 0.0.0.0 \
--model mistralai/Mixtral-8x7B-Instruct-v0.1 \
--tensor-parallel-size 2 # adapt to your GPUs \
--load-format pt # needed since both `pt` and `safetensors` are available
```

Note: If your GPU has CUDA capabilities below 8.0, you will see the error ValueError: Bfloat16 is only supported on GPUs with compute capability of at least 8.0. Your XXX GPU has compute capability 7.0. You need to pass the parameter –dtype half to the Docker command line.

5.2.1 Minimal requirements

While not explicitly mentioned on the website, multiple blogs like HardwareCorner[13] recommend a dual GPU setup with 32 GB of VRAM and 32 GB of RAM. In our evaluation the model ran successfully on a single RTX 4080[14] with 16 GB of VRAM and 32 GB of memory. However a lower precision (4-bit instead of 16-bit) had to be used. For the deployment of the finetuned Mixtral model, please refer to Finetuning 5.3.2.

---

[9] https://developer.nvidia.com/cuda-toolkit
[10] https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1
[11] https://huggingface.co/settings/tokens
[12] https://docs.mistral.ai/deployment/self-deployment/vllm/
[13] https://www.hardware-corner.net/llm-database/Mixtral/
[14] https://www.nvidia.com/de-de/geforce/graphics-cards/40-series/rtx-4080-family/

5.3 Finetuning

5.3.1 Introduction

Finetuning is a process used in machine learning and artificial intelligence to improve the performance of a pre-trained model on a new, specific task. Building and training a machine learning model from scratch requires a lot of data and computational resources. To save time and effort, researchers often use pre-trained models. These are models that have already been trained on large datasets for a similar task. For instance, an image labeling model trained on millions of general images can recognize a wide range of objects and features.

Finetuning is a part of a broader technique called transfer learning. Transfer learning leverages the knowledge a model has acquired from a previous task to enhance performance on a new task. Instead of starting from scratch, a pre-trained model can be used as a base model and be adjusted slightly to fit the specific needs.

For this reason the Instruct[15] variant of Mixtral was used over the base[16] variant as it was pre-trained for a dialog interaction.

The model is then trained on specific data which represents the data that the resulting model should primarily work on. Many finetuning tools use the 'Low-Rank Adaptation of Large Language Models' (LoRA) training paradigm, where most weights are kept and only 1-2% of additional weights are trained [6].

In this project this would relate to sample interactions between users and LLMs which represent each subtask. To evaluate the effectiveness of this finetuning, separate finetuned models were analysed for each subtask.

5.3.2 Process

GPT-3.5:

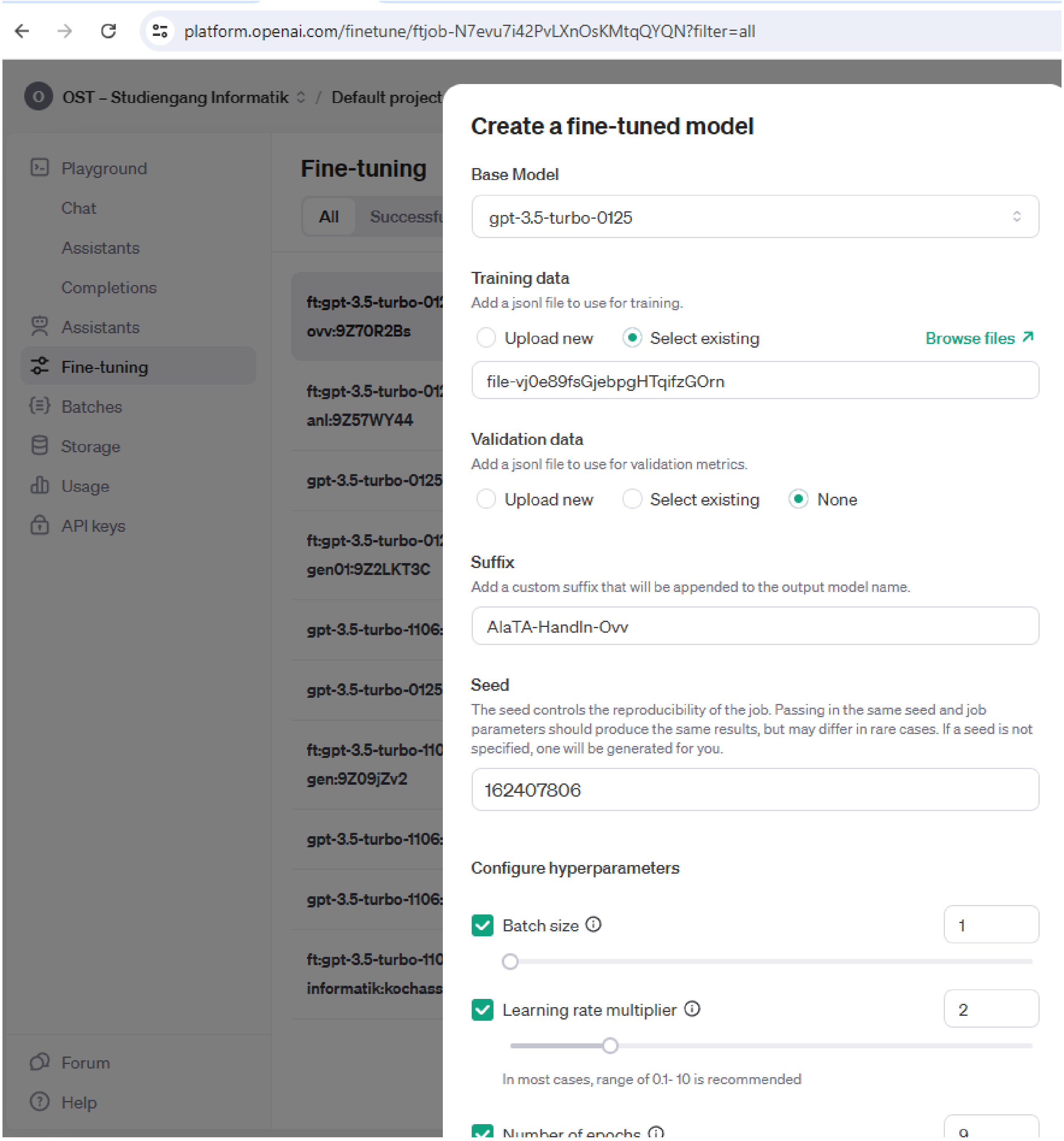
OpenAi provides finetuning through their dashboard[17](see Figure 3). Training data needs to be provided using .jsonl files. The content of these files needs to be in the following format[18] (line breaks in are added for readability, not in the actual .jsonl file). As the acceptable responses of a model are a wide range of possible outputs, using validation data was omitted to prevent overfitting to the exact answers provided in the validation data.

---

[15] https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1
[16] https://huggingface.co/mistralai/Mixtral-8x7B-v0.1
[17] https://platform.openai.com/finetune
[18] https://platform.openai.com/docs/guides/fine-tuning/example-format

```
{
  "messages": [
    {
      "role": "system",
      "content": "You are a teaching assistant. You take lecture infos in pdf format and
                  generate university level questions for students to check if the students
                  have understood the topic. Your responses are the questions in German. The
                  questions should not ask for facts, instead they should let the students
                  compare or argue for a topic (Blooms Taxonomy Level understand and up)."
    },
    {
      "role": "user",
      "content": "Generate a new question."
    },
    {
      "role": "assistant",
      "content": "Erklären Sie, weshalb im Kanton Glarus 16-Jährige an den Ständeratswahlen
                  teilnehmen dürfen!"
    }
  ]
}
```

Figure 2: GPT finetuning training data format example



Create a fine-tuned model

Base Model

gpt-3.5-turbo-0125

Training data

Add a jsonl file to use for training.

○ Upload new ● Select existing

file-vj0e89fsGjebpgHTqifzGOrn

Validation data

Add a jsonl file to use for validation metrics.

○ Upload new ○ Select existing ● None

Suffix

Add a custom suffix that will be appended to the output model name.

AIaTA-HandIn-Ovv

Seed

The seed controls the reproducibility of the job. Passing in the same seed and job parameters should produce the same results, but may differ in rare cases. If a seed is not specified, one will be generated for you.

162407806

Configure hyperparameters

☑ Batch size ⓘ 1

☑ Learning rate multiplier ⓘ 2

In most cases, range of 0.1 - 10 is recommended

☑ Number of epochs ⓘ 9

Figure 3: OpenAi Dashboard Fine-tuning example

The finetuned models are available on the OpenAI platform [19] for the organization OST - Studiengang Informatik.

---

[19] https://platform.openai.com

Use Case	Suffix	Model ID
Use Case 1: Question Generation	AIaTA-HandIn-Gen01	ft:gpt-3.5-turbo-0125:ost-studiengang-informatik:aiata-handin-gen01:9Z2LKT3C
Use Case 2: Answer Evaluation	AIaTA-HandIn-Anl	ft:gpt-3.5-turbo-0125:ost-studiengang-informatik:aiata-handin-anl:9Z57WY44
Use Case 3: Comprehension Evaluation	AIaTA-HandIn-Ovv	ft:gpt-3.5-turbo-0125:ost-studiengang-informatik:aiata-handin-ovv:9Z70R2Bs

Table 1: Use Cases and Model Information

Mixtral:

Multiple open source tools were available and tested for finetuning. These include:

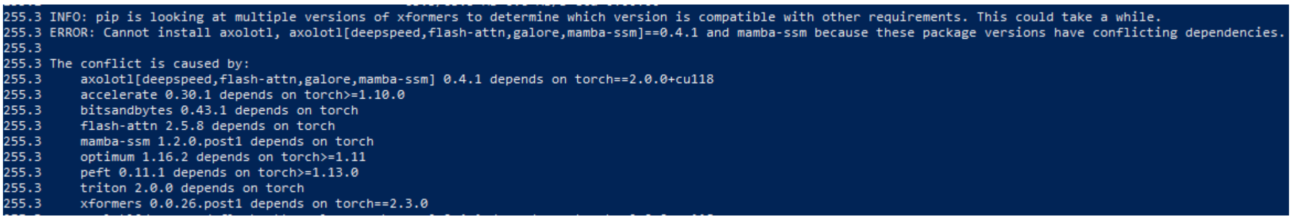
- **Axolotl[20]:** Axolotl provides dockerised finetuning of models including Mixtral-MoE[21] (Mixtral of Experts) which is the official name of Mixtral 8x7B. Axolotl can be run on CUDA[22] enabled devices even on a single GPU. Models can be pulled from huggingface or be used locally. It can use a wide range of training data formats including custom formats [23]. Finetuning pardigms like LoRA and Quantized LoRA (qlora)[4] are available to finetune Mixtral Models. However it must be noted that during the final trial on June 6, the docker image used (winglian/axolotl-base:main-base) did not run properly due to torch dependency conflicts (see Figure 4). Instead the image winglian/axolotl:main-py3.10-cu121-2.1.1 could be used.

- **Mistral Finetune[24]:** Mistral provides a lightweight tool to finetune their models on a CUDA enabled device. It contains python scripts to locally train Mistral and Mixtral models using LoRA.

- **LLaMA-Factory:[25]:** Despite it's name LLaMA-Factory is a tool to finetune language models including LLaMA, Mistral/Mixtral, or Microsoft's Phi-3[26]. The tool provides a dockerised deployment, can be run on CUDA enabled devices and provides a Gradio[27] based GUI (see Figure 5. Additionally it can provide a OpenAI-style API and vLLM[28] for inference with the model, which can be used from the prototype application.

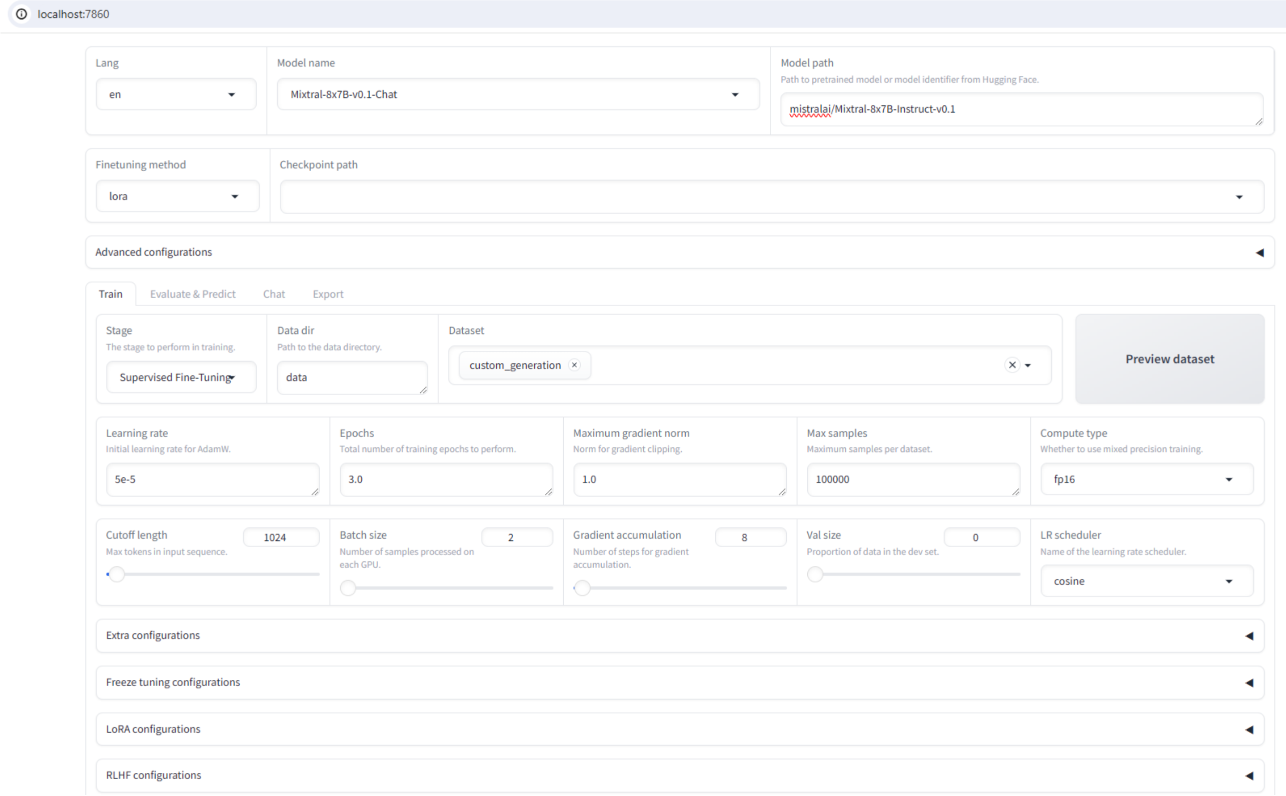
---

[20] https://github.com/OpenAccess-AI-Collective/axolotl
[21] https://mistral.ai/news/mixtral-of-experts/
[22] https://developer.nvidia.com/cuda-toolkit
[23] https://openaccess-ai-collective.github.io/axolotl/docs/dataset-formats/inst_tune.html
[24] https://github.com/mistralai/mistral-finetune
[25] https://github.com/hiyouga/LLaMA-Factory
[26] https://huggingface.co/collections/microsoft/phi-3-6626e15e9585a200d2d761e3
[27] https://www.gradio.app/
[28] https://github.com/vllm-project/vllm



```
255.3 INFO: pip is looking at multiple versions of xformers to determine which version is compatible with other requirements. This could take a while.
255.3 ERROR: Cannot install axolotl, axolotl[deepspeed,flash-attn,galore,mamba-ssm]==0.4.1 and mamba-ssm because these package versions have conflicting dependencies.
255.3
255.3 The conflict is caused by:
255.3     axolotl[deepspeed,flash-attn,galore,mamba-ssm] 0.4.1 depends on torch==2.0.0+cu118
255.3     accelerate 0.30.1 depends on torch>=1.10.0
255.3     bitsandbytes 0.43.1 depends on torch
255.3     flash-attn 2.5.8 depends on torch
255.3     mamba-ssm 1.2.0.post1 depends on torch
255.3     optimum 1.16.2 depends on torch>=1.11
255.3     peft 0.11.1 depends on torch>=1.13.0
255.3     triton 2.0.0 depends on torch
255.3     xformers 0.0.26.post1 depends on torch==2.3.0
```

Figure 4: Dependency issue with current Axolotl container



localhost:7860

Lang: en
Model name: Mixtral-8x7B-v0.1-Chat
Model path (Path to pretrained model or model identifier from Hugging Face.): mistralai/Mixtral-8x7B-Instruct-v0.1
Finetuning method: lora
Checkpoint path:

Stage (The stage to perform in training.): Supervised Fine-Tuning
Data dir (Path to the data directory.): data
Dataset: custom_generation
Learning rate (Initial learning rate for AdamW.): 5e-5
Epochs (Total number of training epochs to perform.): 3.0
Maximum gradient norm (Norm for gradient clipping.): 1.0
Max samples (Maximum samples per dataset.): 100000
Compute type (Whether to use mixed precision training.): fp16
Cutoff length (Max tokens in input sequence.): 1024
Batch size (Number of samples processed on each GPU.): 2
Gradient accumulation (Number of steps for gradient accumulation.): 8
Val size (Proportion of data in the dev set.): 0
LR scheduler (Name of the learning rate scheduler.): cosine

Figure 5: LLama-Factory GUI for finetuning and inference

LLaMA-Factory was selected for finetuning, as it directly enabled the resulting model to be used as an endpoint for our prototype. The OpenAI data format can be used as well to finetune Mixtral using LLaMA-Factory. Each dataset then needs to be declared in dataset_info.json. The default settings were used except for the training data input Cutoff Length which was increased to 4096 tokens, using the model Mixtral-8x7B-v0.1-Chat from mistralai/Mixtral-8x7B-Instruct-v0.1 (see Figure 5).

```
"dataset_name": {
  "file_name": "Generation_Finetuning.jsonl",
  "formatting": "sharegpt",
  "columns": {
    "messages": "messages"
  },
  "tags": {
    "role_tag": "role",
    "content_tag": "content",
    "user_tag": "user",
    "assistant_tag": "assistant",
    "system_tag": "system"
  }
},
```

Figure 6: Finetuning data declaration for LLama-Factory

5.4 Use Cases

For the evaluation of the models three use cases were defined. These use cases are based on the subproblems defined in the task section **??** and are designed to evaluate the usefulness of the models for given tasks.

5.4.1 Use Case 1: Question Generation

The following requirements were given to the project team:

- The models are provided with a German lesson script and are tasked with generating questions based on the content of the script.
- The generated questions are to be written in German, as that represents the language of the script and the target audience.
- The questions are to be generated in a way that they are answerable by the content of the script.
- The questions should be posed as an imperative.
- However the questions should not be simple recall questions, but questions that require the student to think about the content and apply it in a new context. (At least level "Understand" according to Bloom's taxonomy, see Figure 7).

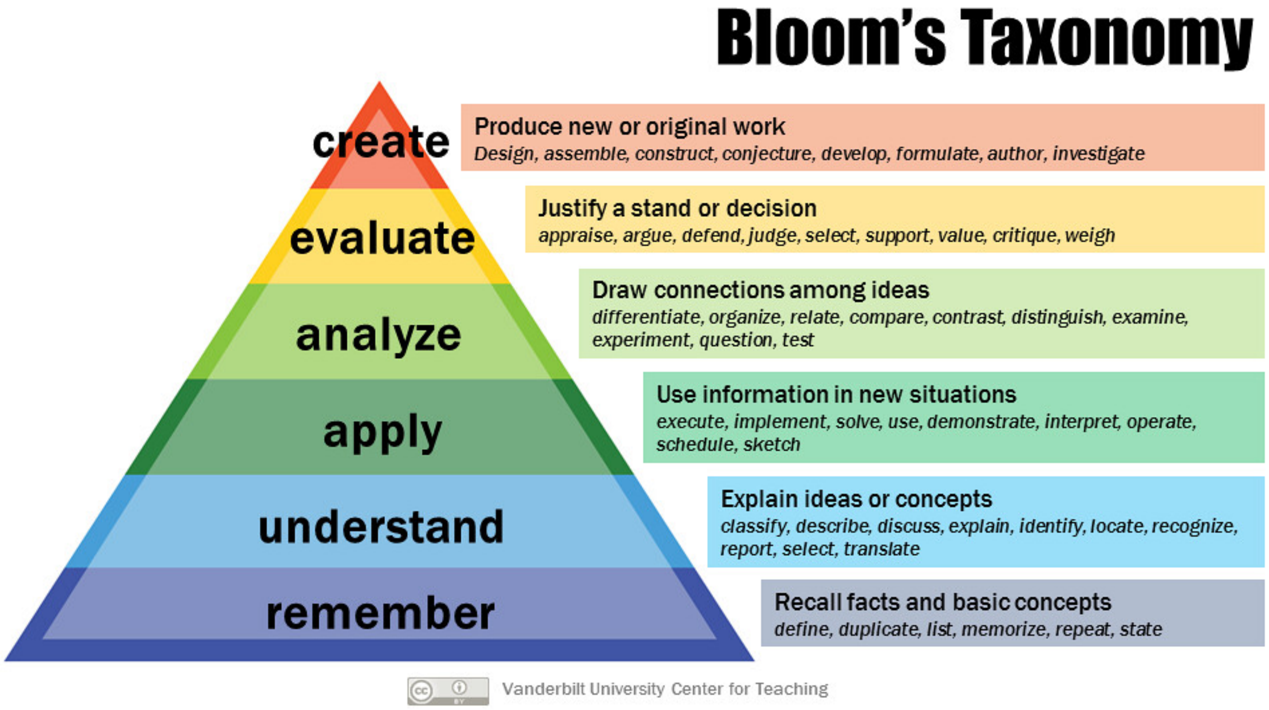

Bloom's Taxonomy
create
evaluate
analyze
apply
understand
remember
Produce new or original work
Design, assemble, construct, conjecture, develop, formulate, author, investigate
Justify a stand or decision
appraise, argue, defend, judge, select, support, value, critique, weigh
Draw connections among ideas
differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test
Use information in new situations
execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch
Explain ideas or concepts
classify, describe, discuss, explain, identify, locate, recognize, report, select, translate
Recall facts and basic concepts
define, duplicate, list, memorize, repeat, state
Vanderbilt University Center for Teaching

Figure 7: The revised Bloom's Taxonomy, showing the cognitive processes by which thinkers encounter and work with knowledge[1].

5.4.2 Use Case 2: Answer Evaluation

The following requirements were given to the project team:

- The models are provided with a German lesson script.
- The models are tasked with evaluating answers provided by hypothetical students to questions about the lesson script.
- The models are to provide feedback to the students based on the correctness of their answers.
- The feedback should be comprehensive, providing the correct answer based on the script or asking a follow up question if the student's answer is wrong or incomplete.
- The feedback should be written in German, as that represents the language of the script and the target audience.

5.4.3 Use Case 3: Comprehension Evaluation

The following requirements were given to the project team:

- The models are provided with a German lesson script.
- The models are tasked with evaluating the comprehension of hypothetical students based on a set of questions and their answers about the lesson script.

- The models are to provide a feedback on the level of comprehension of the students. It should provide topics of the document which should be examined again.
- The feedback should be written in German, as that represents the language of the script and the target audience.

5.5 Evaluation Criteria (Use Cases)

Based on the previous experiences made by the authors in their previous project the challenge of evaluating qualitative attributes efficiently and consistently was already known.
Similar to that project the decision was made to forgo qualitative attributes and instead to replace them with binary "Yes/No" evaluations.
Those evaluations were assigned a respective weight in the range of 1-5 in accordance to their importance to users. The weights were then normalized with a total weight for each category of 100. This allows an even comparison over all use cases and results in a percentage score without needing to convert.
The evaluation is done on a single response per model, usecase and style.[3]

5.5.1 Use Case 1: Question Generation

The following criteria were defined for the evaluation of the models in the question generation use case:

1. **Correctness:** The model generates questions that are answerable by the content of the script.
2. **Language:** The questions are written in correct German.
3. **Understandability:** The questions are understandable on first read through.
4. **Complexity:** The questions are not simple recall questions, but require the student to think about the content and apply it in a new context.
5. **Variety:** The questions are varied in their style. Not more than 40% of the questions are of the same verb.
6. **Topic Variety 5:** The questions cover at least 5 different topics from the script without repeating the same topic.
7. **Topic Variety 10:** The questions cover at least 10 different topics from the script without repeating the same topic.
8. **Variety 20:** The questions cover at least 10 different topics from the script without repeating the same topic and there is no repeating question.

5.5.2 Use Case 2: Answer Evaluation

The following criteria were defined for the evaluation of the models in the answer evaluation use case:

1. **Correctness:** The model evaluates the answers correctly.
2. **Categories:** The model provides feedback based on the categories "Correct", "Incomplete", "Wrong".
3. **Feedback Existence:** The model provides feedback to the student.
4. **Feedback Quality:** The feedback is comprehensive, providing the correct answer based on the script or asking a follow-up question if the student's answer is wrong or incomplete.
5. **Language:** The feedback is written in correct German.

6. **Understandability:** The feedback is understandable on first read through.
7. **Location:** The feedback points to the location in the script where the answer can be found.
8. **On Topic:** The feedback is based solely on the content of the script.

5.5.3 Use Case 3: Comprehension Evaluation

The following criteria were defined for the evaluation of the models in the comprehension evaluation use case:

1. **Correctness:** The model evaluates the comprehension of the students appropriately.
2. **Coverage:** The model identifies where additional questions are needed based on the topics covered by the questions.
3. **Knowledge Gaps:** The model identifies where the student has knowledge gaps.
4. **Location:** The model specifies the location in the script where the student has knowledge gaps.
5. **Detail:** The model provides the student with detailed feedback on their comprehension (not just a list of locations).
6. **Language:** The feedback is written in correct German.
7. **Understandability:** The feedback is understandable on first read through.
8. **Follow-up:** Additional questions or suggestions are provided.

5.5.4 Overview

Criteria	**Weight**	**Normalized Weight**
Correctness	5	20
Language	4	16
Understandability	4	16
Complexity	2	8
Variety	2	8
Topic Variety 5	3	12
Topic Variety 10	3	12
Variety 20	2	8
Total	**25**	**100**

Table 2: Criteria weights: Use Case 1: Question Generation

Criteria	**Weight**	**Normalized Weight**
Correctness	5	17
Categories	4	13
Feedback Existence	4	13
Feedback Quality	4	14
Language	4	13
Understandability	4	13
Location	2	7
On Topic	3	10
Total	**30**	**100**

Table 3: Criteria weights: Use Case 2: Answer Evaluation

Criteria	**Weight**	**Normalized Weight**
Correctness	5	17
Coverage	4	13
Knowledge Gaps	4	14
Location	2	7
Detail	3	10
Language	4	13
Understandability	4	13
Follow up	4	13
Total	**30**	

Table 4: Criteria weights: Use Case 3: Comprehension Evaluation

5.6 Prompting

Large Language Models play a vital role in natural language processing, serving as the foundation for a variety of applications including machine translation, text generation, and speech recognition. The effectiveness of LLMs largely hinges on their ability to produce output based on prompts, which effectively steer the model toward generating the intended results.

In the context of LLMs, prompting involves providing the model with an initial input text to elicit a particular output. This prompt acts as the model's starting point, and its quality has a significant impact on the resulting output. A well-crafted prompt can lead the model to produce relevant, thorough, and contextually precise responses, ensuring a high standard of output. Prompting is important in LLMs for several reasons:

- **Enhances Contextual Understanding:** Effective prompts enable the model to comprehend the task's context,

resulting in more accurate and pertinent responses.
- **Improves Coherence:** The structure and clarity of prompts influence the coherence and logical flow of the LLM's output. A carefully designed prompt can help the model generate a coherent narrative.
- **Ensures Relevance:** The specificity and clarity of the prompt affect the relevance of the output. Clear, concise, and detailed prompts can lead the LLM to produce highly relevant and focused content.
- **Encourages Creativity:** The type of prompt can spark the LLM's creativity. Open-ended or complex prompts can inspire the model to generate innovative and diverse responses.

For this project, the prompt engineering guidelines provided by OpenAI were utilized as appropriate[10].

5.6.1 Measures

The measures taken are explained with the following question generation system prompt:

> You are a teaching assistant. You look for documents in the vector store and generate university level questions for students to check if the students have understood the topic. Your responses are the questions in German. The questions should not ask for facts, instead they should let the students compare or argue for a topic (Blooms Taxonomy Level understand and up). Only output the questions. Write the questions as imperatives. No introduction or finishing text. Generate 20 questions that cover the entire document.

- **Include details in your query to get more relevant answers:** Sharpening the output questions to 'university level' and specifying further using Blooms Taxonomy increases the consistency of output questions. While some regular "What is"- style questions can show up the frequency gets reduced. Originally questions were generated with a broad statement: *Generate 20 questions based on your available document*. This lead to questions like: *Welches Sitzzuteilungsverfahren wird in Proporzsystemen zur Verteilung der Parlamentssitze gemäß dem PDF nicht verwendet?*
  A) Hare-Niemeyer-Verfahren B) D'Hondt-Verfahren C) Hagenbach-Bischoff-Verfahren D) Jefferson-Verfahren
  In a discussion with Prof. David Kobler he made clear that these simple fact recalling questions are not wished for. The prompt was therefor adjusted to include Blooms Taxonomy and add a restrictions against straight factual questions. This lead to implicitly prohibiting multiple choice questions as well. The prompt was further adjusted until all models were able to generally respond in the required style.
- **Ask the model to adopt a persona:** In the system prompt above the model was given the role of a teaching assistant. This does not include a specific sentiment. Without a sentiment specified the responses tended to be emotionally neutral. However this can lead to very harsh seeming statements when the level of understanding of a student. Example:
  Der Studierende hat sich weiterhin ungenügend mit den behandelten Themen auseinandergesetzt. (...)
  Whilst factually correct as there was no effort present in the answers ("Don't know.", "No idea" etc.) providing these responses are not helpful for the study preparation efforts of students. A simple adjustment would be to adjust the role assigning prompt to:
  You are a helpful teaching assistant, helping students to prepare for a lecture.

This was not used during evaluation where the straight factual approach was more clear to evaluate. Evaluating statements like:
Guter Gedanke, was würdest du aber sagen unterscheidet (...) would introduce additional fuzziness as it is not a clear *correct/incomplete/false* differentiation.

- **Split complex tasks into simpler subtasks:** This is the underlying thought behind the separation into three separate use cases. Focusing only on the question generation with its requirements allows for more complex individual requirements.
- **Provide examples:** This measure was not used in the evaluation as it would favour a specific style of questioning, which reduces response variety. Having the same style of questions asked over and over was deemed to be a potential cause of study fatigue so it was not used. A possible example addition for question generation system prompt could look like this:
Old: "Write the questions as imperatives." / New: "Write the questions as imperatives like this example: 'Argumentiere inwiefern die Landsgemeinde politische Rechte verletzen könnte!'"
- **Language:** All evaluated models were capable of interacting in German or English. As GPT-3 is partially trained on data scraped off the internet using Common Crawl[29] English was the most prevalent language it was trained on. English was kept for system prompts to avoid possible misunderstandings. User prompts were kept in German to simulate the intended interaction.[?]

---

[29] https://commoncrawl.org/

5.6.2 System Prompts

The LLMs were evaluated using the following system prompts:

- **Use Case 1: Question Generation System Prompt**: *You are a teaching assistant. You look for documents in the vector store and generate university level questions for students to check if the students have understood the topic. Your responses are the questions in German. The questions should not ask for facts, instead they should let the students compare or argue for a topic (Blooms Taxonomy Level understand and up). Only output the questions. Write the questions as imperatives. No introduction or finishing text. Generate 20 questions that cover the entire document.*
- **Use Case 2: Answer Evaluation System Prompt**: *You are a teaching assistant. You take questions about 'Politische Rechte in der Schweiz' with multiple answers. You assess the correctness of each answer to each question, give your reasoning for why it is correct or wrong based on the data in your vector store and provide a follow up question and an explanation if the answer was wrong. Your responses are the evaluations and followup questions in German.*
- **Use Case 3: Comprehension Evaluation System Prompt**: *You are a teaching assistant. You evaluate a series of questions and answers provided by a student based on the file on your vector storage. You assess the level of understanding of of the student on the given text (are there any areas not covered or with wrong answers) and provide followup questions on the rest of the text to which there has not yet been a correct answer by the student. Your responses are the evaluation in German. Your output is only the level of understanding, the reasoning behind it, possible followup questions. Finally ask new questions based on the File that were not covered yet.*

5.6.3 User Prompts

The response for the Evaluation was triggered using the following user prompts:

- **Use Case 1: Question Generation User Prompt**:
  Generiere 20 Fragen basierend auf dem bereitgestellten Dokument über politische Rechte in der Schweiz. Die Fragen sollen im Imperativ und mindestens auf der Ebene des Verstehens nach Blooms Taxonomie gestellt werden. Die Ausgabe beinhaltet nur die Fragen.
- **Use Case 2: Answer Evaluation User Prompt**:
  Frage A: Was sind die formalen Voraussetzungen für die Ausübung politischer Rechte auf Bundesebene in der Schweiz?
  Antwort A1: Zu den Voraussetzungen gehören das Schweizer Bürgerrecht, das Erreichen des 18. Lebensjahres und das Fehlen verfassungsmäßig definierter Ausschlussgründe.
  Antwort A2: Die Voraussetzungen umfassen das Schweizer Bürgerrecht und das 18. Lebensjahr.
  Antwort A3: Alle Einwohner in der Schweiz, die mindestens 16 Jahre alt sind, können an Wahlen teilnehmen.
  Frage B: Was versteht man unter dem aktiven und passiven Wahlrecht?
  Antwort B1: Das aktive Wahlrecht ermöglicht es, an Wahlen teilzunehmen, während das passive Wahlrecht das Recht beinhaltet, gewählt zu werden.

Antwort B2: Aktives Wahlrecht bedeutet, für politische Ämter zu kandidieren, und passives Wahlrecht erlaubt die Teilnahme an der Wahl.
Antwort B3: Aktives Wahlrecht bedeutet, wählen zu dürfen.
Frage C: Was beschreibt ein obligatorisches Referendum?
Antwort C1: Ein obligatorisches Referendum kann von 50.000 Bürgern ausgelöst werden, wenn sie dies fordern.
Antwort C2: Ein obligatorisches Referendum findet statt, wenn die Regierung es bestimmt.
Antwort C3: Ein obligatorisches Referendum tritt bei wichtigen rechtlichen Änderungen,wie Verfassungsänderungen, automatisch in Kraft und muss vom Volk bestätigt werden.

- **Use Case 3: Comprehension Evaluation User Prompt**:
  Frage1: Weshalb dürfen im Kanton Glarus 16-Jährige an den Ständeratswahlen teilnehmen?
  Antwort1: Weiss ich nicht.
  Frage2: Was sind politische Rechte?
  Antwort2: Keine Ahnung.
  Frage3: Welche politischen Rechte gibt es in der Schweiz auf Bundesebene?
  Antwort3: Frei zu leben.
  Frage4: Begründe warum eine Listenverbindung bei Proporzwahlen für eine grosse Partei mit kleineren Parteien Sinn machen könnte.
  Antwort 4: Bessere Ausstrahlungskraft.
  Frage5: Erläutere den Grundsatz der Einheit der Materie.
  Antwort5: Die Einheit der Materie besagt, dass in einer Vorlage nur Fragen umfasst werden dürfen, welche einen inneren Zusammenhang haben.
  Frage 6: Benenne den Unterschied zwischen folgenden Situationen: A) Ein Wähler streicht ein Kandidat von seiner Liste B) Ein Wähler streicht ein Kandidat von seiner Liste und ersetzt die Person durch einen anderen Kandidaten derselben Liste.
  Antwort6: Für die Liste macht dies kein Unterschied, die leere Stimme wird dennoch für die Liste gezählt. Für den Kandidaten macht dies aber ein Unterschied, da er so eine Stimme mehr bekommen hat und so bessere Chancen auf ein Sitz hat.

5.6.4 Reasoning User Prompts

- **Use Case 1: Question Generation:** The instruction from the system prompt was repeated without role assigning as a user prompt. This was done to keep the structure of system prompts similar between use cases. However a user input was needed to create a response.
- **Use Case 2: Answer Evaluation** A mixture of correct, incomplete and false answers was utilised to analyse whether the LLMs were capable of differentiating the correctness with a proper level of detail. Additionally it can be checked if LLMs react differently between incorrect and incomplete answers. The prompt was kept to 9 answers in order to make sure the LLMs are respond properly within a single response without crossing the token limit.

- **Use Case 3: Comprehension Evaluation**: Blocks of non-answers followed by wrong answers followed by correct answers were used to evaluate if LLMs are able to keep an overview without a bias to the start or end of an answer. It allows for a nuanced response mentioning partial gaps in knowledge and allowing further questions on wrongly answered or not covered topics.

5.7 Evaluation Environment

Model responses were collected and evaluated in Jupyter Notebooks(see **??**). This was based on the previous work from the authors where Llama2 and GPT-3 models were evaluated using a unified structure using Langchain (see 10.4). It was thought to keep the identical structure and implement calls to Mixtral and GPT in the same fashion. However OpenAI's Assistants and file_search tools were still in beta and there was no Langchain implementation yet. This meant that if RAG was used, OpenAI calls would need to be done using OpenAI's python SDK[30] v2 (requires openai version of >=1.20.0). Assistants were created through the dashboard and through the sdk, however when trying to create agents through the SDK following the documentation[31], we were presented with errors like: *Error code: 400 - 'error': 'message': "The requested model 'gpt-4o' cannot be used with the Assistants API in v1. (...)* even when explicitly following the directions given by OpenAI to force v2.[32]. Trying to retrieve assistants created in the dashboard through *client.beta.assistants.create* worked, however their interaction with the vector store was no longer working properly (following instructions on file_search[33] after assistant retrieval):

Sample Response of assistant when prompted to generate questions based on the file in the vector store:
"Kein Dokument vorhanden. Bitte stellen Sie ein Dokument bereit, aus welchem Fragen generiert werden sollen."

It was decided to rather use the dashboard directly and forgo calling via the beta Assistant / file_search SDK. Prompts, responses and system prompts were persisted in the Jupyter Notebooks and attached files.
As Mixtral prompts required the RAG implementation first, it was decided to use the prototype application to generate the Mixtral responses and to collect and evaluate them in the identical fashion in Jupyter Notebooks.

The evaluation was done manually by marking the fulfilled criteria on each run. Those results were then pragmatically collected, associated with their criterion weight and aggregated in graphs through another Jupyter Notebook.

---

[30] https://platform.openai.com/docs/libraries
[31] https://platform.openai.com/docs/assistants/tools/file-search/quickstart
[32] https://platform.openai.com/docs/assistants/migration
[33] https://platform.openai.com/docs/assistants/tools/file-search/step-5-create-a-run-and-check-the-output

6 Evaluation Results

6.1 Numerical Results

The numerical evaluation came to the following resulting scores:

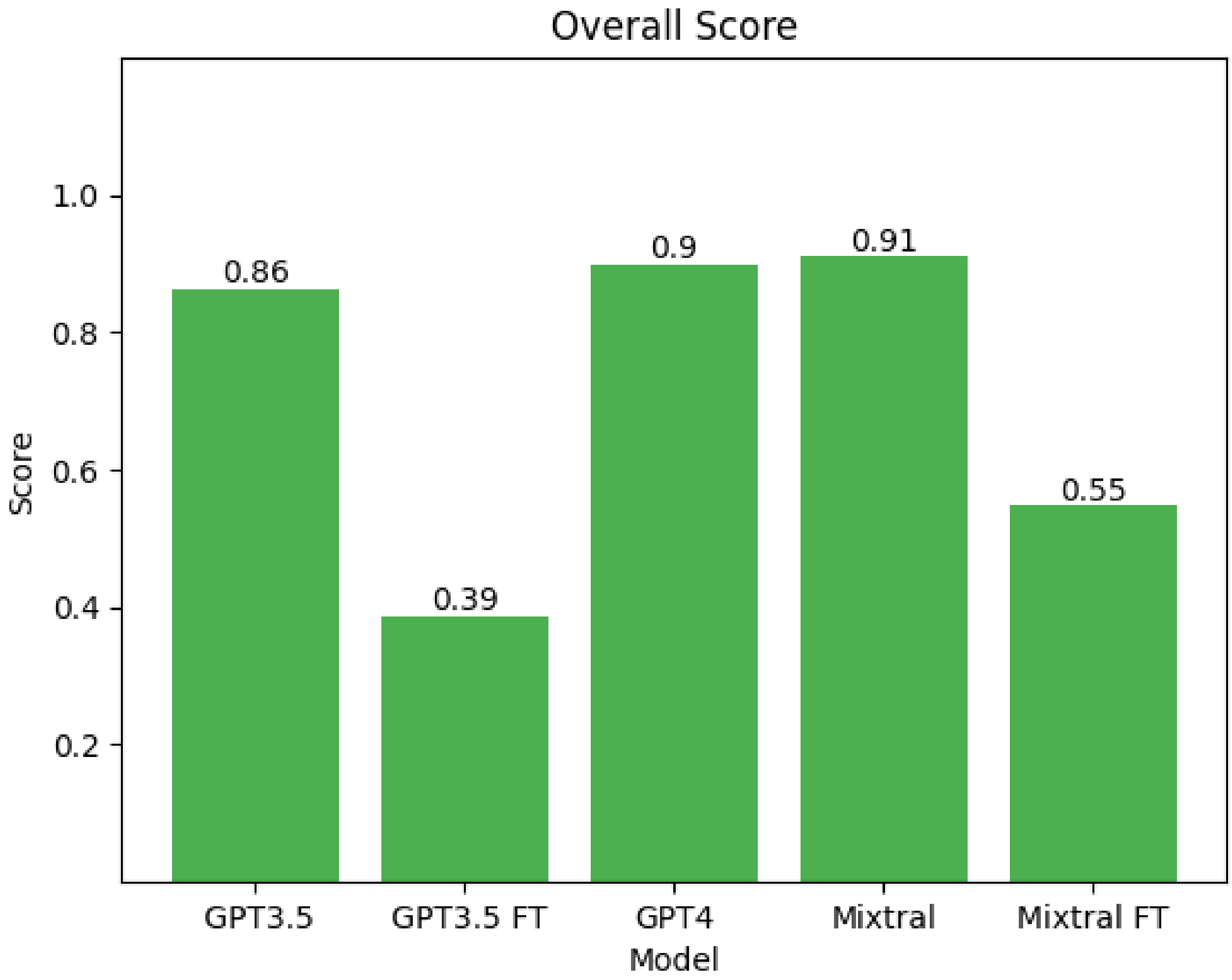


Figure 8: Overall scores of the models based on the evaluation criteria5.5.

6.2 Individual Numerical Results

6.2.1 Score GPT 4

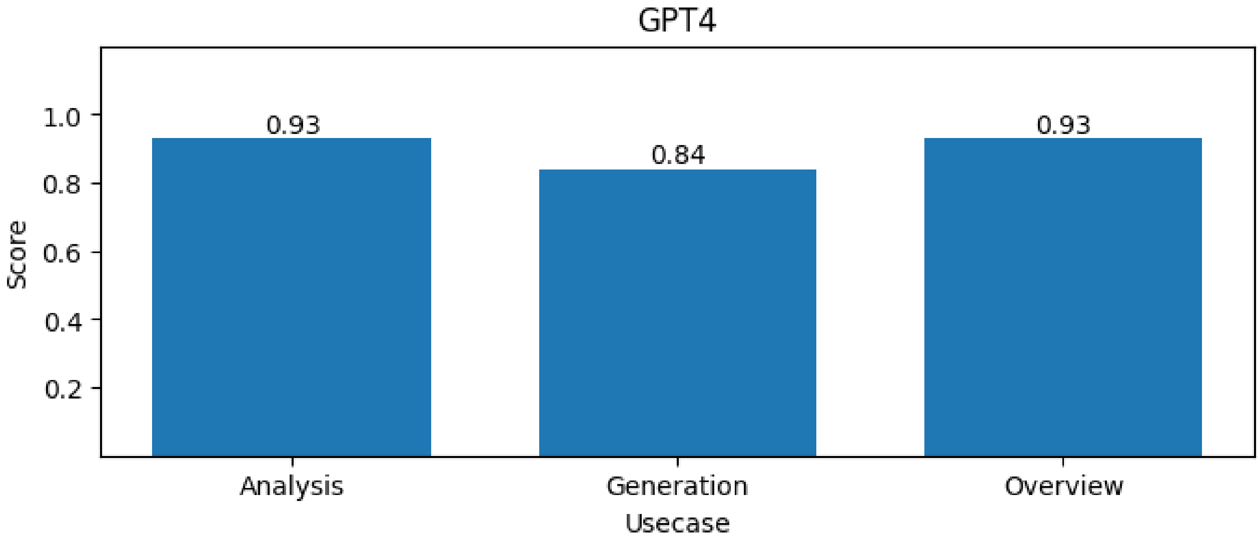


Figure 9: Individual score GPT-4-turbo

6.2.2 Score GPT 3

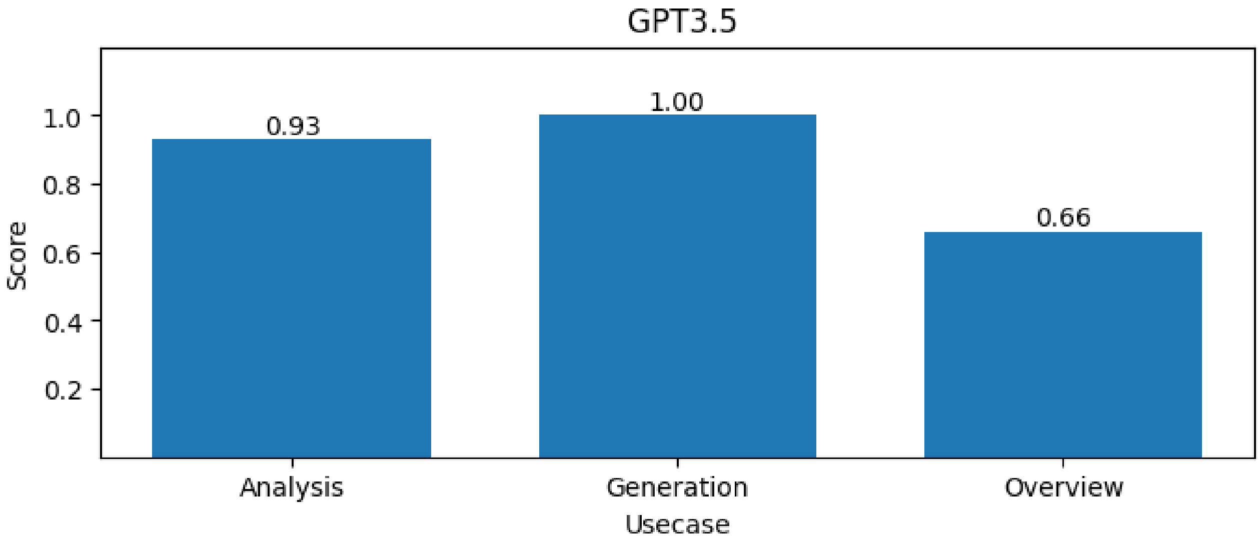


Figure 10: Individual score GPT-3.5-turbo

6.2.3 Score GPT 3.5 Finetuned

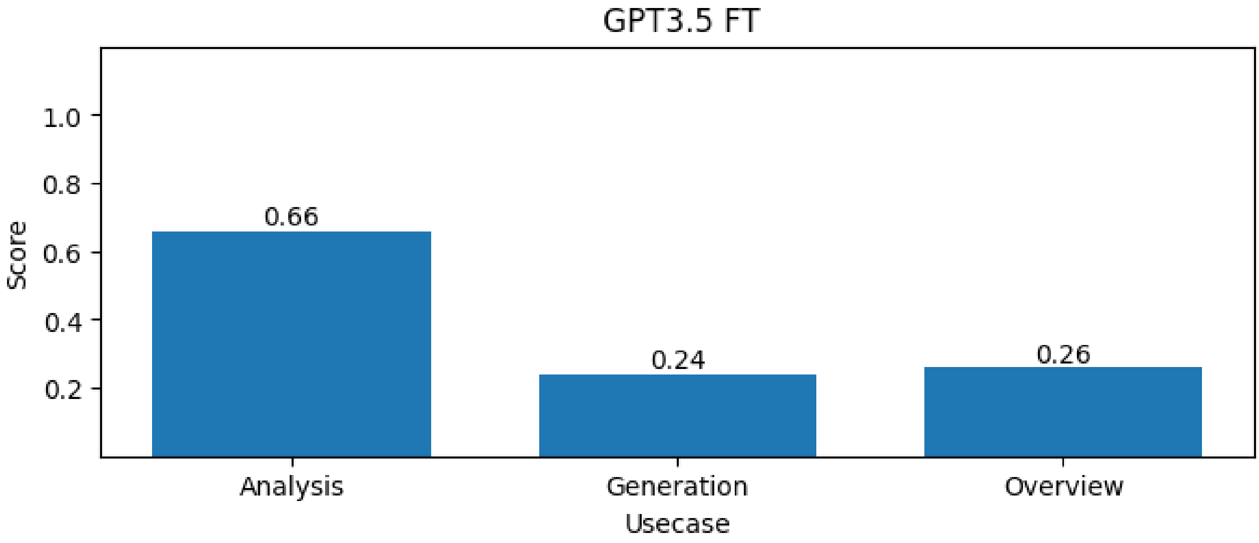


Figure 11: Individual score finetuned GPT-3.5-turbo

6.2.4 Score Mixtral

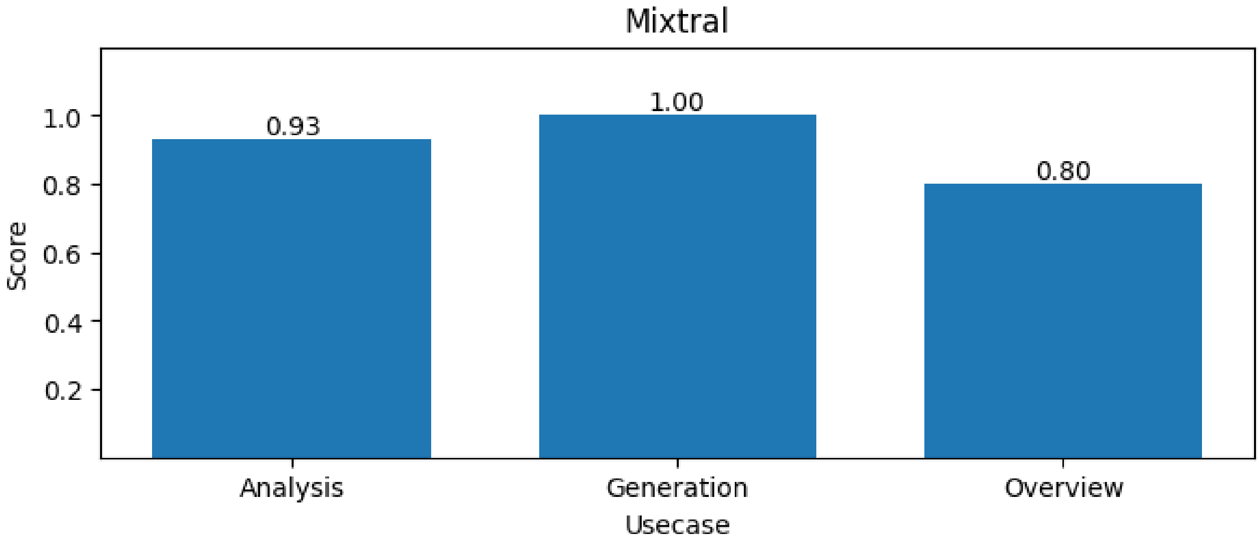


Figure 12: Individual score mistralai/Mixtral-8x7B-Instruct-v0.1

6.2.5 Score GPT 3.5 Finetuned

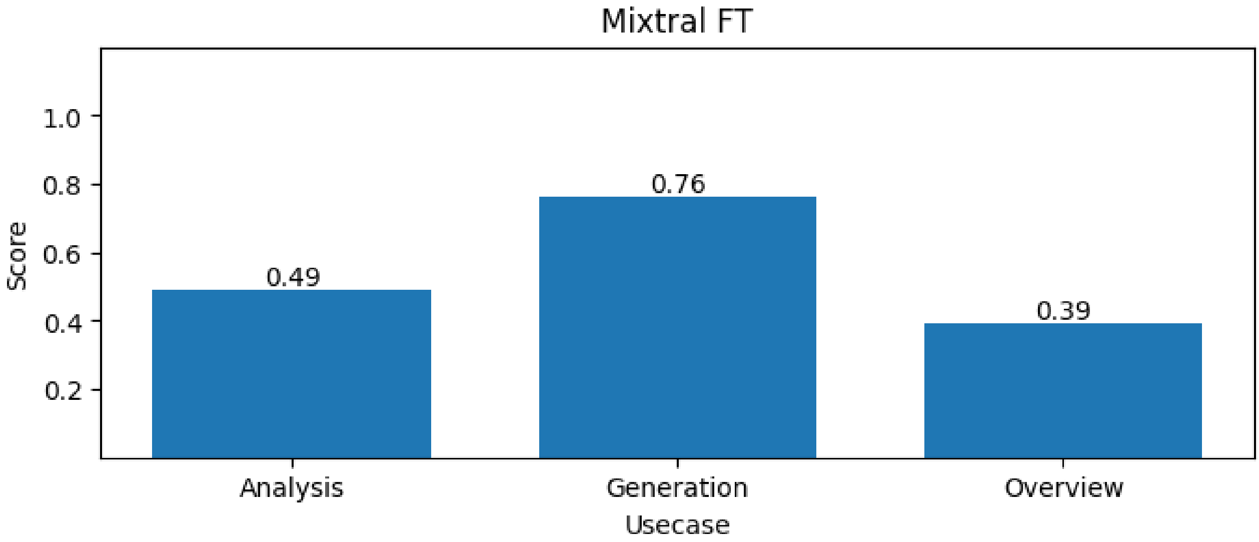


Figure 13: Individual score finetuned mistralai/Mixtral-8x7B-Instruct-v0.1

6.3 Textresults

All responses can be found in the attachments (11.1). The criterion by criterion evaluation can be found in the Jupyter Notebooks.

6.4 Insights

6.4.1 General Performance

Excluding the finetuned models all models were able to perform well on the given tasks with only GPT-3.5 on the task overview performing below 80%. In general, Mixtral and GPT-4 seemed to provide more detailed responses and context. This can be seen in the following example (generated question, use case 1):

GPT-3.5 *Beurteilen Sie die Bedeutung der Mindestanzahl an Unterzeichnenden für Wahlvorschläge.*
GPT-4 *Analysieren Sie die Rolle von Volksmehr und Ständemehr bei der Annahme von Verfassungsänderungen in einer föderalistischen Struktur.*
Mixtral *Beschreibe das "Two-round system" (absolutes Mehrheitswahlrecht in Einerwahlkreisen) und gib ein Beispiel an, in dem dieses System verwendet wird.*

GPT-3.5 tended to use more straight forward questions, while GPT 4 used extra details to broaden the question a bit which is fine but not completly necessary given the context ("in einer föderalistischen Struktur" with the context of the lecture being 'Political rights in Switzerland').
This lead to some questions that were not intuitive for example:

GPT-4 *Bewerten Sie die Bedeutung der geheimen Wahl im Kontext der Einheit der Materie und der korrekten Information der Stimmberechtigten.*

Here the correct answer would be that there is no significance to the given context. While technically a legitimate question, questions like this might lead to more confusion with the students.

6.4.2 Finetuning vs. Prompting

One of the big issues with the finetuned model was the large discrepancy of the number of finetuning datapoints to the amount of possible and wished for outputs. It can be seen that the models tend to stick with the structure of the finetuned training. This leads to less useful results especially for Use Case 3: Comprehension Evaluation where the nuances of how a student shows knowledge gaps gains more importance rather than just factual correctness.

In general, pretrained large language models such as Mixtral and ChatGPT offer a robust and highly usable base for user interactions right out of the box. These models are designed to understand and generate human-like text based on extensive pretraining on diverse datasets. This enables them to interpret and respond to prompts effectively without the need for further fine-tuning.

One of the key advantages of using these pretrained models is their ability to work with provided textual context through well-crafted prompts. By providing clear and specific instructions in the systemprompt, the model's capabilities can be leveraged to perform a wide range of tasks, including the given subtasks in this project.

In contrast, fine-tuning these models with a limited dataset can often lead to overfitting. Overfitting occurs when a model becomes too tailored to the specific examples in the fine-tuning dataset, resulting in a loss of generalization and flexibility. This can enforce a style or approach that is overly rigid, diminishing the model's versatility and its ability to handle diverse or novel inputs.

Therefore, employing proper system prompts is a more effective strategy for utilizing large language models. Prompts allow users to guide the model's behavior dynamically, ensuring that it can adapt to various contexts and requirements without the drawbacks associated with overfitting. This approach preserves the inherent strengths of the pretrained models, maintaining their broad applicability and responsiveness across a wide range of tasks and interactions.

6.5 Mixtral vs. GPT-4

On the given tasks both models performed well, with GPT-4 having a tendency to ask in a very broad manner but performing better in the overview subtask, being able to better sugest other parts of the document which have not yet been covered with the identical prompt. However this could be further improved for Mixtral by providing an example of a structure for a gold standard response in the system prompt. Based on response quality, either model could be recommended to be used.

6.6 Token Usage & Cost

A brief introduction to tokens:
Tokens are the basic units of text that language models use to process and generate language.
A token can be as short as one character (like "a" or ",") or as long as one word (like "apple") or even a part of a word (like "un-", "-ing"). Tokens are the pieces of text that the model breaks down to analyze and generate responses.
Language models process and generate text one token at a time. For instance, when generating text, the model predicts the next token based on the tokens it has seen so far. This sequential processing allows the model to construct coherent and contextually relevant sentences.

The number of tokens can affect the performance and efficiency of the model. For example, longer texts with more tokens require more computational resources to process. Additionally, most models have a limit on the number of tokens they can handle in a single input, which can impact how much text can be processed at once.

In usage-based pricing models (like the one employed by OpenAI), the cost is typically based on the number of tokens processed. This means that more verbose inputs or outputs can be more expensive to process compared to shorter ones.

According to OpenAi[34] the following values can be used as a rule of thumb:

$$\text{1 token} \approx \text{4 chars in English}$$

$$\text{1 token} \approx \frac{3}{4}\text{ words}$$

$$\text{100 tokens} \approx \text{75 words}$$

$$\text{1-2 sentence} \approx \text{30 tokens}$$

$$\text{1 paragraph} \approx \text{100 tokens}$$

$$\text{1,500 words} \approx \text{2048 tokens}$$

The provided lecture script contained 5295 words (34'345 characters) which translates to roughly 7000-8000 tokens. Using the file-search tool with OpenAIs Assitants cost between 7000 and 15000 tokens, however after loading the files from the vector store, the contents could be used with that assistant without loading them again unless data was required that was not looked at before.

A 'Use Case 3: Comprehension Evaluation' response after loading the files used around 1000-3000 tokens.

According to OpenAi's pricing page[35] and detailed pricing in regards to GPT-4[36] where prices lie at 0.0005$ per 1000 tokens input and 0.0015$ per 1000 tokens output for GPT-3.5-turbo and 0.01$ / 0.03$ for GPT-4 would result in an expected costs of 0.0035$ with GPT-3.5-turbo and 0.08$ with GPT-4.

---

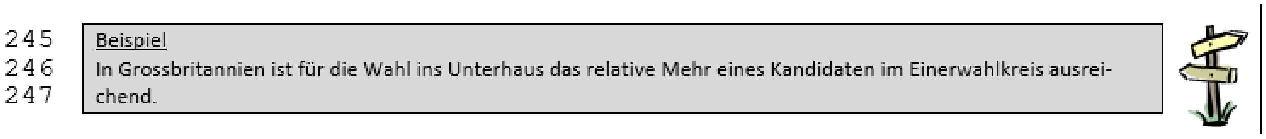
[34] https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them

[35] https://openai.com/api/pricing/

[36] https://help.openai.com/en/articles/7127956-how-much-does-gpt-4-cost

6.7 Image Handling

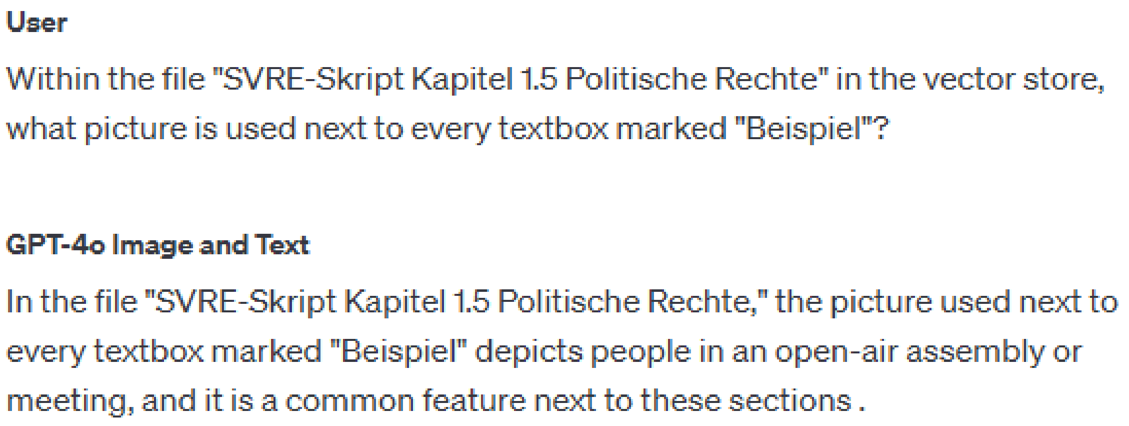
Trying to refer to images within the text does currently not work by itself using these chat models directly.



Beispiel

In Grossbritannien ist für die Wahl ins Unterhaus das relative Mehr eines Kandidaten im Einerwahlkreis ausreichend.

Figure 14: Example of a reoccurring picture element throughout the lecture script: a signpost next to an example box



User

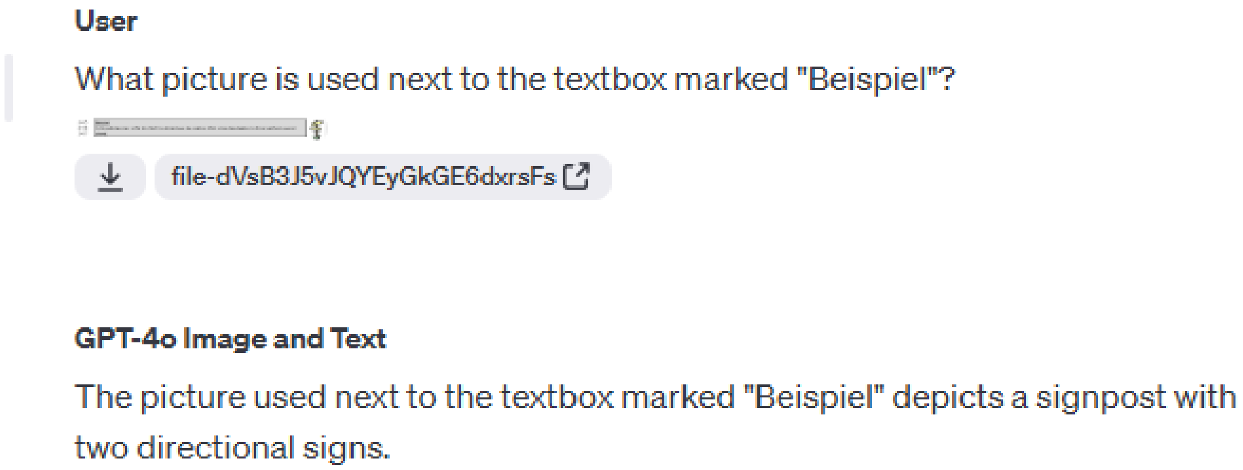
Within the file "SVRE-Skript Kapitel 1.5 Politische Rechte" in the vector store, what picture is used next to every textbox marked "Beispiel"?

GPT-4o Image and Text

In the file "SVRE-Skript Kapitel 1.5 Politische Rechte," the picture used next to every textbox marked "Beispiel" depicts people in an open-air assembly or meeting, and it is a common feature next to these sections .

Figure 15: Response of GPT-4o when asked what image there is.

While GPT-4o can be used to work on images as input, they must be provided separately as images. When the image is provided separately GPT-4o can recognise it. Here the model was asked about the image from figure 14.



User

What picture is used next to the textbox marked "Beispiel"?

GPT-4o Image and Text

The picture used next to the textbox marked "Beispiel" depicts a signpost with two directional signs.

Figure 16: Response of GPT-4o when asked what image there is next to the textbox.

7 Prototype Application

7.1 Requirements

The prototype application is supposed to fulfill the following formal requirements based on the task given to us:

- Creating questions from a lesson script or predefined topic.
- Evaluating the answers on correctness and providing interactive feedback.
- Evaluating the student's comprehension of a certain topic and pose appropriate follow up questions.

Upon further evaluation, we devised the following additional requirements:

- Configuration of chat bots within the web user interface.
- Support for locally hosted LLMs to allow the use of confidential data to enhance the chat bots.
- The data structure must reflect the real life scenario of a lecturer preparing a chat bot for the students to prepare with, and the students having access to a range of chat bots from multiple courses.

7.2 Architecture Documentation Style

7.2.1 Final Prototype Architecture

The final architecture is documented with the following visualizations:

- A C4[37] Container Diagram
- C4 Component Diagrams
- Backend Logical Layers: Overview
- Entity Relationship Model
- C4 Dynamic Diagrams for the most important Use Cases

We refrain from providing C4 code diagrams for the implemented classes due to their fast-changing nature and will not provide a C4 system context diagram, as no information would be provided that can not be derived from the C4 container diagram. Instead, we provide an overview over each logical layer of our API as a replacement for the C4 code diagrams. This overview provides a brief description of the involved classes, their relationships and their responsibilities.

A detailed REST API documentation for our API is not provided in this document. A live version, powered by Swagger[38], can be found on a running instance by accessing the API root URL (Default: http://localhost:8000) with a modern browser. We also provide the OpenAPI[39] specification as a separate JSON file.

---

[37] https://c4model.com/
[38] https://swagger.io/
[39] https://www.openapis.org/

7.2.2 Architecture Decision Records

A multitude of architectural decisions were made over the course of this project. As some of them are not reflected in the final architecture documentation, we are providing them in a separate chapter. Each architecture decision record is designed to fit a single A4 page and is presented in the following format:

ADR#	ADR000
Author	<Author Name>
State	<Proposed|Accepted|Rejected>
Related	<Section Name> (p. <Page Number>) <Section Name> (p. <Page Number>)

Decision Record
<A detailed description of the decision made, as a Y-Statement>
In the context of <use case|component>
facing <problem to solve>
we decided <decision>
and neglected <other options>
to achieve <upsides>
accepting <downsides>

(Optional) Images
<Images or diagrams that visualize the decision>

(Optional) Comments
<Justifications or comments if a single Y statement is too limiting>

7.3 Final Prototype Architecture

7.3.1 C4 Container Diagram

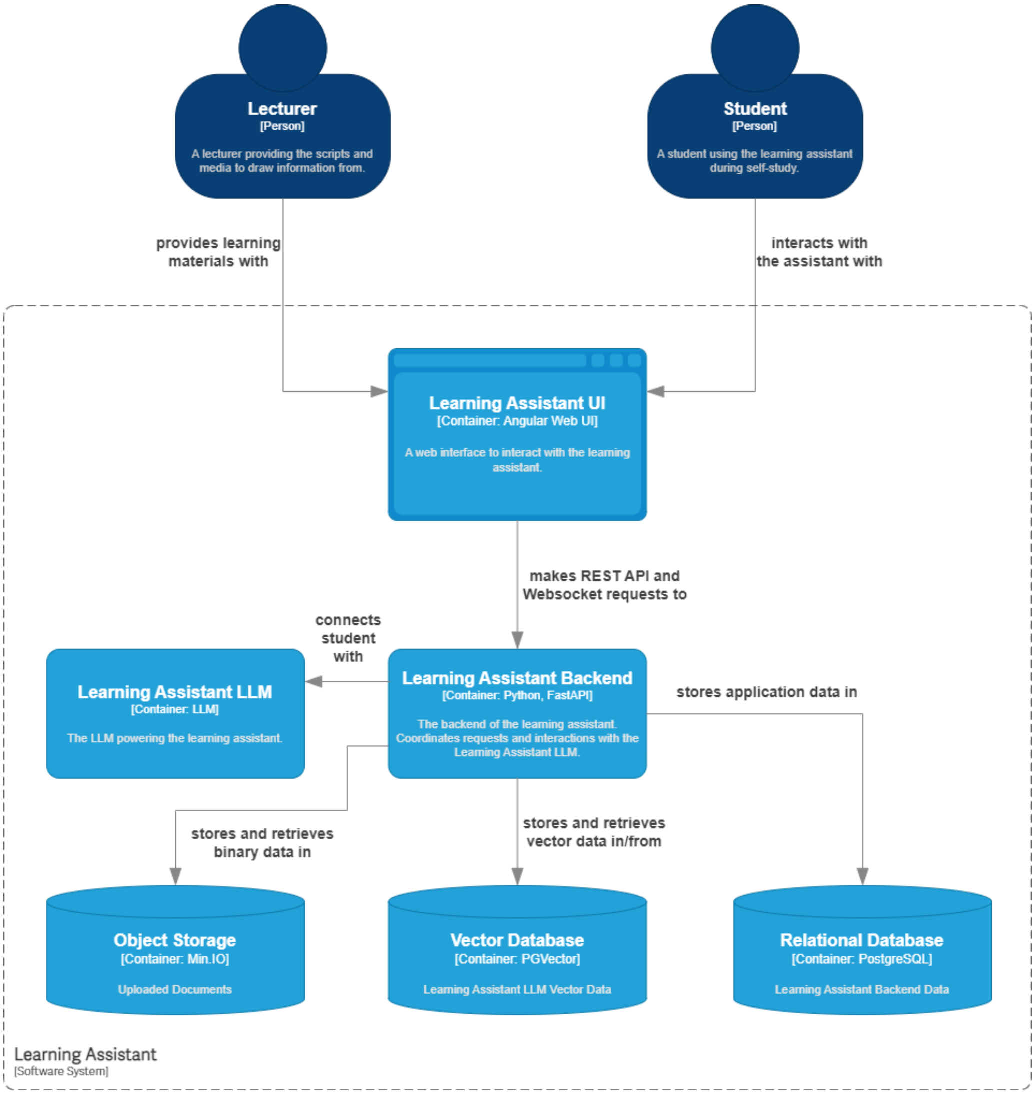

Lecturer
[Person]
A lecturer providing the scripts and media to draw information from.
Student
[Person]
A student using the learning assistant during self-study.
provides learning materials with
interacts with the assistant with
Learning Assistant UI
[Container: Angular Web UI]
A web interface to interact with the learning assistant.
makes REST API and Websocket requests to
connects student with
Learning Assistant LLM
[Container: LLM]
The LLM powering the learning assistant.
Learning Assistant Backend
[Container: Python, FastAPI]
The backend of the learning assistant. Coordinates requests and interactions with the Learning Assistant LLM.
stores application data in
stores and retrieves binary data in
stores and retrieves vector data in/from
Object Storage
[Container: Min.IO]
Uploaded Documents
Vector Database
[Container: PGVector]
Learning Assistant LLM Vector Data
Relational Database
[Container: PostgreSQL]
Learning Assistant Backend Data
Learning Assistant
[Software System]

Figure 17: Container Diagram

7.3.2 C4 Component Diagrams

Component Diagram: API

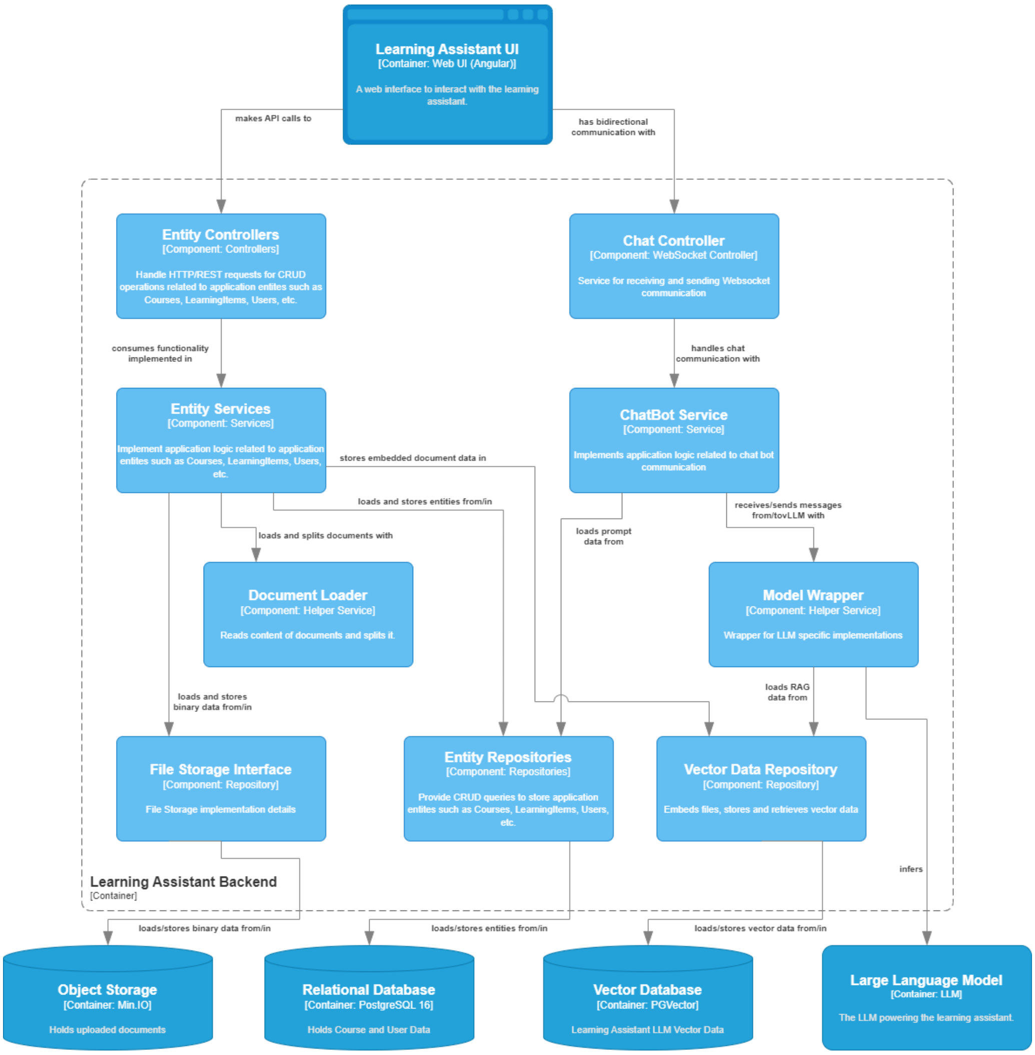

Learning Assistant UI
[Container: Web UI (Angular)]
A web interface to interact with the learning assistant.
makes API calls to
has bidirectional communication with
Entity Controllers
[Component: Controllers]
Handle HTTP/REST requests for CRUD operations related to application entites such as Courses, LearningItems, Users, etc.
Chat Controller
[Component: WebSocket Controller]
Service for receiving and sending Websocket communication
consumes functionality implemented in
handles chat communication with
Entity Services
[Component: Services]
Implement application logic related to application entites such as Courses, LearningItems, Users, etc.
stores embedded document data in
ChatBot Service
[Component: Service]
Implements application logic related to chat bot communication
loads and stores entities from/in
loads and splits documents with
loads prompt data from
receives/sends messages from/tovLLM with
Document Loader
[Component: Helper Service]
Reads content of documents and splits it.
Model Wrapper
[Component: Helper Service]
Wrapper for LLM specific implementations
loads and stores binary data from/in
loads RAG data from
File Storage Interface
[Component: Repository]
File Storage implementation details
Entity Repositories
[Component: Repositories]
Provide CRUD queries to store application entites such as Courses, LearningItems, Users, etc.
Vector Data Repository
[Component: Repository]
Embeds files, stores and retrieves vector data
infers
Learning Assistant Backend
[Container]
loads/stores binary data from/in
loads/stores entities from/in
loads/stores vector data from/in
Object Storage
[Container: Min.IO]
Holds uploaded documents
Relational Database
[Container: PostgreSQL 16]
Holds Course and User Data
Vector Database
[Container: PGVector]
Learning Assistant LLM Vector Data
Large Language Model
[Container: LLM]
The LLM powering the learning assistant.

Figure 18: Simplified Web API component diagram. Configuration Service, Database Engine and Seeding Data have been omitted and individual Entity Controllers, Services and Repositories have been merged into single components for the sake of readability.

Component Diagram: Angular Web UI

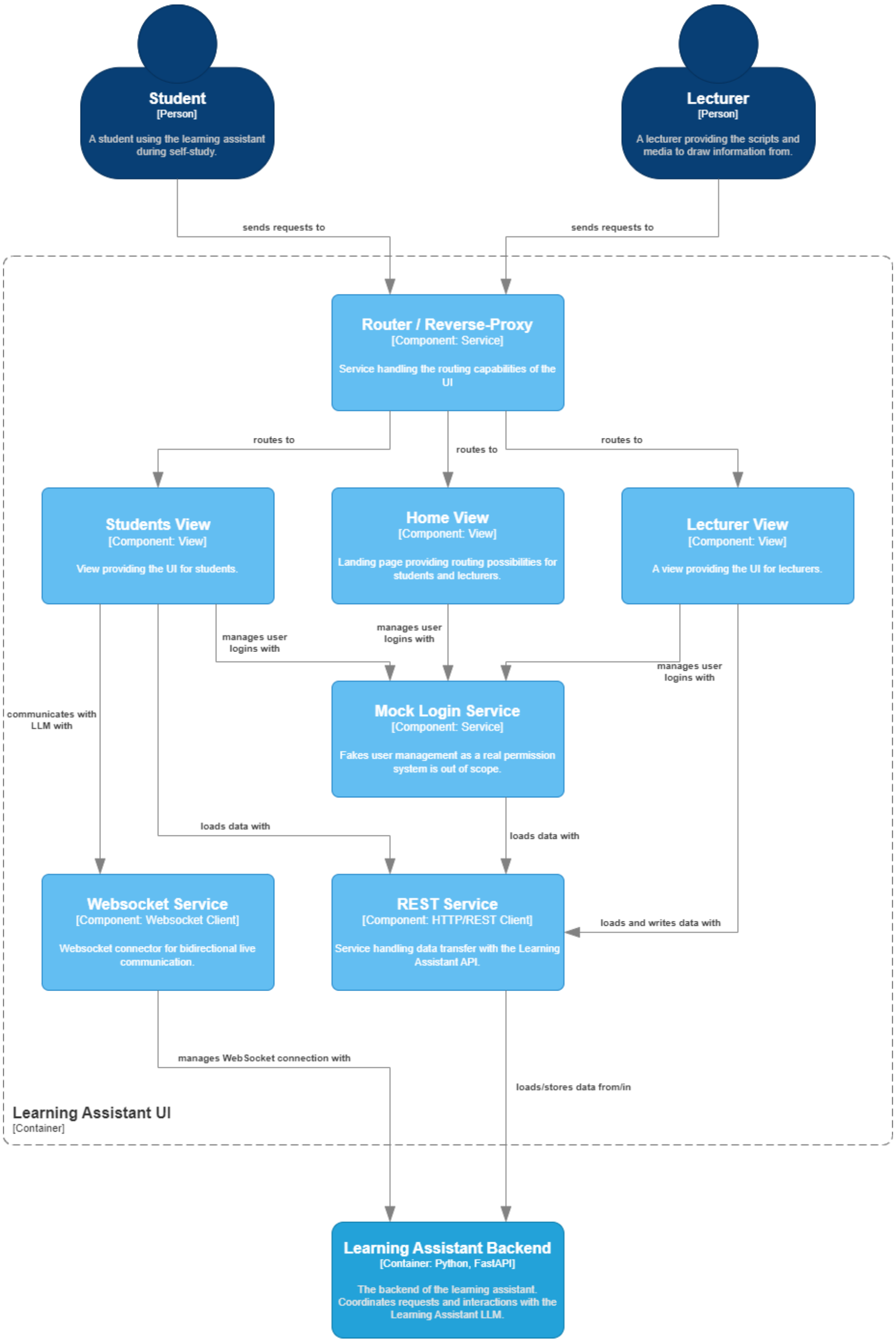
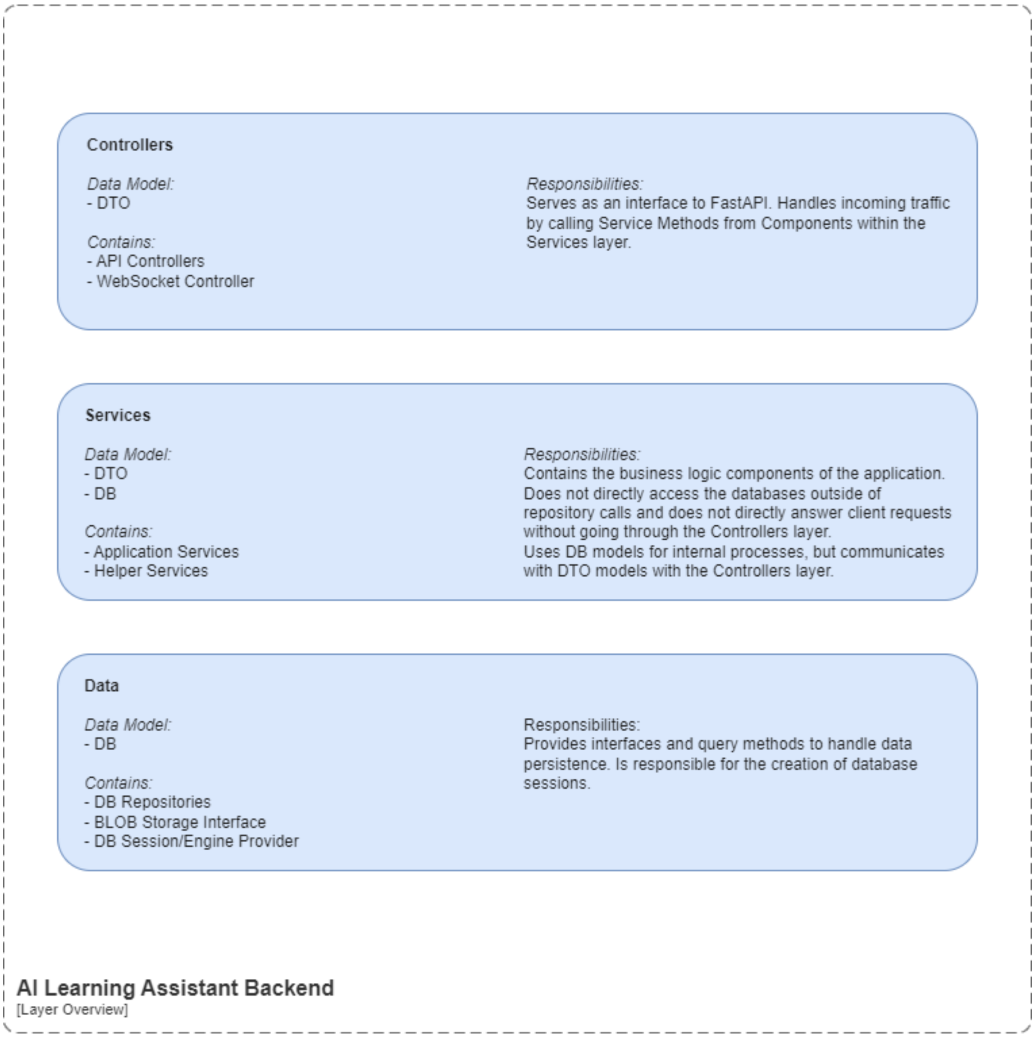

Student
[Person]
A student using the learning assistant during self-study.
Lecturer
[Person]
A lecturer providing the scripts and media to draw information from.
sends requests to
sends requests to
Router / Reverse-Proxy
[Component: Service]
Service handling the routing capabilities of the UI
routes to
routes to
routes to
Students View
[Component: View]
View providing the UI for students.
Home View
[Component: View]
Landing page providing routing possibilities for students and lecturers.
Lecturer View
[Component: View]
A view providing the UI for lecturers.
manages user logins with
manages user logins with
manages user logins with
communicates with LLM with
Mock Login Service
[Component: Service]
Fakes user management as a real permission system is out of scope.
loads data with
loads data with
loads and writes data with
Websocket Service
[Component: Websocket Client]
Websocket connector for bidirectional live communication.
REST Service
[Component: HTTP/REST Client]
Service handling data transfer with the Learning Assistant API.
manages WebSocket connection with
loads/stores data from/in
Learning Assistant UI
[Container]
Learning Assistant Backend
[Container: Python, FastAPI]
The backend of the learning assistant. Coordinates requests and interactions with the Learning Assistant LLM.

Figure 19: Component Diagram of the Web Application

7.3.3 Web API Logical Layers: Overview



Controllers

Data Model:
- DTO

Contains:
- API Controllers
- WebSocket Controller

Responsibilities:
Serves as an interface to FastAPI. Handles incoming traffic by calling Service Methods from Components within the Services layer.

Services

Data Model:
- DTO
- DB

Contains:
- Application Services
- Helper Services

Responsibilities:
Contains the business logic components of the application. Does not directly access the databases outside of repository calls and does not directly answer client requests without going through the Controllers layer.
Uses DB models for internal processes, but communicates with DTO models with the Controllers layer.

Data

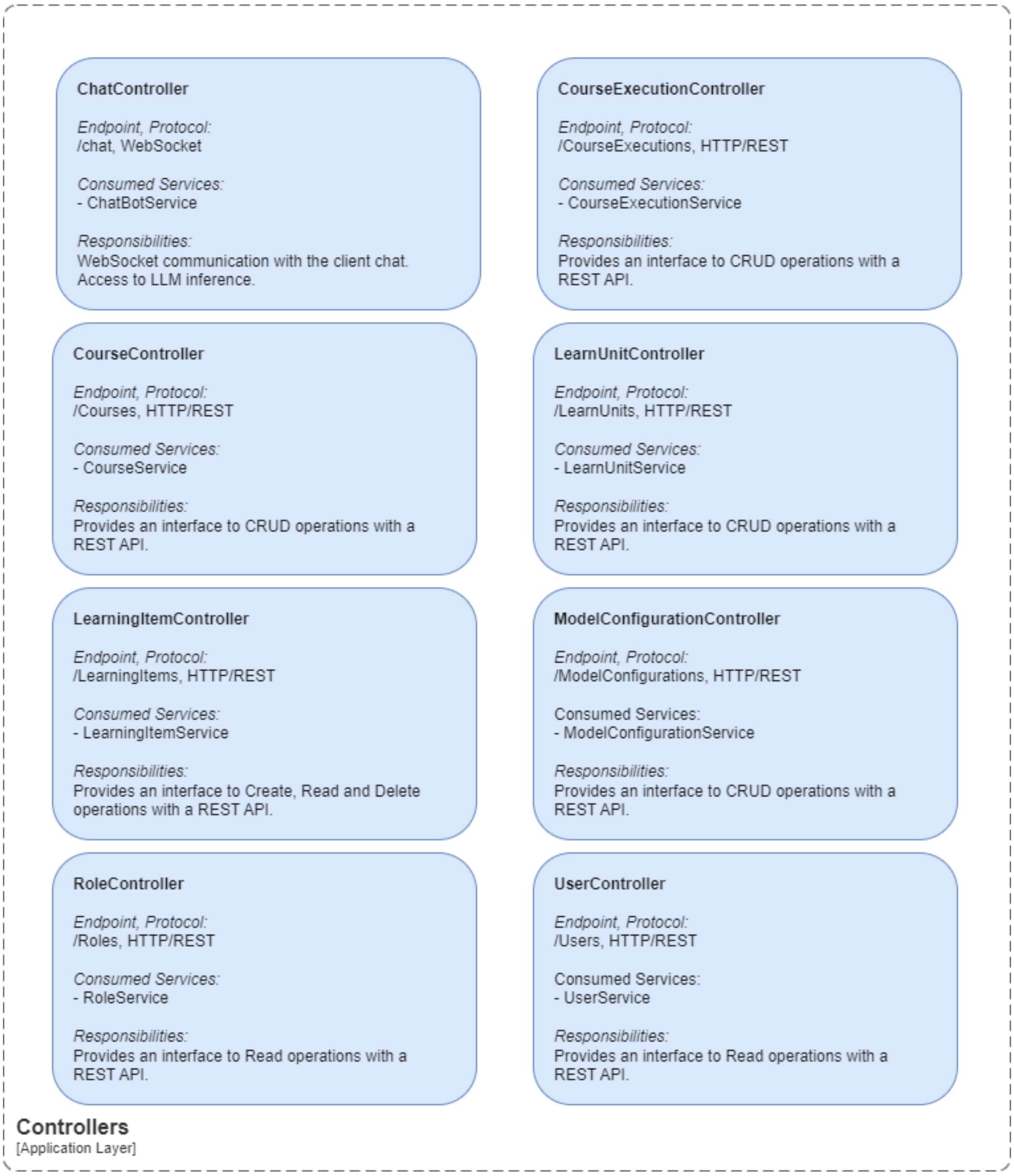
Data Model:
- DB

Contains:
- DB Repositories
- BLOB Storage Interface
- DB Session/Engine Provider

Responsibilities:
Provides interfaces and query methods to handle data persistence. Is responsible for the creation of database sessions.

Figure 20: An overview over the logical layers of the Web API application

7.3.4 Web API Logical Layers: Controllers



ChatController

Endpoint, Protocol:
/chat, WebSocket

Consumed Services:
- ChatBotService

Responsibilities:
WebSocket communication with the client chat.
Access to LLM inference.

CourseExecutionController

Endpoint, Protocol:
/CourseExecutions, HTTP/REST

Consumed Services:
- CourseExecutionService

Responsibilities:
Provides an interface to CRUD operations with a REST API.

CourseController

Endpoint, Protocol:
/Courses, HTTP/REST

Consumed Services:
- CourseService

Responsibilities:
Provides an interface to CRUD operations with a REST API.

LearnUnitController

Endpoint, Protocol:
/LearnUnits, HTTP/REST

Consumed Services:
- LearnUnitService

Responsibilities:
Provides an interface to CRUD operations with a REST API.

LearningItemController

Endpoint, Protocol:
/LearningItems, HTTP/REST

Consumed Services:
- LearningItemService

Responsibilities:
Provides an interface to Create, Read and Delete operations with a REST API.

ModelConfigurationController

Endpoint, Protocol:
/ModelConfigurations, HTTP/REST

Consumed Services:
- ModelConfigurationService

Responsibilities:
Provides an interface to CRUD operations with a REST API.

RoleController

Endpoint, Protocol:
/Roles, HTTP/REST

Consumed Services:
- RoleService

Responsibilities:
Provides an interface to Read operations with a REST API.

UserController

Endpoint, Protocol:
/Users, HTTP/REST

Consumed Services:
- UserService

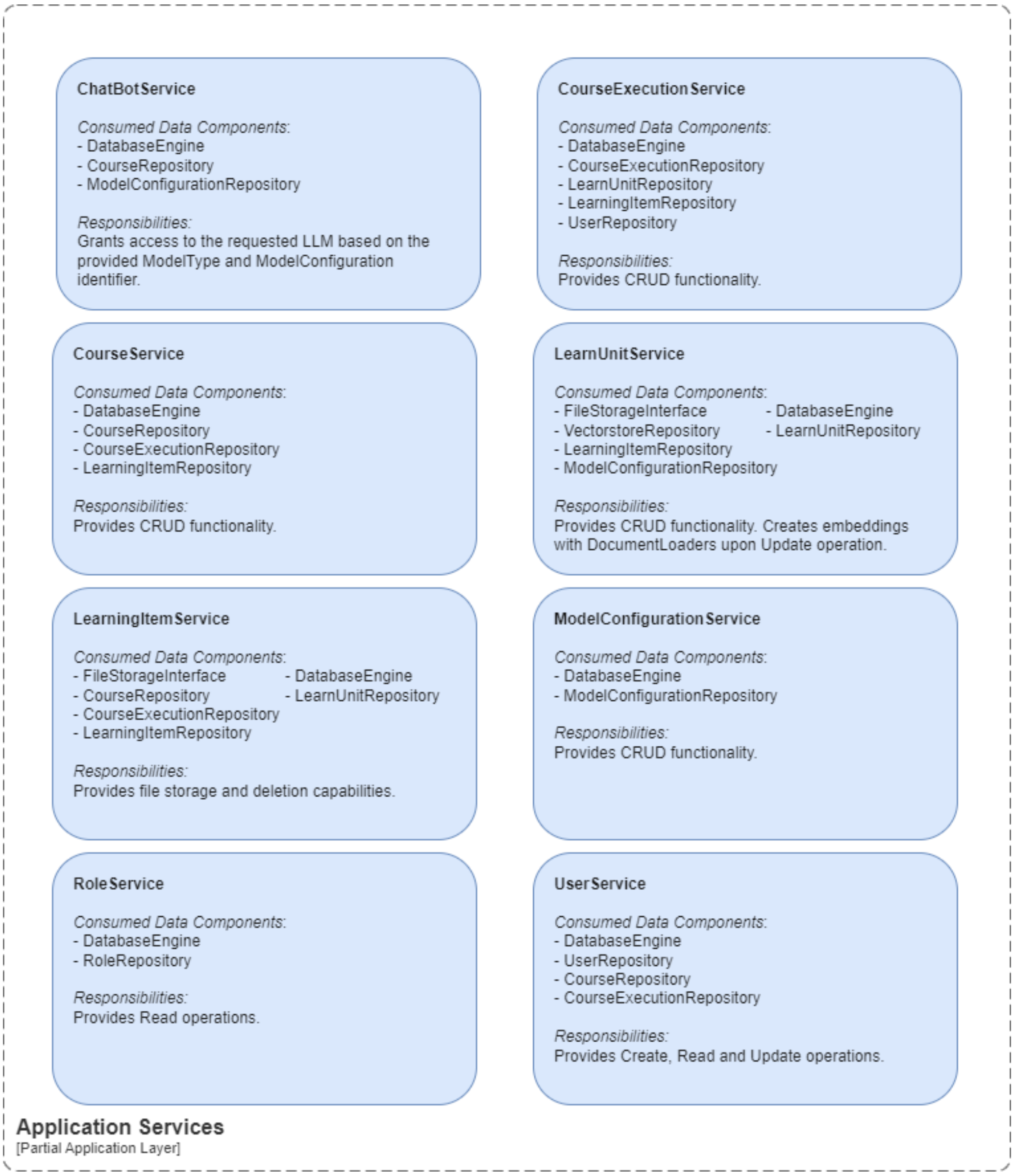
Responsibilities:
Provides an interface to Read operations with a REST API.

Controllers
[Application Layer]

Figure 21: Classes contained in the Controllers layer of the Web API application

7.3.5 Web API Logical Layers: Services

Main Services



ChatBotService

Consumed Data Components:
- DatabaseEngine
- CourseRepository
- ModelConfigurationRepository

Responsibilities:
Grants access to the requested LLM based on the provided ModelType and ModelConfiguration identifier.

CourseExecutionService

Consumed Data Components:
- DatabaseEngine
- CourseExecutionRepository
- LearnUnitRepository
- LearningItemRepository
- UserRepository

Responsibilities:
Provides CRUD functionality.

CourseService

Consumed Data Components:
- DatabaseEngine
- CourseRepository
- CourseExecutionRepository
- LearningItemRepository

Responsibilities:
Provides CRUD functionality.

LearnUnitService

Consumed Data Components:
- FileStorageInterface
- VectorstoreRepository
- LearningItemRepository
- ModelConfigurationRepository
- DatabaseEngine
- LearnUnitRepository

Responsibilities:
Provides CRUD functionality. Creates embeddings with DocumentLoaders upon Update operation.

LearningItemService

Consumed Data Components:
- FileStorageInterface
- CourseRepository
- CourseExecutionRepository
- LearningItemRepository
- DatabaseEngine
- LearnUnitRepository

Responsibilities:
Provides file storage and deletion capabilities.

ModelConfigurationService

Consumed Data Components:
- DatabaseEngine
- ModelConfigurationRepository

Responsibilities:
Provides CRUD functionality.

RoleService

Consumed Data Components:
- DatabaseEngine
- RoleRepository

Responsibilities:
Provides Read operations.

UserService

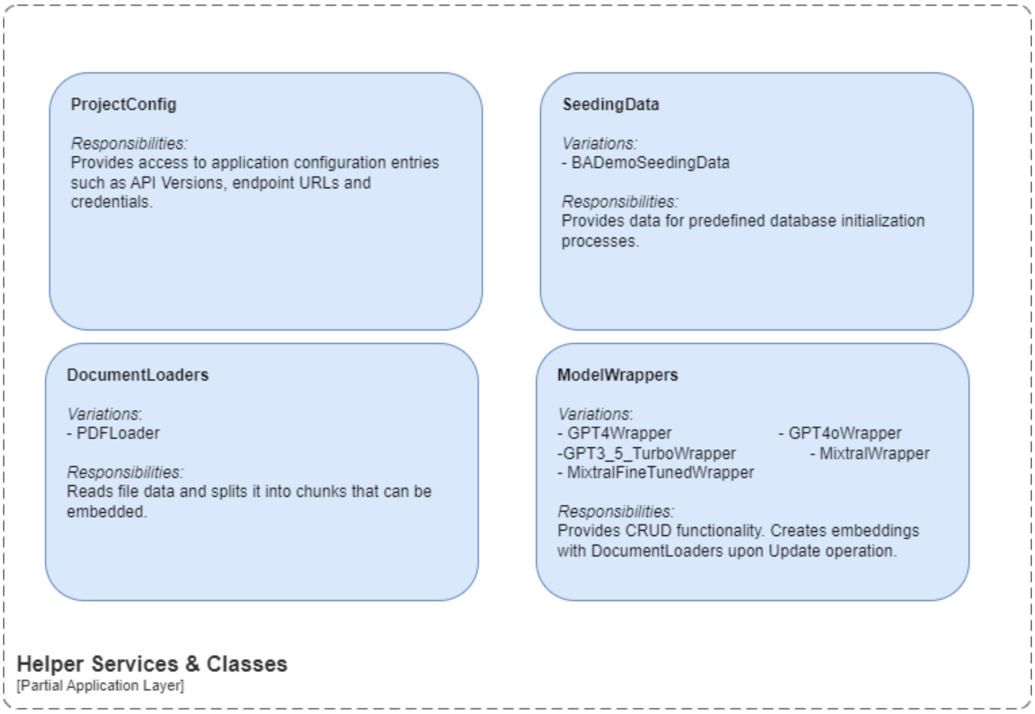
Consumed Data Components:
- DatabaseEngine
- UserRepository
- CourseRepository
- CourseExecutionRepository

Responsibilities:
Provides Create, Read and Update operations.

Application Services
[Partial Application Layer]

Figure 22: Primary service classes contained in the Service layer of the Web API application

Helper Classes



ProjectConfig

Responsibilities:
Provides access to application configuration entries such as API Versions, endpoint URLs and credentials.

SeedingData

Variations:
- BADemoSeedingData

Responsibilities:
Provides data for predefined database initialization processes.

DocumentLoaders

Variations:
- PDFLoader

Responsibilities:
Reads file data and splits it into chunks that can be embedded.

ModelWrappers

Variations:
- GPT4Wrapper
-GPT3_5_TurboWrapper
- MixtralFineTunedWrapper
- GPT4oWrapper
- MixtralWrapper

Responsibilities:
Provides CRUD functionality. Creates embeddings with DocumentLoaders upon Update operation.

Figure 23: Helper classes contained in the Service layer of the Web API application

7.3.6 Web API Logical Layers: Data

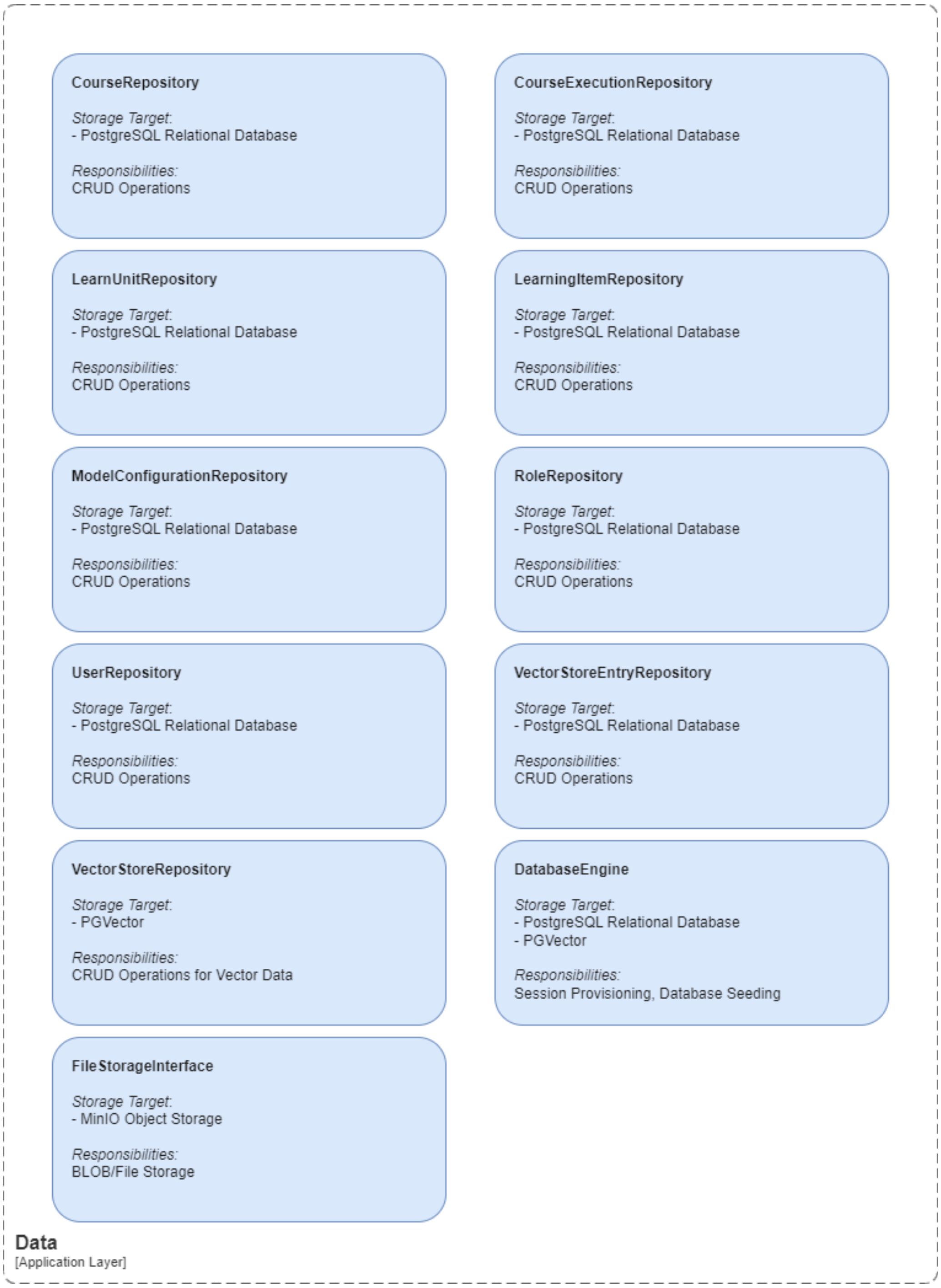


Figure 24: Classes contained in the Data layer of the Web API application

7.3.7 Application Database: Entity Relationship Model

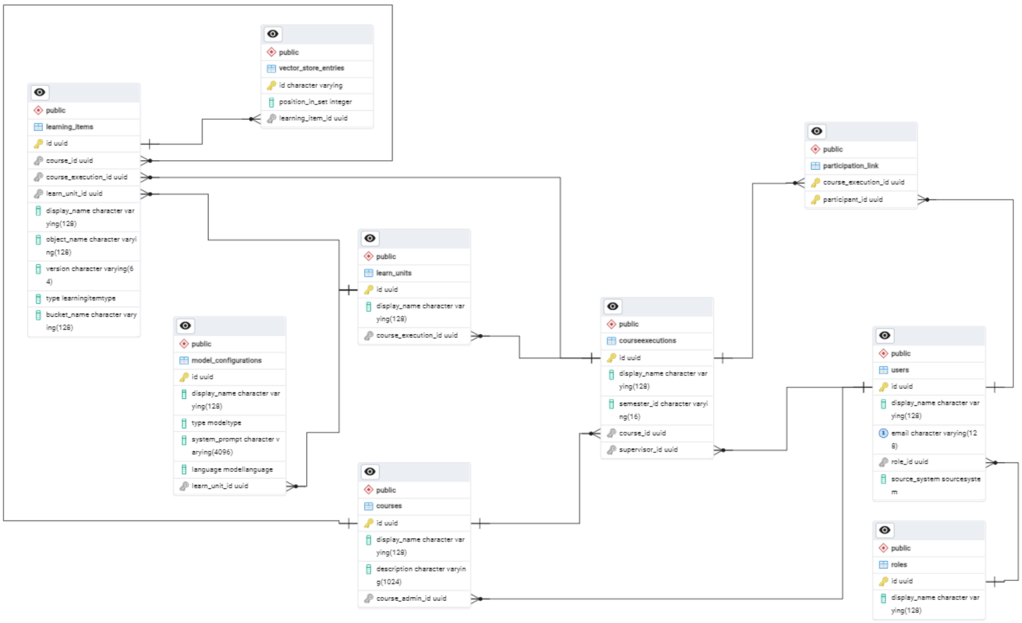


Figure 25: Entity Relationship Model of the Application Database

7.3.8 C4 Dynamic Diagrams

Model Inference: User answers a question by the AI Teacher's assistant

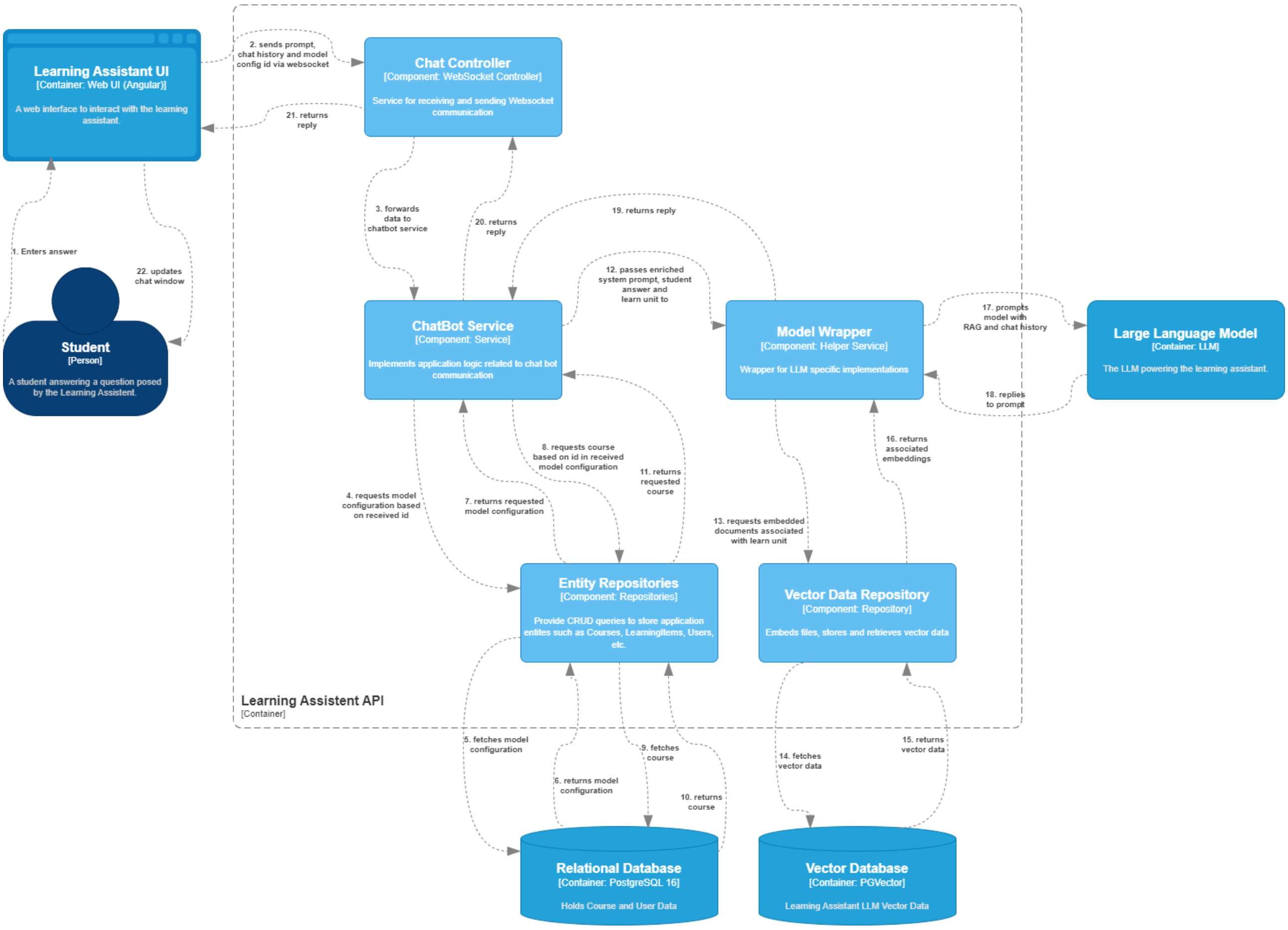

Learning Assistant UI
[Container: Web UI (Angular)]
A web interface to interact with the learning assistant.
2. sends prompt, chat history and model config id via websocket
Chat Controller
[Component: WebSocket Controller]
Service for receiving and sending Websocket communication
21. returns reply
3. forwards data to chatbot service
20. returns reply
19. returns reply
1. Enters answer
22. updates chat window
Student
[Person]
A student answering a question posed by the Learning Assistent.
12. passes enriched system prompt, student answer and learn unit to
ChatBot Service
[Component: Service]
Implements application logic related to chat bot communication
17. prompts model with RAG and chat history
Model Wrapper
[Component: Helper Service]
Wrapper for LLM specific implementations
Large Language Model
[Container: LLM]
The LLM powering the learning assistant.
18. replies to prompt
8. requests course based on id in received model configuration
16. returns associated embeddings
4. requests model configuration based on received id
7. returns requested model configuration
11. returns requested course
13. requests embedded documents associated with learn unit
Entity Repositories
[Component: Repositories]
Provide CRUD queries to store application entites such as Courses, LearningItems, Users, etc.
Vector Data Repository
[Component: Repository]
Embeds files, stores and retrieves vector data
Learning Assistant API
[Container]
5. fetches model configuration
9. fetches course
6. returns model configuration
10. returns course
14. fetches vector data
15. returns vector data
Relational Database
[Container: PostgreSQL 16]
Holds Course and User Data
Vector Database
[Container: PGVector]
Learning Assistant LLM Vector Data

Figure 26: Dynamic diagram illustrating the path the request takes through the application

Document Embedding: User saves a learn unit with associated documents

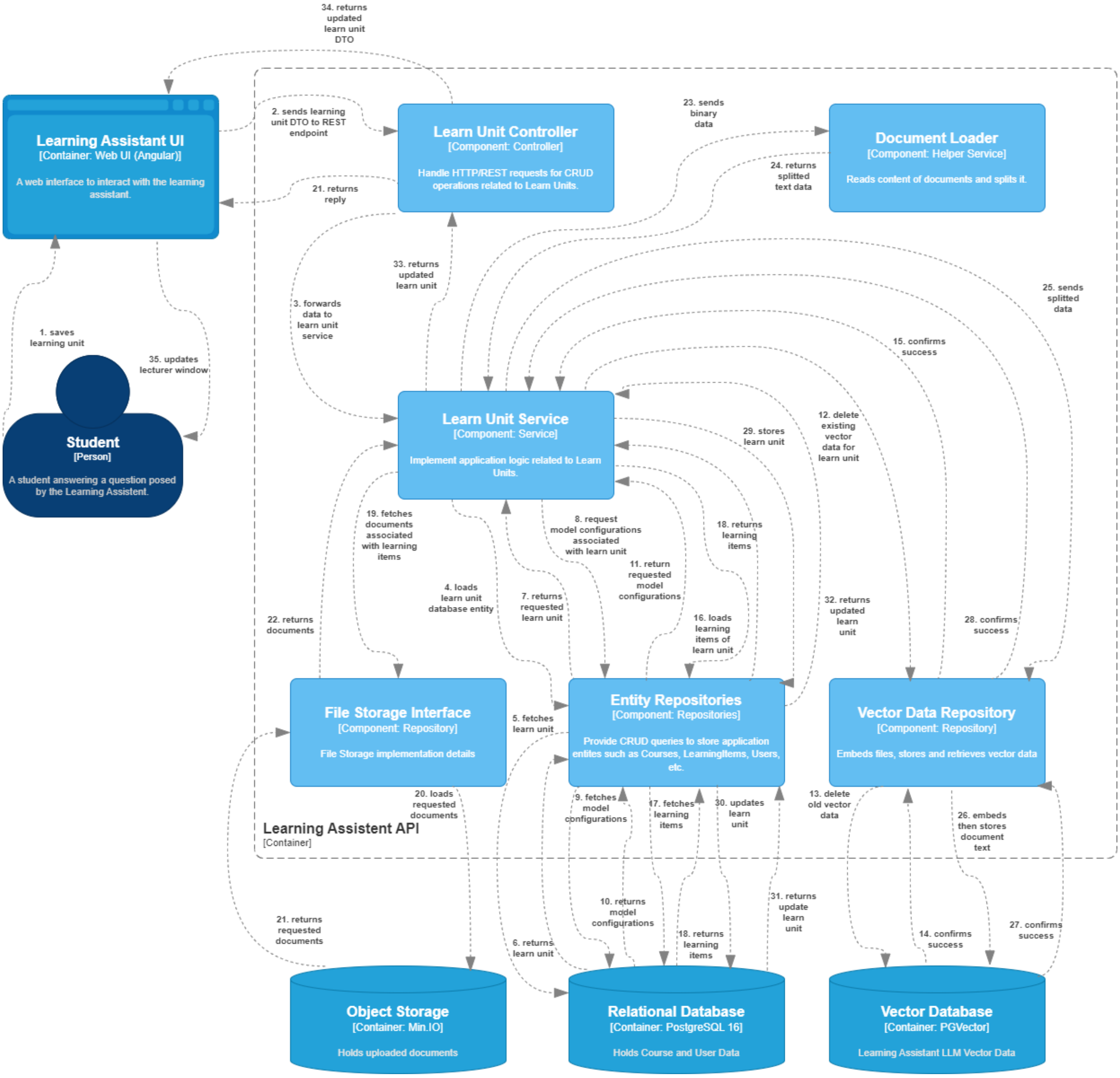

34. returns updated learn unit DTO
Learning Assistant UI
[Container: Web UI (Angular)]
A web interface to interact with the learning assistant.
2. sends learning unit DTO to REST endpoint
21. returns reply
Learn Unit Controller
[Component: Controller]
Handle HTTP/REST requests for CRUD operations related to Learn Units.
23. sends binary data
24. returns splitted text data
Document Loader
[Component: Helper Service]
Reads content of documents and splits it.
1. saves learning unit
35. updates lecturer window
Student
[Person]
A student answering a question posed by the Learning Assistent.
33. returns updated learn unit
3. forwards data to learn unit service
25. sends splitted data
15. confirms success
Learn Unit Service
[Component: Service]
Implement application logic related to Learn Units.
29. stores learn unit
12. delete existing vector data for learn unit
19. fetches documents associated with learning items
8. request model configurations associated with learn unit
18. returns learning items
11. return requested model configurations
4. loads learn unit database entity
7. returns requested learn unit
16. loads learning items of learn unit
22. returns documents
32. returns updated learn unit
28. confirms success
File Storage Interface
[Component: Repository]
File Storage implementation details
5. fetches learn unit
Entity Repositories
[Component: Repositories]
Provide CRUD queries to store application entites such as Courses, LearningItems, Users, etc.
Vector Data Repository
[Component: Repository]
Embeds files, stores and retrieves vector data
20. loads requested documents
9. fetches model configurations
17. fetches learning items
30. updates learn unit
13. delete old vector data
26. embeds then stores document text
Learning Assistant API
[Container]
21. returns requested documents
10. returns model configurations
6. returns learn unit
18. returns learning items
31. returns update learn unit
14. confirms success
27. confirms success
Object Storage
[Container: Min.IO]
Holds uploaded documents
Relational Database
[Container: PostgreSQL 16]
Holds Course and User Data
Vector Database
[Container: PGVector]
Learning Assistant LLM Vector Data

Figure 27: Dynamic diagram illustrating the path the request takes through the application

7.4 Architecture Decision Records (ADRs)

7.4.1 ADR001: Initial Application Data Model

ADR#	ADR001
Author	Dominik Castelberg
State	Accepted
Related	ADR023: Final Application Data ERD (p. 75)

Decision Record

In the context of the creation of a usable PoC application,
facing the need to store application data unrelated to the LLM,
we decided to use a relational database with the following data model,
and neglected other database models,
to achieve an easy to implement and scalable solution for our prototype,
accepting that the addition of another database adds a minor amount of complexity.

Images

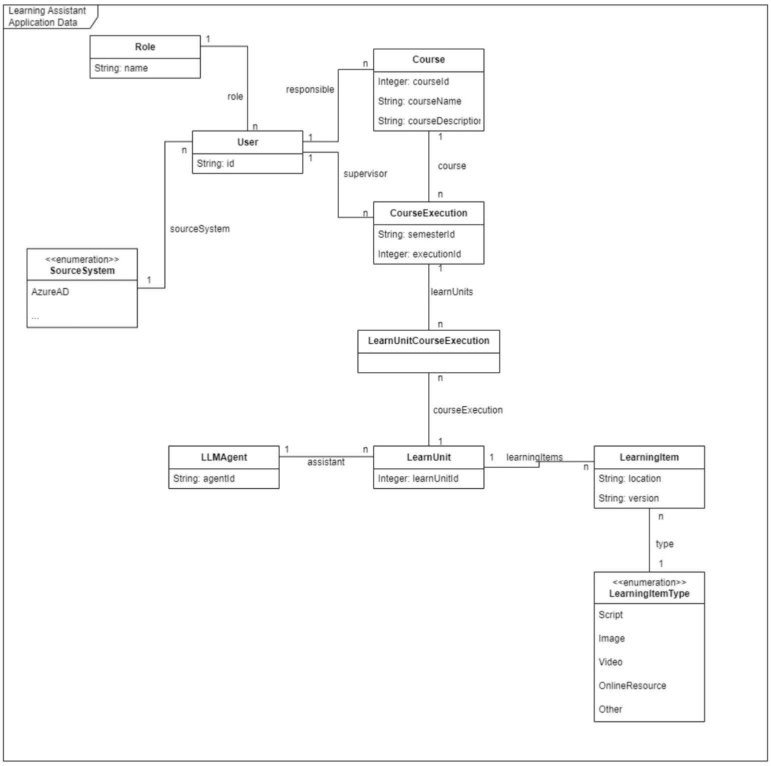


Figure 28: Projected data model of the finalized solution

7.4.2 ADR002: Initial C4 Container Diagram

ADR#	ADR002
Author	Dominik Castelberg
State	Accepted
Related	ADR003: Initial C4 API Component Diagram (p. 55)
	ADR004: Initial C4 Web UI Container Diagram (p. 56)
	ADR024: Final Container Diagram (p. 76)

Decision Record

In the context of the creation of a usable PoC application,
facing the need to map out the required sub-applications,
we decided to create a C4 container diagram,
and neglected other architecture documentation formats,
to create an initial overview over our required components,
accepting the risk of us needing to adapt the diagrams depending on our findings during evaluation.

Images

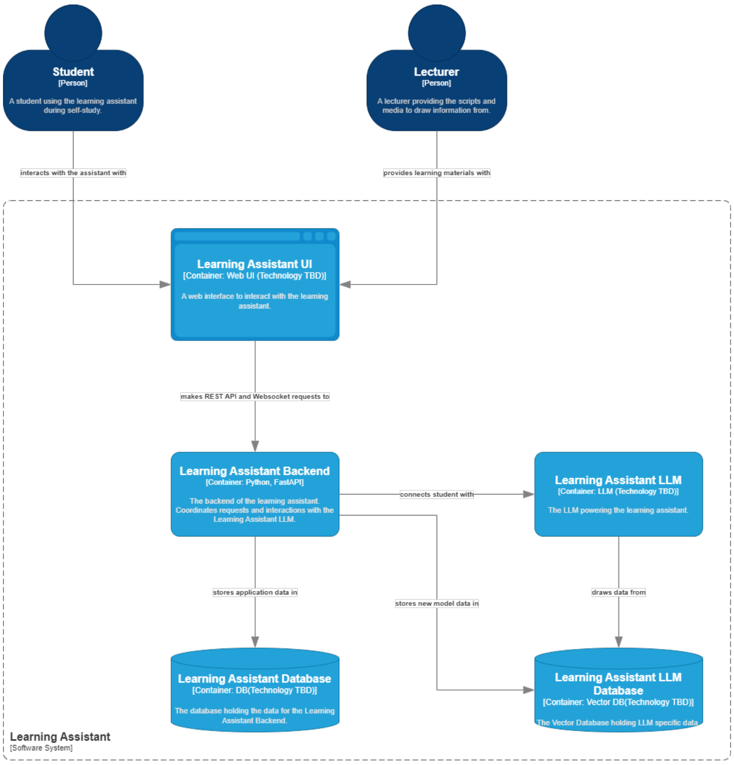

Student
[Person]
A student using the learning assistant during self-study.
Lecturer
[Person]
A lecturer providing the scripts and media to draw information from.
interacts with the assistant with
provides learning materials with
Learning Assistant UI
[Container: Web UI (Technology TBD)]
A web interface to interact with the learning assistant.
makes REST API and Websocket requests to
Learning Assistant Backend
[Container: Python, FastAPI]
The backend of the learning assistant. Coordinates requests and interactions with the Learning Assistant LLM.
connects student with
Learning Assistant LLM
[Container: LLM (Technology TBD)]
The LLM powering the learning assistant.
stores application data in
stores new model data in
draws data from
Learning Assistant Database
[Container: DB(Technology TBD)]
The database holding the data for the Learning Assistant Backend.
Learning Assistant LLM Database
[Container: Vector DB(Technology TBD)]
The Vector Database holding LLM specific data
Learning Assistant
[Software System]

Figure 29: Projected container diagram of the finalized solution

7.4.3 ADR003: Initial C4 API Component Diagram

ADR#	ADR003
Author	Dominik Castelberg
State	Accepted
Related	ADR002: Initial C4 Container Diagram (p. 54) ADR005: Permission System Declared Out of Scope (p. 57) ADR021: Final API Architecture (p. 73)

Decision Record

In the context of the creation of a usable PoC application,
facing the need for an initial API architecture,
we decided to create a C4 component diagram,
and neglected other architecture documentation formats,
to create an initial overview over our required components,
accepting the risk of us needing to adapt the diagrams depending on our findings during evaluation.

Images

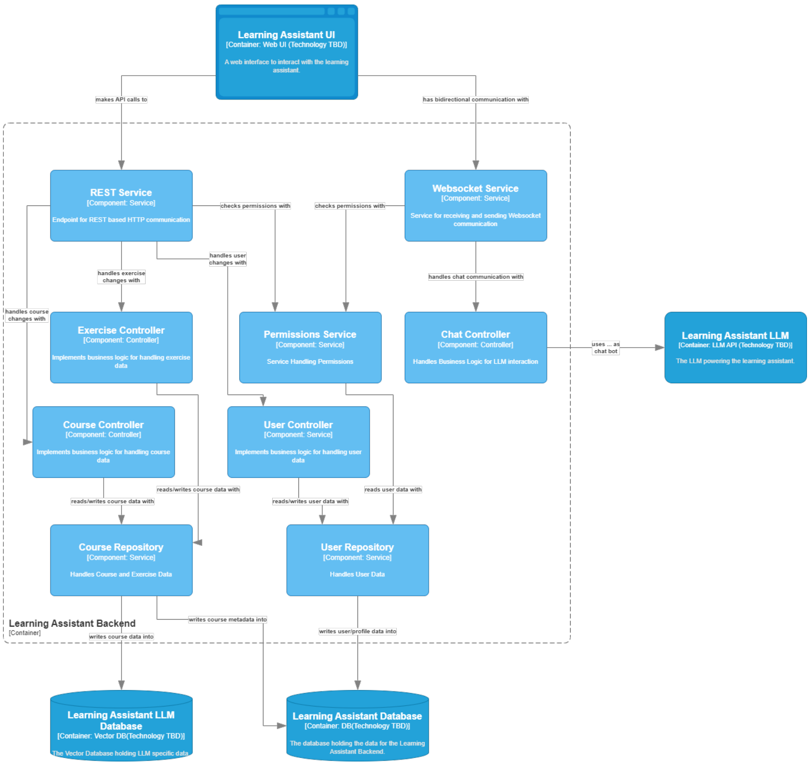

Learning Assistant UI
[Container: Web UI (Technology TBD)]
A web interface to interact with the learning assistant.
makes API calls to
has bidirectional communication with
REST Service
[Component: Service]
Endpoint for REST based HTTP communication
checks permissions with
checks permissions with
Websocket Service
[Component: Service]
Service for receiving and sending Websocket communication
handles user changes with
handles exercise changes with
handles chat communication with
handles course changes with
Exercise Controller
[Component: Controller]
Implements business logic for handling exercise data
Permissions Service
[Component: Service]
Service Handling Permissions
Chat Controller
[Component: Controller]
Handles Business Logic for LLM interaction
uses ... as chat bot
Learning Assistant LLM
[Container: LLM API (Technology TBD)]
The LLM powering the learning assistant.
Course Controller
[Component: Controller]
Implements business logic for handling course data
User Controller
[Component: Service]
Implements business logic for handling user data
reads/writes course data with
reads/writes course data with
reads/writes user data with
reads user data with
Course Repository
[Component: Service]
Handles Course and Exercise Data
User Repository
[Component: Service]
Handles User Data
writes course metadata into
writes user/profile data into
Learning Assistant Backend
[Container]
writes course data into
Learning Assistant LLM Database
[Container: Vector DB(Technology TBD)]
The Vector Database holding LLM specific data
Learning Assistant Database
[Container: DB(Technology TBD)]
The database holding the data for the Learning Assistant Backend.

Figure 30: Projected component diagram of the API

7.4.4 ADR004: Initial C4 Web UI Container Diagram

ADR#	ADR004
Author	Dominik Castelberg
State	Accepted
Related	ADR002: Initial C4 Container Diagram (p. 54) ADR005: Permission System Declared Out of Scope (p. 57) ADR022: Web UI Architecture (p. 74)

Decision Record

In the context of the creation of a usable PoC application,
facing the need for an initial Frontend architecture,
we decided to create a C4 component diagram,
and neglected other architecture documentation formats,
to create an initial overview over our required components,
accepting the risk of us needing to adapt the diagrams depending on our findings during evaluation.

Images

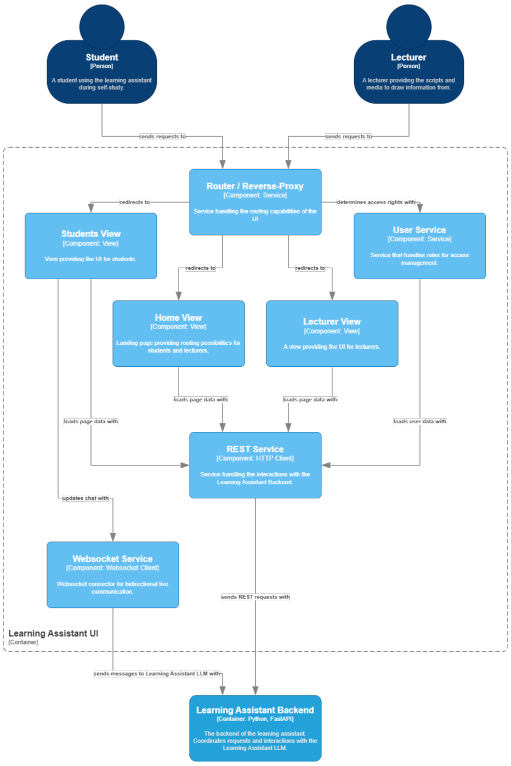

Student
[Person]
A student using the learning assistant during self-study.
Lecturer
[Person]
A lecturer providing the scripts and media to draw information from.
sends requests to
sends requests to
Router / Reverse-Proxy
[Component: Service]
Service handling the routing capabilities of the UI
redirects to
determines access rights with
Students View
[Component: View]
View providing the UI for students.
User Service
[Component: Service]
Service that handles roles for access management.
redirects to
redirects to
Home View
[Component: View]
Landing page providing routing possibilities for students and lecturers.
Lecturer View
[Component: View]
A view providing the UI for lecturers.
loads page data with
loads page data with
loads page data with
loads user data with
REST Service
[Component: HTTP Client]
Service handling the interactions with the Learning Assistant Backend.
updates chat with
Websocket Service
[Component: Websocket Client]
Websocket connector for bidirectional live communication.
sends REST requests with
Learning Assistant UI
[Container]
sends messages to Learning Assistant LLM with
Learning Assistant Backend
[Container: Python, FastAPI]
The backend of the learning assistant. Coordinates requests and interactions with the Learning Assistant LLM.

Figure 31: Projected component diagram of the web UI

7.4.5 ADR005: Permission System Declared Out of Scope

ADR#	ADR005
Author	Dominik Castelberg
State	Accepted
Related	ADR003: Initial C4 API Component Diagram (p. 55) ADR004: Initial C4 Web UI Container Diagram (p. 56)

Decision Record

In the context of the creation of a usable PoC application,
facing time and project scope constraints,
we decided to only implement components that are absolutely necessary for our prototype to function,
and neglected to implement a permission system,
to have more time for the goals of this project,
accepting that this must be implemented in a subsequent project.

7.4.6 ADR006: Deployment Tools

ADR#	ADR006
Author	Dominik Castelberg
State	Accepted
Related	-

Decision Record
In the context of our deployment strategy,
facing the need for an easy to set up and adjustable deployment model,
we decided to use Docker Compose,
and neglected more sophisticated and complex deployment tools such as Kubernetes and Docker Swarm,
to create a simple and quick deployment setup,
accepting the risk of major changes to our deployment strategy when moving the application out of the deployment stage and into production.

Comments
While we considered the the mentioned frameworks, their usage for our prototype would have been out of scope. If there is interest in bringing this project into a production environment, we recommend the use of an orchestration solution that enables the application to run distributed across multiple hosts to distribute the computational load.

Nonetheless will this prototype be delivered as a single host application. Under these circumstances it would not make sense to use more complex frameworks such as Kubernetes or Docker Swarm, which is why we opt for Docker Compose due to its superiority in the following points:

- Ease of use (less complexity)
- Development overhead (less configuration needed)
- Familiarity (we worked with it before)

7.4.7 ADR007: Language Selection for API

ADR#	ADR007
Author	Dominik Castelberg
State	Accepted
Related	ADR008: Usage of LangChain (p. 60) ADR009: Usage of FastAPI (p. 61)

Decision Record

In the context of the development of our API,
facing the need for a familiar high level language with a mature ecosystem in machine learning and web development,
we decided to use Python 3[40],
and neglected options such as Rust, Go, C++ C# and TypeScript,
to work with established tools such as LangChain and PyTorch,
accepting the risk of performance issues.

Comments

We were looking into other languages as well, but in the end still ended up using python for a multitude of reasons. The most important ones are:

- Familiarity
- The LangChain ecosystem
- Established and reliable web frameworks

C#, Go and TypeScript were additional languages we are familiar with, but they do not have a machine learning ecosystem that is comparable to the ecosystem of Python 3. Rust and C++ boast a strong presence in the machine learning space, but their low level nature and our relative lack of experience with them were considered to be major risks for the success of the project.

An older version of Python (2.X) was considered as well, but we opted for the more modern approach for the sake of future proofing.

---

[40] https://python.org

7.4.8 ADR008: Usage of LangChain

ADR#	ADR008
Author	Dominik Castelberg
State	Accepted
Related	ADR007: Language Selection for API (p. 59)

Decision Record
In the context of the development of our API,
facing the need for an easy way to work with large language models,
we decided to use LangChain[41],
and neglected another option in Haystack[42],
to work with a tool we are familiar with,
accepting no risks to speak of.

Comments
LangChain and Haystack two of the most well-known end-to-end frameworks if one wants to work with large language models. We are confident that we could have completed the project with Haystack as well, as its ease of use is often quoted as its biggest upside over LangChain.

We decide against Haystack and in favor of LangChain as we are already familiar with LangChain and therefore would not profit from switching to an unknown tech stack.

---

[41] https://www.langchain.com/
[42] https://haystack.deepset.ai/

7.4.9 ADR009: Usage of FastAPI

ADR#	ADR009
Author	Dominik Castelberg
State	Accepted
Related	ADR007: Language Selection for API (p. 59) ADR10: Usage of SQLModel (p. 62)

Decision Record

In the context of the development of our API,
facing the need for a REST API framework that supports WebSockets[43] connections,
we decided to use FastAPI[44],
and neglected other options such as Django[45] or Flask[46],
to work with a feature-rich and yet lightweight framework we are familiar with,
accepting no risks to speak of.

Comments

We think that Django, while feature rich-and mature, requires more development overhead and takes longer to take off the ground when compared to the light-weight options in FastAPI and Flask. Between FastAPI and Flask we have more familiarity with FastAPI, and the ASGI[47] compatibility of FastAPI means that it can be hosted with Uvicorn[48] and therefore is supporting WebSockets.

---

[43] https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API
[44] https://fastapi.tiangolo.com/
[45] https://www.djangoproject.com/
[46] https://flask.palletsprojects.com/en/3.0.x/
[47] https://asgi.readthedocs.io/en/latest/index.html
[48] https://www.uvicorn.org/

7.4.10 ADR10: Usage of SQLModel

ADR#	ADR010
Author	Dominik Castelberg
State	Accepted
Related	ADR007: Language Selection for API (p. 59) ADR009: Usage of FastAPI (p. 61)

Decision Record

In the context of application data persistence in our API,
facing the need for a way to store data in a database,
we decided to use SQLModel[49],
and neglected other options such as SQLAlchemy[50] or psycopg2[51],
to use a framework by the same developer as FastAPI and hence ensuring compatibility while also profiting from the the ease-of-use of an ORM,
accepting the risk of reduced control and potentially vendor lock-in in the future.

Comments

SQLModel uses both Pydantic[52] and SQLAlchemy under the hood for its entities, which provides some convenience.
SQLModel also allows the use of SQLAlchemy tools and classes when interacting with SQLModel entities.
SQLModel being endorsed as the recommended ORM/database adapter for working with FastAPI put it over psycopg2, which allows for more control.

---

[49] https://sqlmodel.tiangolo.com/
[50] https://www.sqlalchemy.org/
[51] https://www.psycopg.org/
[52] https://docs.pydantic.dev/latest/

7.4.11 ADR011: Usage of Angular

ADR#	ADR011
Author	Dominik Castelberg
State	Accepted
Related	-

Decision Record

In the context of the development of our Web UI,
facing the need for a framework that allows us to quickly prototype a user interface,
we decided to use Angular[53],
and neglected other options such as React[54] or Vue[55],
to work with an established framework we are familiar with,
accepting that future developers may have other preferences.

---

[53] https://angular.dev/
[54] https://react.dev/
[55] https://vuejs.org/

7.4.12 ADR012: Usage of a local database

ADR#	ADR012
Author	Dominik Castelberg
State	Accepted
Related	ADR013: Usage of PostgreSQL (p. 65)

Decision Record

In the context of application data persistence,
facing the need for a database management system,
we decided to use a locally hosted database,
and neglected other options such Azure Cosmos DB,
to ensure that our prototype can work with sensitive data and to keep development costs low,
accepting the risk of needing to switch to a cloud based approach for scaling purposes if the data privacy risk is deemed negligible.

Comments

To use classified or otherwise sensitive data with our prototype was a desired feature. The easiest and most reliable way to achieve this is to make sure that the data used for RAG or to fine-tune a model never leaves the system the user has control over.

Using a local database also keeps costs low during development, especially if a DBMS without licensing fees is used.

7.4.13 ADR013: Usage of PostgreSQL

ADR#	ADR013
Author	Dominik Castelberg
State	Accepted
Related	ADR012: Usage of a local database (p. 64)

Decision Record

In the context of application data persistence,
facing the need for a database management system,
we decided to use PostgreSQL[56],
and neglected other options such as SQL Server[57] or SQLite[58],
to profit from its vector database extension pgvector[59],
accepting the risk of the relational database and the vector database potentially needing to be separated in the future due to scalability concerns.

Comments

PostgreSQL supports vector data with its extension pgvector, which allows us to hold all our data in a single database. This is perfect for a prototype application, as it reduces the system complexity. It is, however, not the best architecture for a large scale application, as such a software would profit from splitting the vector database from the relational database to distribute the computational load onto multiple hosts.

SQLite also support vector data, but we are not familiar with this DBMS. We have found no evidence for SQL Server supporting vector data.

---

[56] https://www.postgresql.org/
[57] https://www.microsoft.com/en-us/sql-server
[58] https://sqlite.org/
[59] https://github.com/pgvector/pgvector

7.4.14 ADR014: Usage of Min.IO

ADR#	ADR014
Author	Dominik Castelberg
State	Accepted
Related	-

Decision Record
In the context of user data persistence,
facing the need for an object store,
we decided to use Min.IO[60],
and neglected another option in PostreSQL BLOB storage,
to use a locally hostable service that is specialized in storing and organizing large files,
accepting a mildly higher system complexity.

Comments
PostgreSQL would support the storage of BLOB data, but we are already familiar with the use of Min.IO and features such as multi-node and multi-drive hosting act as future proofing in case of interest in horizontal scaling.

---

[60] https://www.min.io/

7.4.15 ADR015: Introduction of LearningItemContainer

ADR#	ADR015
Author	Dominik Castelberg
State	Accepted
Related	ADR001: Initial Application Data Model (p. 53) ADR016: (REVERT) Introduction of LearningItemContainer (p. 68) ADR023: Final Application Data ERD (p. 75)

Decision Record

In the context of the application data model,
facing the need to not force a lecturer to upload a script for every LearnUnit,
we decided to introduce a joined table inheritance pattern for LearnUnit, CourseExecution and Course,
and neglected other options such as single table inheritance or not using inheritance at all,
to provide a way for lecturers to upload a file to a single course and it propagating down to individual LearnUnits,
accepting a mild increase in model complexity.

Images

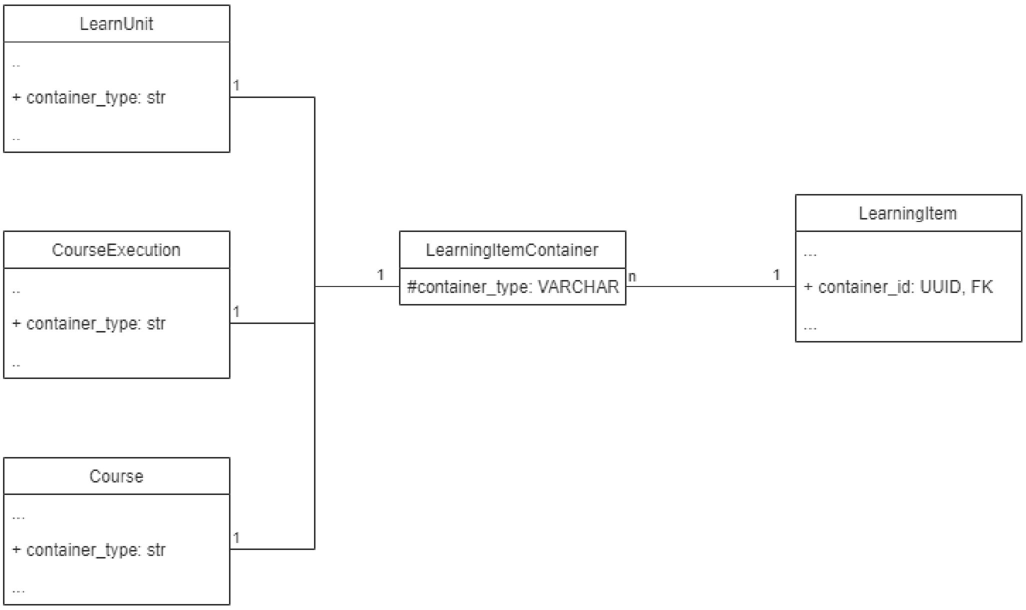


Figure 32: Cutout of a data model illustrating the new setup with a new joined table "LearningItemContainer"

7.4.16 ADR016: (REVERT) Introduction of LearningItemContainer

ADR#	ADR016
Author	Dominik Castelberg
State	Accepted
Related	ADR015: Introduction of LearningItemContainer (p. 67) ADR023: Final Application Data ERD (p. 75)

Decision Record

In the context of the application data model,
facing the need to work around the inability to properly work with inheritance of SQLModel,
we decided to revert the introduction of the inheritance pattern for the classes LearnUnit, CourseExecution and Course,
and neglected other options such as single table inheritance,
to ensure our application is stable without compromising on introduced features,
accepting a dirty architecture at data level.

Images

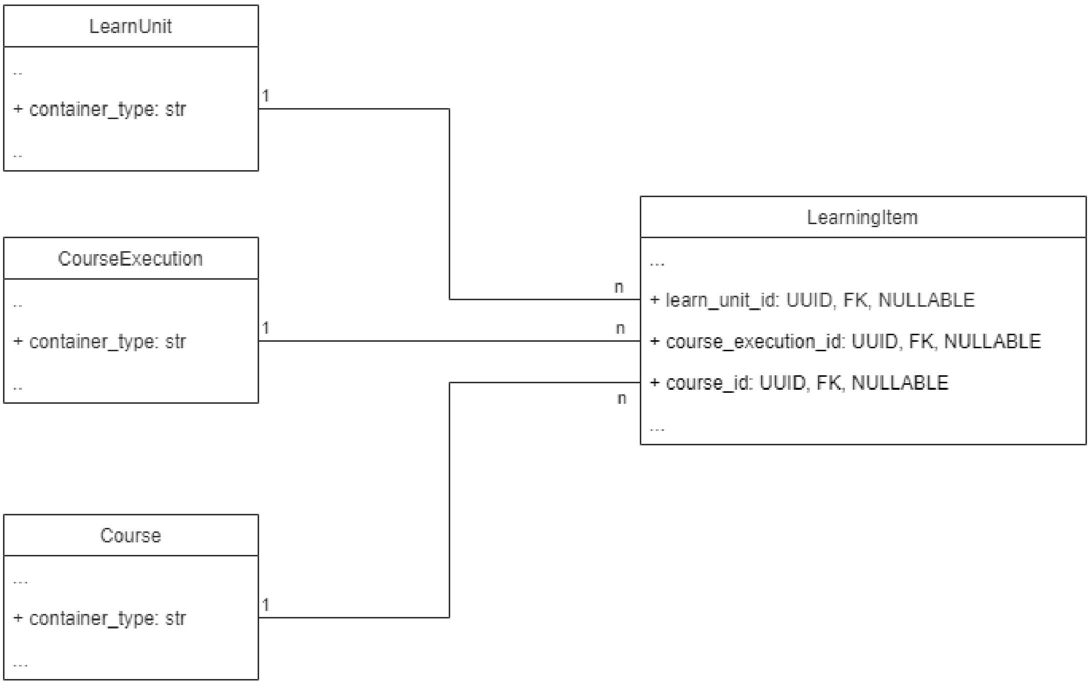


Figure 33: Cutout of a data model illustrating the new setup without inheritance

7.4.17 ADR017: Introduce Model Wrapper

ADR#	ADR017
Author	Dominik Castelberg
State	Accepted
Related	ADR021: Final API Architecture (p. 73)

Decision Record

In the context of the development of our API,
facing the requirement to work with multiple models,
we decided to wrap model specific code behind wrapper classes inheriting from a pseudo-abstract base class,
and neglected other options such as dependency injection,
to not compromise on user convenience while keeping the architecture complexity low,
accepting a possible switch to dependency injection after moving out of the prototyping stage.

Images

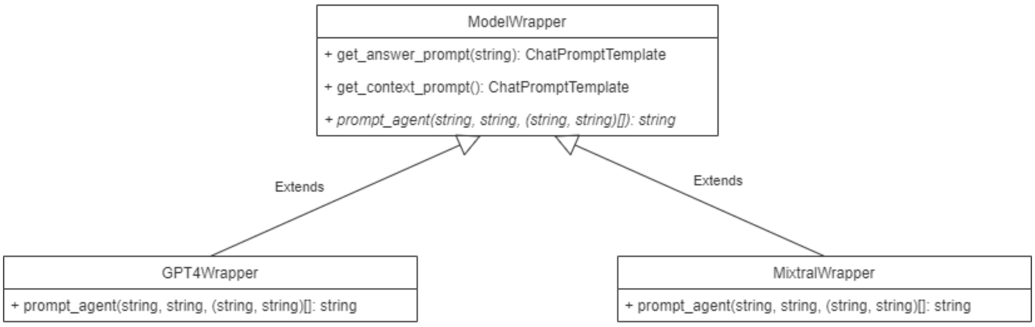

ModelWrapper
+ get_answer_prompt(string): ChatPromptTemplate
+ get_context_prompt(): ChatPromptTemplate
+ prompt_agent(string, string, (string, string)[]): string
Extends
Extends
GPT4Wrapper
+ prompt_agent(string, string, (string, string)[]: string
MixtralWrapper
+ prompt_agent(string, string, (string, string)[]: string

Figure 34: Cutout of a class diagram illustrating ModelWrappers

7.4.18 ADR18: Introduce DocumentLoaders

ADR#	ADR018
Author	Dominik Castelberg
State	Accepted
Related	ADR021: Final API Architecture (p. 73)

Decision Record

In the context of the development of our API,
facing the requirement to work with multiple data types,
we decided to create static, document type specific helper classes,
and neglected other options such as dependency injection,
to not compromise on user convenience while keeping the architecture complexity low,
accepting a possible switch to dependency injection after moving out of the prototyping stage.

Images

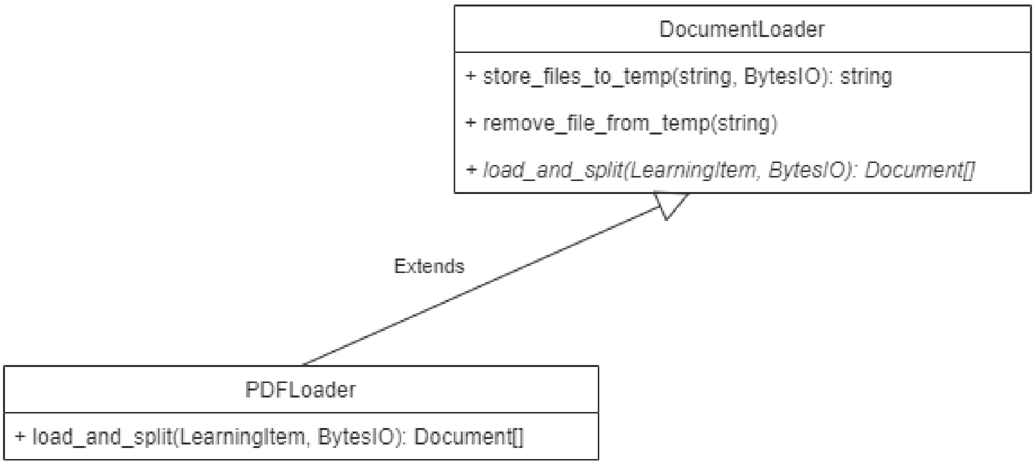

DocumentLoader
+ store_files_to_temp(string, BytesIO): string
+ remove_file_from_temp(string)
+ *load_and_split(LearningItem, BytesIO): Document[]*
Extends
PDFLoader
+ load_and_split(LearningItem, BytesIO): Document[]

Figure 35: Cutout of a class diagram illustrating DocumentLoaders

7.4.19 ADR19: Implement Services and Repositories as collections of static methods

ADR#	ADR019
Author	Dominik Castelberg
State	Accepted
Related	ADR021: Final API Architecture (p. 73)

Decision Record

In the context of the development of our API,
facing the requirement to keep the prototype simple,
we decided write to our services and repositories as collections of static methods,
and neglected other options such as dependency injection,
to keep development overhead and system complexity low,
accepting our components being services and repositories in name only.

7.4.20 ADR20: Use of Multilingual-E5-base as the embedding model

ADR#	ADR020
Author	Dominik Castelberg
State	Accepted
Related	ADR021: Final API Architecture (p. 73)

Decision Record

In the context of the development of our API,
facing the need of an embedding model to turn documents into vector data,
we decided to use Multilingual-E5-base,
and neglected other options such as Multilingual-E5-large or Multilingual-E5-small,
to find a balance between resource consumption and performance,
accepting no major risks to speak of.

Comments

Multilingual-E5 is a series of established embedding models capable of working with multiple languages, which allows the AI Assistant to handle documents in numerous languages. We decided against using Multilingual-E5-large due to resource consumption and therefore scalability concerns, while still recognizing that it is a valid option when the application is being run on a more powerful host.

7.4.21 ADR021: Final API Architecture

ADR#	ADR021
Author	Dominik Castelberg
State	Accepted
Related	ADR003: Initial C4 API Component Diagram (p. 55) ADR017: Introduce Model Wrapper (p. 69)

Decision Record

In the context of the creation of our API,
facing the need for a quick to implement and yet maintainable API architecture,
we decided to use a monolith approach as defined in the attached image,
and neglected more complex architecture types,
to maintain both extensibility and time consciousness,
accepting the risk of major refactoring work when moving the application out of the deployment stage and into production.

Images

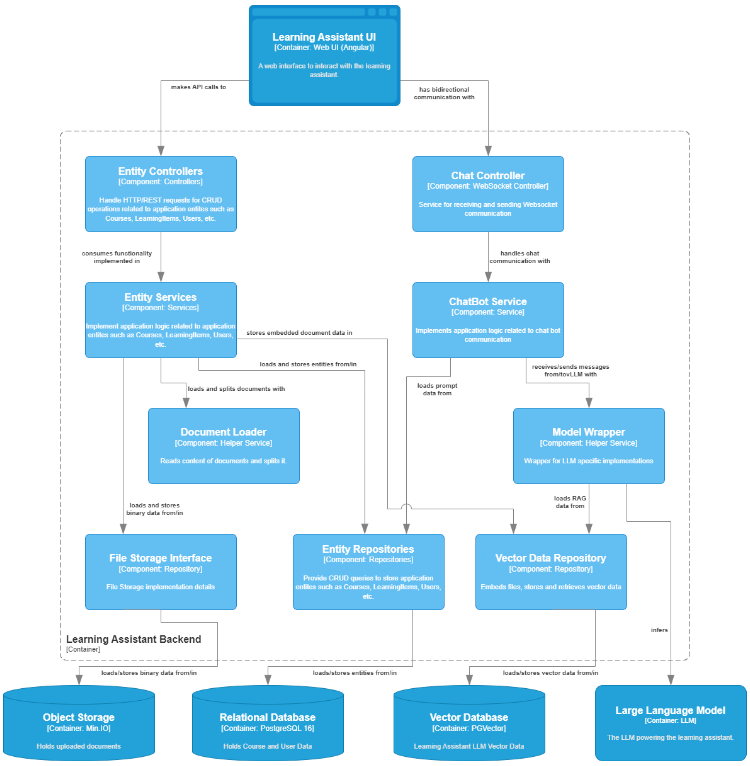

Learning Assistant UI
[Container: Web UI (Angular)]
A web interface to interact with the learning assistant.
makes API calls to
has bidirectional communication with
Entity Controllers
[Component: Controllers]
Handle HTTP/REST requests for CRUD operations related to application entites such as Courses, LearningItems, Users, etc.
Chat Controller
[Component: WebSocket Controller]
Service for receiving and sending Websocket communication
consumes functionality implemented in
handles chat communication with
Entity Services
[Component: Services]
Implement application logic related to application entites such as Courses, LearningItems, Users, etc.
stores embedded document data in
ChatBot Service
[Component: Service]
Implements application logic related to chat bot communication
loads and stores entities from/in
receives/sends messages from/tovLLM with
loads and splits documents with
loads prompt data from
Document Loader
[Component: Helper Service]
Reads content of documents and splits it.
Model Wrapper
[Component: Helper Service]
Wrapper for LLM specific implementations
loads and stores binary data from/in
loads RAG data from
File Storage Interface
[Component: Repository]
File Storage implementation details
Entity Repositories
[Component: Repositories]
Provide CRUD queries to store application entites such as Courses, LearningItems, Users, etc.
Vector Data Repository
[Component: Repository]
Embeds files, stores and retrieves vector data
infers
Learning Assistant Backend
[Container]
loads/stores binary data from/in
loads/stores entities from/in
loads/stores vector data from/in
Object Storage
[Container: Min.IO]
Holds uploaded documents
Relational Database
[Container: PostgreSQL 16]
Holds Course and User Data
Vector Database
[Container: PGVector]
Learning Assistant LLM Vector Data
Large Language Model
[Container: LLM]
The LLM powering the learning assistant.

Figure 36: Component diagram of the API

7.4.22 ADR022: Web UI Architecture

ADR#	ADR022
Author	Dominik Castelberg
State	Accepted
Related	ADR004: Initial C4 Web UI Container Diagram (p. 56)

Decision Record

In the context of the creation of our front end application,
facing the need for a quick to implement front end architecture,
we decided to use a basic Angular standalone application as described in the attached image,
and neglected the use of other front end frameworks,
to work with a framework we are familiar with,
accepting the risk of future maintainers preferring other frameworks and hence re-writing it.

Images

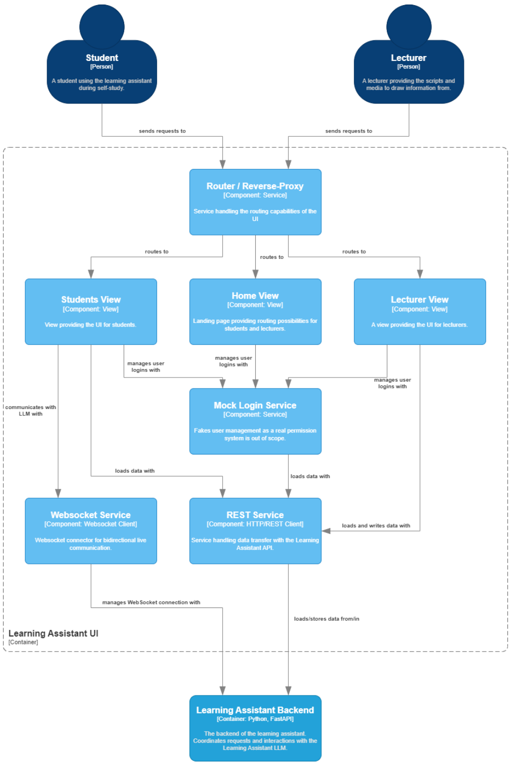

Student
[Person]
A student using the learning assistant during self-study.
Lecturer
[Person]
A lecturer providing the scripts and media to draw information from.
sends requests to
sends requests to
Router / Reverse-Proxy
[Component: Service]
Service handling the routing capabilities of the UI
routes to
routes to
routes to
Students View
[Component: View]
View providing the UI for students.
Home View
[Component: View]
Landing page providing routing possibilities for students and lecturers.
Lecturer View
[Component: View]
A view providing the UI for lecturers.
manages user logins with
manages user logins with
manages user logins with
communicates with LLM with
Mock Login Service
[Component: Service]
Fakes user management as a real permission system is out of scope.
loads data with
loads data with
Websocket Service
[Component: Websocket Client]
Websocket connector for bidirectional live communication.
REST Service
[Component: HTTP/REST Client]
Service handling data transfer with the Learning Assistant API.
loads and writes data with
manages WebSocket connection with
loads/stores data from/in
Learning Assistant UI
[Container]
Learning Assistant Backend
[Container: Python, FastAPI]
The backend of the learning assistant. Coordinates requests and interactions with the Learning Assistant LLM.

Figure 37: Component diagram of the web UI

7.4.23 ADR023: Final Application Data ERD

ADR#	ADR023
Author	Dominik Castelberg
State	Accepted
Related	ADR001: Initial Application Data Model (p. 53)

Decision Record

In the context of data persistence,
facing the need for the documentation of our data model,
we decided to generate the final data model diagram based on the state of our database,
and neglected to manually create it,
to obtain the attached image,
accepting the risk of changes requiring a re-export of the diagram.

Images

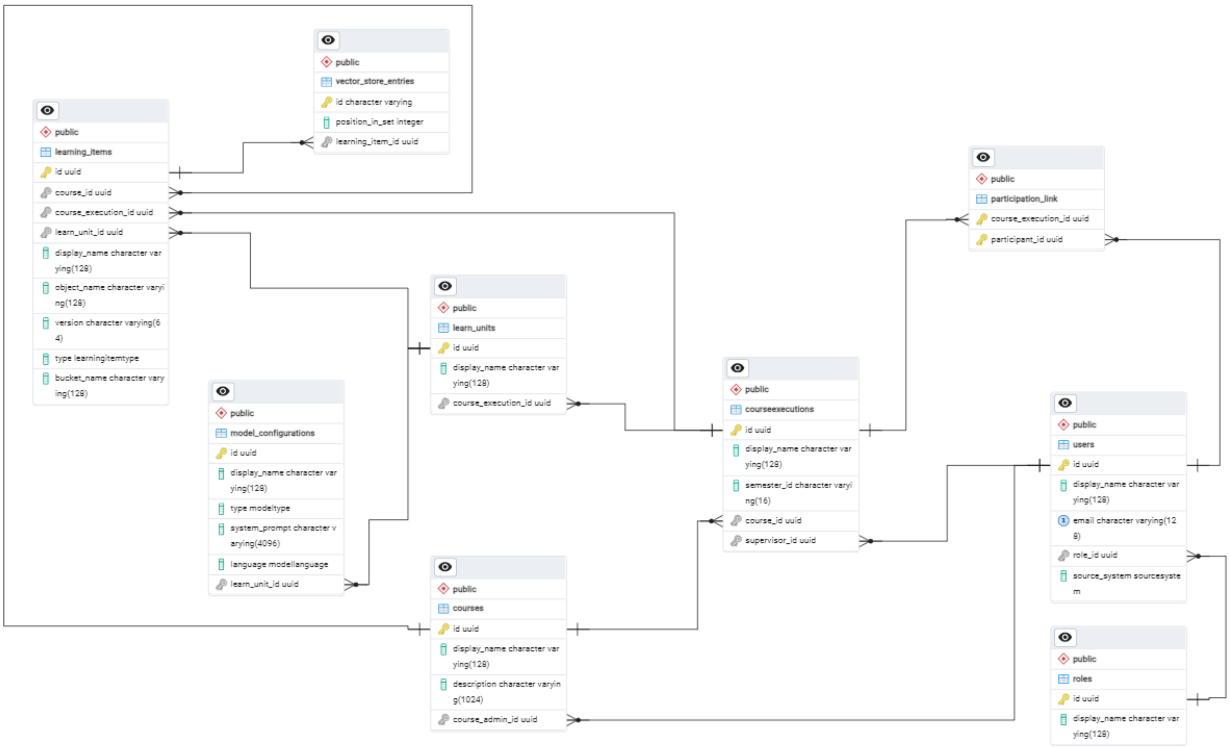


Figure 38: Entity Relationship Diagram of the Learning Assistant Application

7.4.24 ADR024: Final Container Diagram

ADR#	ADR99
Author	Dominik Castelberg
State	Accepted
Related	ADR002: Initial C4 Container Diagram (p. 54)

Decision Record

In the context of our deployment strategy,
facing the need for a distributed container setup,
we decided to group our containers as described in the attached image,
and neglected to use other setups,
to distribute the load across as many containers as reasonably possible,
accepting the risk of an increased solution complexity.

Images

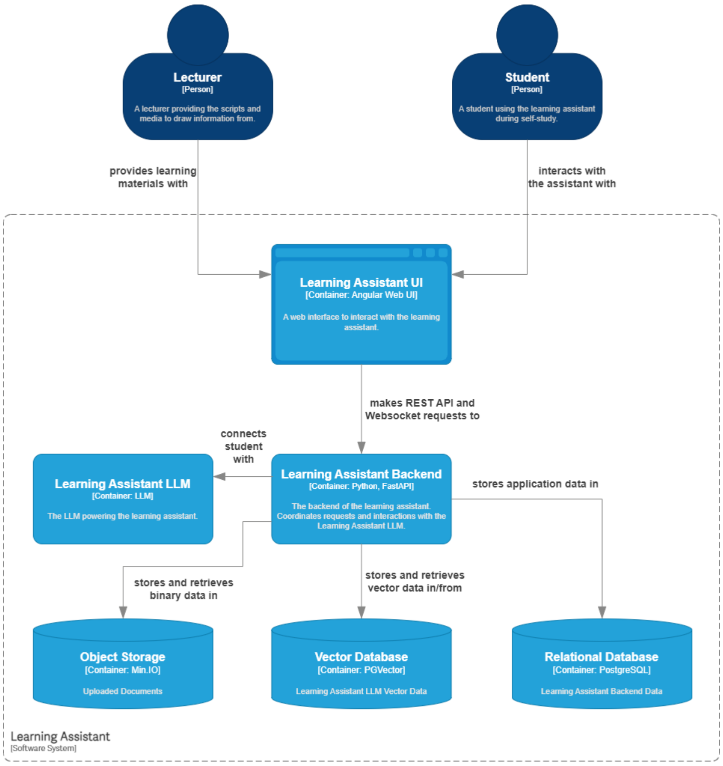

Lecturer
[Person]
A lecturer providing the scripts and media to draw information from.
Student
[Person]
A student using the learning assistant during self-study.
provides learning materials with
interacts with the assistant with
Learning Assistant UI
[Container: Angular Web UI]
A web interface to interact with the learning assistant.
makes REST API and Websocket requests to
connects student with
Learning Assistant LLM
[Container: LLM]
The LLM powering the learning assistant.
Learning Assistant Backend
[Container: Python, FastAPI]
The backend of the learning assistant. Coordinates requests and interactions with the Learning Assistant LLM.
stores application data in
stores and retrieves binary data in
stores and retrieves vector data in/from
Object Storage
[Container: Min.IO]
Uploaded Documents
Vector Database
[Container: PGVector]
Learning Assistant LLM Vector Data
Relational Database
[Container: PostgreSQL]
Learning Assistant Backend Data
Learning Assistant
[Software System]

Figure 39: Container diagram of the Learning Assistant

7.5 Deployment & Usage Guide

7.5.1 Configuration

The prototype comes with a Docker Compose[61] template for development purposes. It can be found in the root folder of the prototype and is named `docker-compose.development.yaml`. It deploys the following services:

database	This container provides a PostgreSQL 16 server with the pgvector extension installed.
database-manager	This container provides a webserver hosting pgAdmin, a database server management suite for postgres. It allows you to inspect the current state of your database within the database container. The login data can be found in the environment variable declaration of the Docker Compose configuration file.
minio	Provides a local server instance of Min.IO, an object storage. It saves the uploaded documents in their original state. It also hosts a management interface on port 9001 by default. The login data can be found in the environment variable declaration of the Docker Compose configuration file.
ai-assistant-backend	The FastAPI-based backend of the application. It exposes a REST API and a Websocket endpoint that can be consumed. Accessing its root URL on port 8000 with a browser opens a Swagger UI providing an API documentation.
ai-assistant-frontend	The Angular web frontend of the application. It can be accessed on port 4200 by default. The frontend provides a user friendly way to interact with the backend and its functionalities.

Two launch profiles have been pre-configured: `slim` and `dev`. `slim` only launches the containers that are necessary for the application to function, while `dev` launches all containers. In the original setup, this means that `slim` forgoes to launch the database-manager service.

A number of environment variables are configured as well. Some environment variables in the ai-assistant-backend service are optional and their values can also be configured in the application-internal `config.docker_compose.json` which can be found in the `./web_api/code/config` sub-folder of the code base.

WARNING: If environment variables have been set, they take precedent over set values in config files!

---

[61] https://docs.docker.com/compose/

A list of available environment variables to configure for both the database and the minio service can be found in their official documentation. To the ai-assistant-backend service, the following environment variables are available:

Name	Optional|Required	Comment
AI_ASSISTANT_ENVIRONMENT	Required	Tells the prototype, which application config file to read.
AI_ASSISTANT_SEED	Optional	Tells the prototype, if and which database seed to use.
POSTGRES_USER	Required unless set in config file	Must be identical to the value set for the database service.
POSTGRES_PASSWORD	Required unless set in config file	Must be identical to the value set for the database service.
MINIO_ROOT_USER	Required unless set in config file	Must be identical to the value set for the minio service.
MINIO_ROOT_PASSWORD	Required unless set in config file	Must be identical to the value set for the minio service.

This setup does **not** deploy a local Mixtral-8x7b instance. If its use is desired, it must be provided by the user.

7.5.2 Starting the Application

It is recommended to start the application in its Docker Compose setup. This can be achieved with the following command, executed in the project's root folder:

```
docker-compose --file docker-compose.development.yaml --profile dev up --build
```

Note the `--profile` and `--build` flags. They select the aforementioned launch profile and force Docker Compose to re-build images with code changes, respectively.

To shut down the prototype, the following command can be executed:

```
docker-compose --file docker-compose.development.yaml --profile dev down --volumes
```

Note the `--volumes` flag. This flag tells Docker to delete any created data after the application has been shut down. If data created during the runtime of the prototype must be persisted after shutdown, this flag must be omitted and any AI_ASSISTANT_SEED entries in the environment variables must be removed to prevent the prototype from resetting the application data database.

The Learning Assistant application does not include a built in authentication system, as this feature was declared to be out of scope. It is still possible to select user profiles to simulate using the application as specific users. This can be done with a drop down menu item at the top right of your screen.

7.5.3 Configuring a Chat Bot

The page **Course Administration** provides tools to create and manage courses that students can be enrolled in. They are structured as follows:

Courses are the top level entity and define the name and description of the course. A user with the **Lecturer** role can get assigned to the position of course administrator. This has no effect in the current prototype, but can be used as the basis for a more sophisticated permission system implemented after this project has ended. **Learning Item**s, or **files**, can be assigned to a course, if the goal is for every person enrolled in this course to have a chat bot with access to the contained information.

Course Executions are instances of when a course is actively being taught. They can be named and are assigned to a specific semester. A way to use this abstraction level is to provide the same course in multiple languages, or to make adjustments for specific semesters. A user with the **Lecturer** role can get assigned to the position of course supervisor. This also has no effect in the current prototype. Users with the role **Student** can get assigned to course executions, which gives them access to all chat bots that are part of them. **Learning Items** can also be assigned to course executions, making them available to all chat bots contained within a course execution.

Learn Units are a more abstract part of a course execution. A potential use would be to model individual exercise sessions or lectures as learn units. Learn units can be given a name and **Learning Items** can get assigned to them. Upon saving a learn unit, all learning items within the learn unit and its parent course execution and course get embedded and definitely assigned to this learn unit. Chat bots assigned to this learn unit have now access to all documents that were embedded.

Model Configurations are the individual chat bots assigned to a learn unit. As mentioned before, chat bots have access to all resources that are assigned to their parent learn unit. A model configuration can be configured with a display name, a system prompt, the LLM type and the language the chat bot should communicate in. While Swiss German is offered as an option, it is mostly available for entertainment purposes and should not be used without disclaimer in a production environment.

A starting point for a reasonably well performing system prompt would be the following:

> You are a helpful Teacher's assistant supporting university students during their studies. The students you are talking to are preparing for an inverted classroom type lecture and your job is to evaluate if the students are adequately prepared. Try to rate the student based on Bloom's Taxonomy scale, but do NOT mention the scale itself! After every interaction, assess the student's readiness for the upcoming lecture. If you are confident in your assessment, tell the student whether they are ready or not. If they are not ready, also tell them why you think they should keep studying. Avoid statements that start with "can you..." and use more demanding formulations like "Explain..".

WARNING: If the chat bots are to be used with confidential data, the following model types should NOT be used:

1. GPT-4o
2. GPT-4
3. GPT-3.5-Turbo

These model types are hosted behind an API by OpenAI[62], and vector data based on the documents uploaded to the Learning Assistant system will be send to this API. We have no insight into the internal business processes of OpenAI and do not know how they handle this data. We therefore can not confirm that the confidentiality of any sent data is guaranteed.

7.5.4 Using the Chat Bot

Users with the **Student** role can access their chat bots on the page **Learning Assistant**. Upon selecting a chat bot, an initial message will be sent to the back end, prompting the chat bot to greet the student and start with the readiness assessment. Chat messages will be displayed on the right side of the screen.

The chat history with each chat bot gets saved in the browser local storage and is not cached on the Learning Assistant system. This allows users to continue the chat even after long stretches of inactivity.

7.5.5 Troubleshooting

In general, the **Limitations** chapter lists known bugs with the prototype itself. Other than that, we expect the following issues to occasionally appear:

[Errno -3] Temporary failure in name resolution	This is a known issue in Docker for Windows: https://github.com/docker/for-win/issues/13808. Restarting the host Docker service usually resolves this issue.
I restarted the backend and now my data is gone	It is likely that the environment variable AI_ASSISTANT_SEED was set. This leads to the API application cleaning the database and then seeding it with new data. You can still recover your documents from the object storage.
I can not delete some Course Data	This is a known bug that sadly can not be fixed without a major rewrite of the data layer components. The object-relational mapper (ORM) used by this application does not support cascading deletes, which was not discovered until late into the development of this prototype.
I can not see any content on the page, or chat bots I have configured are not showing up	Make sure you have selected a user that has access to the requested resource.

---

[62] https://openai.com/

7.6 Limitations

7.6.1 Known Bugs

Cascade Delete does not work

Items that have children can not be deleted at this point in time. The reason for this is the inability of SQLModel to handle cascade clauses on foreign keys.

Chat window can freeze

The chat window occasionally stops scrolling after a while. Can be remedied by re-loading the page.

Model Configuration gets detached from Learn Unit when it gets created before documents get embedded

This bug was found late during development and could not get investigated before project end. Likely an issue in the front end application, where the Lecturer View component does not get notified of changes in the Model Configuration Editor.

7.6.2 Potential Improvements

Replace/Remove SQLModel

SQLModel has proven itself as far more limited less mature than anticipated and advertised. Switching to using SQLAlchemy and Pydantic separately would not lead to a loss in features and would allow us to properly configure cascading deletes.

Database Query Optimizations

As of right now, we are repeatedly fetching data from our database instead of using join statements in our query or working with stored procedures. Once a large number of users are accessing the UI, this leads to a large number of transactions and IO latency issues. In order to scale the application, this must get resolved.

Deployment Strategy

We are currently deploying the application as a single-host container application with docker compose. This puts a large computational load on the host, especially during embedding processes or LLM inference with a locally hosted Mixtral instance.

To combat these issues, we recommend using a container orchestration framework that is able to control containers on multiple hosts. Examples for a product like this would be Kubernetes or Docker Swarm. This way, the computational load can be distributed and horizontal scaling with multiple application instances behind a session-supporting load balancer such as Traefik[63] can be achieved.

---

[63] https://traefik.io/traefik/

We selected our tech stack carefully for our application to support horizontal scaling with minor in-code changes:

- Min.IO natively deploys multi-node, multi-host run modes and is designed to work in large scale distributed systems
- PostgreSQL can be turned into a distributed database with Citus[64]
- Self-hosted LLMs can be accessed with our used LangChain adapter as long as they are running behind a vLLM[65] API.

Required in-code changes would include changes to the database and file storage adapter configurations so they can handle sharded endpoints. We do not anticipated changes outside of the data layer of the API application.

---

[64] https://github.com/citusdata/citus
[65] https://docs.vllm.ai/en/stable/

8 Development Environment

8.1 Prototype Development Environment

To continue development on this application, one needs to have the following tools installed:

1. The package manager **conda**
2. A Python3-compatible IDE
3. The package manager **npm**
4. Angular CLI
5. A Angular-compatible IDE
6. Docker
7. Docker Compose

The file `/code/web_api/conda_env.yml` can be used to create a new conda environment with the required packages. This setup allows to start the application with docker and edit the code in an IDE.

To run the API application locally, one must pull the content of the Multilingual-E5-base repository and store its content in the folder `/model_images/multilingual-e5-base`. This is not recommended.

The recommended IDE for prototype development is Visual Studio Code.

8.2 Model Evaluation Development Environment

In the current form the Jupyter Notebooks can be executed needing only the following python packages:

- *jupyter*
- *ipywidgets*

If further development or automated testing is to be implemented the following python packages are recommended:

- *openai >= 1.20.0*
- *langchain >= 0.2.4 (requires Python <4.0, >=3.8.1)*

Further details and instructions can be found in the Evaluation README.md file in the ZIP archive.

9 Discussion

9.1 Interpretation

All out of the box models that were evaluated showed great promise to be used for each of the provided subtasks. During the evaluation they did not hallucinate incorrect data and stayed within the border of the given lecture script based on the system prompt. While GPT-4-turbo and Mixtral-8x7B-Instruct-v0.1 both produced responses of similar high quality the differences in output style could be steered towards the preferred style through system prompting. GPT-3.5-turbo lacked a bit behind in regards to the level of detail within it's responses.

RAG shows a lot of promise. It is possible to generate high quality user experiences and achieve high specificity with little effort. In fact, the biggest hurdle for an organisation willing to introduce RAG is the infrastructure required to build services that are capable of using RAG at scale. When compared to applications with fine-tuned models, RAG requires a strategy for document handling and object storage. The reward for such a system is, as demonstrated by our prototype, the flexibility one can achieve with such infrastructure. Instead of going through the long and resource-heavy process of fine-tuning a model, one can create new expert chat bots within minutes.

Another major strength of RAG as a process is the amount of control a user or service provider has over their data when using RAG. A major weakness in privacy for fine-tuned models is that their training data could get reconstructed and leaked with skilled prompting, while RAG provides its context data to the model on a request-by-request basis in a process that is completely controlled by the service provider.

AI Assistants enhanced with RAG were able to create questions related to the lecture script without being provided a set of sample questions. It also was able to evaluate the quality of an answer to their question and was able to provide a correct and exhaustive answer if the student failed.

With our prototype, the configuration effort by lecturers is minimal. All they need to provide to an AI assistant are PDF files containing the content they wish the students to prepare for and a prompt that provides in-depth instructions for how the model is meant to evaluate the answer quality.
The AI Assistants produced by the prototype application have shown to be able to converse in multiple languages and can be provided with course material from different subjects.

9.2 Answers to the initial questions

In regards to the questions posed at Goals 3.2

9.2.1 Necessary Information

Questions:

- *What information is needed for the system to work, for example, does it suffice for the teacher to just provide the Lesson Script or does the teacher need to provide predefined questions, do these questions need an associated gold standard solution or can system figure out a correct answer based on the Script?*
- *What input from the teacher is needed for the system to work?*
- *Does it work on the provided script?*

Each subproblem can be solved using Mixtral or GPT only needing the lecture script and an initial system prompt tailored to the subproblem to work. Evaluating the correctness of answers worked reliably using only the lecture script without providing gold standard answers. However the images in the lecture script can not be used.

9.2.2 Feedback

Questions:

- *How can the system evaluate what is the student's comprehension of a certain topic and generate/look up new questions based on this information?*
- *Is the system be able to give comprehensive feedback to the answers provided by the student? If the answer is not complete or wrong the system should not just tell that the answer is wrong but instead give the correct answer / full answer.*

The models were able to analyse the student's level of comprehension based on a series of questions and answers. They were able to provide followup questions for wrong questions and ask new questions on topics that were not covered before. The models used have a context window of 32'000 tokens (Mixtral [7]), 16'000 tokens (GPT-3.5-turbo[66]) and 128'000 tokens (GPT-4-turbo). Using the estimations under 6.6 and removing an estimate of 5000 words for the lecture script of that week, this equates to roughly 18500/6750/88750 maximum words fitting in a context window. When estimating a question to be 25 words, an average answer being 100 words and the feedback to the answer also being 100 words this would equate to roughly 80/30/400 question-answer-feedback loops that could be analysed for the level of the student's comprehension of that weeks lecture script.

---

[66] https://platform.openai.com/docs/models

9.2.3 Existing Products

Questions:

- *What are the pre-existing products/services (including sub-problems)?*

There are already existing products tackling the situation posed for this project:

- **Q-Chat by Quizlet**[67]**:** However this is not yet available in Switzerland and the demo does not show to which degrees the styles and difficulty level of questions can be adjusted.
- **Brian — Amplified Teaching**[68]**:** Brian is a learning app leveraging gamification and AI to turn study materials uploaded by teachers into digital learning materials to students.
- **Lernnavi**[69]: A learning app guiding students though their learning periods with gamification elements and boasting a number of exercise types.

Further products such as **code.org's teaching assistant**[70] focus more on subproblems (teaching assistant seems to be focusing on providing feedback). The vast majority of AI based teaching assistants focus on generating quizzes (example: **Quizbot**[71]) which allows to further customize question styles based on input files.
However these products except for Q-Chat focus more on assisting a teacher by automating individual tasks and are not in direct interaction with students.

9.2.4 Existing Research

Questions:

- *What research was already done in this field?*

Existing research around the topic of artificial intelligence in education can be extensively explored by examining literature on Intelligent Tutoring Systems (ITS). Recent review studies in this area provide a comprehensive starting point for understanding the current state of the field[9]. Of further interest would also be 'AI-Tutoring in Software Engineering Education' which focuses on a similar situation as this project but with the focus on software engineering lectures using GPT-3.5-turbo and how students interacted with it.[?]

---

[67] https://quizlet.com/qchat-personal-ai-tutor
[68] https://brian.study
[69] https://lernnavi.ch/
[70] https://code.org/ai/teaching-assistant
[71] https://quizbot.ai/

9.2.5 Other applications

Questions:

- *What are the current limitations of such a system?*
- *Is such a system scalable to other application -> Different subjects, creative classes, and visual questions?*
- *Does it work on the provided script?*

The models could be used in a similar fashion for other courses. In order to perform well they would still need the content of the course in text form. They could be used for text based creative classes. However visual interpretation within a text context are currently not possible for these models. See: Limitations9.4

9.2.6 Relevant Metrics

Questions:

- *What are relevant metrics to evaluate the systems?*

See Evaluation Criteria 5.5

9.3 Applications

All out of the box chat models were able to solve the subproblems provided to them. They are trained to interact in a dialogue and are moldable by forcing them to reference provided text files and providing system prompts defining the way they are expected to behave. This makes the highly flexible to adjust to other courses by adjusting the system prompts and provided files.

RAG convinces with its ease of use, low resource and time cost. If hosted fully on premises, it also grants the service provider a lot of control over the privacy of their data, as the data will not be used to adjust the weights of the model itself. These properties allow RAG to solve a wide range of business problems and especially shine in use cases where data privacy is a critical requirement.

9.4 Limitations

9.4.1 Models

The most significant limiting factor in utilizing large language models like GPT-4 and Mixtral-Instruct for applications involving lecture scripts is their inability to interpret images embedded within these scripts. To address this, a specialized approach involving a team of expert models would be required. This would involve separating the images from the text, converting the image content to text, and then integrating this text appropriately before running the chatbot. While this application is theoretically possible, it remains highly complex and presents substantial challenges.

Another limiting factor would be the associated cost and the potential abuse of it. Self hosted applications are quite resource intensive and external solutions like GPT have costs per token utilized. Ill meaning users could create huge amounts of requests to either drive up costs or clog up the system. Measures would need to be taken in the application to recognise and stop such behaviours.

Additionally sessions contexts have a limited size. If a student wishes to go over a topic over and over again, the application would find it difficult to recognise the current level of a user the more messages are sent. On top of that, cross referencing the users answer from another week would also not be possible as it would require another thread for that AI Assistant which does not have the context of other threads.
Chatbots tend to be people-pleasers. They tend to struggle to have an argument with their users and tend to try to incorporate the user's idea into it's response. This could lead to factually wrong responses. A safety measure would need to be included in the system prompt to make sure that the model refers to the lecturer when a user tries to argue.

9.4.2 Prototype

As mentioned in chapter 6.7, our prototype application is not without issues. Scaling up to a large number of users requires a re-write of parts of the data layer due to inefficient querying and it being deployed as a complete system on a single host puts the host under high load. Horizontal scaling is feasible with minor adjustments, as all technologies that were selected for the prototype are either capable of horizontal scaling or were even designed to be used this way.

The prototype currently only supports the use of PDF files for RAG and is not able to recognize or analyse images contained in them.

Additionally, there are still some known bugs in our prototype that require fixing.
Further detail on the limitations of the prototype application can be found in chapter 6.7 of this report.

9.5 Recommendations

Response quality wise, either GPT-4-turbo or Mixtral-8x7B-Instruct-v0.1 can be recommended to be used to solve the given tasks for text based lecture scripts as discuss above 9.1. We encourage to use extensive system prompting over finetuning to achieve specific answer styles.
The main deciding factors will likely be on the ease of implementation and scaling (GPT) versus having complete control and keeping your data inhouse (Mixtral) while having to provide RAG services yourself. Additional cost estimates for both approaches would need to be done aswell, including analysis of the newly released GPT-4o over GPT-4-turbo which would reduce costs per token by 50%.

If you are considering fine-tuning Mixtral models, we recommend using Axolotl if you have prior experience with this tool. Axolotl is well-suited for those already familiar with its features and workflow. For those who do not have experience with Axolotl, we suggest using LLama-Factory as an alternative with an easier learning curve (see Finetuning 5.3 for more details).

RAG already is an established practice in the industry, and the result of our experimentation only reinforces this notion. We highly recommend using RAG as an approach to create expert chat models.

In order for the prototype to be used at scale, we encourage to address its existing scalability concerns and to replace SQLModel with a more mature ORM framework. We also suggest further experimentation with the usage of fine-tuned models for this prototype application to leverage the advantages of both RAG and fine-tuning.

9.6 Reflection: Dominik Castelberg

I thoroughly enjoyed working on this project. Learning more about fine-tuning and RAG and how they can be embedded in a software system are invaluable for my desired career path and I am truly grateful for the opportunities we have received with this project. While the final prototype is not without flaws, I am overall happy with the result. I still wish I could have spent more time on it, as I feel like the last remaining issues could be resolved within a couple days.

Working under the supervision of Prof. Dr. Mitra Purandare was a pleasure. While she granted us many freedoms and let us explore a number of ideas on our own, she also was a consistent source of guidance and advice and, occasionally, a very much needed critical voice. We as a project team have grown greatly under her leadership and I am looking forward to keep working with her in future projects.

Linus and I are collaborating well. Our confidence each other's abilities allows us to assign large tasks to each other with the expectation for them to be completed with a high quality without the need of constant supervision aside from the occasional sync meeting. Working with Linus was efficient, enjoyable and the quality of his work is undeniable. I wish only the best for his professional future and am planning to keep in touch with him.

The rest of the project team had a more passive role, but was always helpful and eager to answer any questions we had. I am grateful for their contributions.

We anticipated that our projected schedule was a very optimistic estimate, especially considering that our work load outside of university can be quite high. Learning from our past mistakes, we planned for two weeks of buffer time at the end of the project, which turned out to become a life saver, as we both were hit with high external workloads over the course of this project. We once again barely made it to the finish line, and if we did not anticipate those struggles, we likely would have needed to reduce the project scope by a sizeable amount.

While I overall am happy with my work, I think that I dropped the ball with my insistence on using SQLModel as our ORM of choice, fearing that I would lose too much time refactoring the data layer, despite issues with the library piling up over the weeks. In hindsight, refactoring the data layer to use a different ORM or database adapter would have paid off with freed up development time later on, where bug investigations and bug fixing started to cost far more time than the refactoring process would have.

9.7 Reflection: Linus Flury

Overall, the project provided a valuable learning experience, allowing me to delve into a more practical approach within my chosen field of study. The continuous evolution of the field, highlighted by the release of new models of the same family that were evaluated in this project: GPT-4o and Mixtral-8x22B-Instruct-v0.1 during the project, made the experience even more engaging and relevant.

Working with Prof. Purandare was a pleasant and enriching experience. She provided additional ideas, introduced us to new developments and trends, and offered critical feedback that significantly improved our project. Her approach of granting us independence while still offering guidance was greatly appreciated. However, it seems we were unable to apply the lessons from previous projects regarding even distribution of time over the semester.

Collaborating with Dominik was smooth and efficient. We maintained open communication, were responsive to issues, and were able to work independently on our respective parts after planning the project together. Our respectful and coordinated work dynamic was a key factor in the project's success.

Despite these positives, we faced challenges such as suboptimal time management. Larger projects during the semester and at work made it difficult to consistently dedicate 23 hours per week to this project, resulting in a workload that had to be cleared in the final weeks of the semester. This could have been avoided with better time management.

Additionally, while the original project plan included generous time slots to account for unforeseen delays, specific tasks like fine-tuning required more time than anticipated. This realization came only towards the end of the allocated time in each block, leading to frustration during development, feeling like issues could be solved if you "just find the right way" but then needed to be circumvented because of time concerns. Examples for that would be the OpenAI Assistant SDK, automating the evaluation response generation through Jupyter notebooks.

In conclusion, the project was an invaluable experience, and I am excited to apply the knowledge and skills I gained to future endeavors. Setting smaller, fixed deadlines to ensure specific tasks were completed on time would have been useful to secure the general pace, however in the end it would still cost a similar amount of sleep over the semester as the work items depended on the completion of previous tasks. Additionally, being more mindful of available resources and justifying their use for tasks that require greater investment would be beneficial. Overall, the project was a success, and I am grateful for the opportunity to learn and grow through this experience.

9.8 Acknowledgements

We would like to thank Prof. Dr. Mitra Purandare for her leadership, constant support and guidance.

We would also like to thank Prof. David Kobler for the pleasant cooperation, the provided resources and the opportunity to work on this project. The trust he put into us is deeply appreciated.

10 Project Plan

10.1 Project Structure

10.1.1 Project Members

Role	Name	Comment
Client	Prof. David Kobler	Domain Expert
Advisor	Tobias Rothlin	Technical Expert
Supervisor	Prof. Dr. Mitra Purandare	
Developer, BA Author	Dominik Castelberg	
Developer, BA Author	Linus Flury	

10.1.2 Project Meetings

Weekly Meetings are held on Sunday 10:00 to 11:00. The current progress, what needs to be discussed at the supervisor meetings and the next steps are discussed. Additionally the tasks which require both developers during the next week are defined and scheduled to be done after the meeting.

Supervisor Meetings are held on Wednesday 12:00 to 13:00, either client can take part if needed and scheduled ahead of time.

10.1.3 Development Roles

As a development team of two, we only assign responsibilities to areas of work instead of assigning project roles. This leads to more efficient work on topics and produces less overhead.

During the initial setup phase, work on similar topic is done by both. Once the setup and model selection phase has passed (M2) the work areas are split until the implementation and model evaluation has been reached (M6).

General assigned areas are as follows:

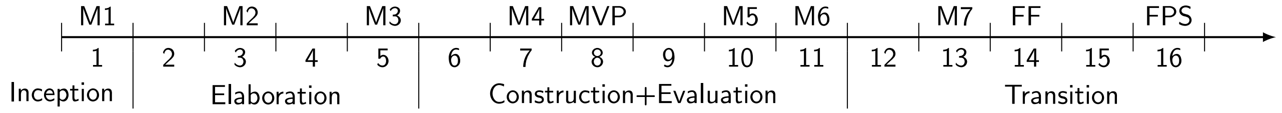
Dominik Castelberg:

- Prototype Architecture & Implementation
- R&D Retrieval Augmented Generation
- R&D Scalability

Linus Flury

- Use Cases
- Test Cases: Implementation + Evaluation
- R&D Model Fine-Tuning

10.2 Long Term Plan



10.3 Milestones

The milestones have been defined based on the project requirements. Each milestone has a deadline and a list of artifacts that must be produced in time.

M1, 27.02.2024 Initial Project Setup
Development process is defined, a long term plan exists, Architecture Sketch based on given requirements.
Artifacts: Documentation (including plan)

M2, 12.03.2024 Model Selection Process
Model requirements and evaluation criteria for model selection are defined, definition of allowed type of preprocessing
Artifacts: Documentation (including model requirements, evaluation criteria, model selection process, candidate models)

M3, 26.03.2024 Model Selection, Usecases
Models have been selected, definition of model variations to check, use cases are defined per subtask.
Artifacts: Documentation (including exact documentation of selected models, model variations, usecases documented in appropriate detail)

M4, 09.04.2024 Subtask Evaluation Criteria
Evaluation criteria for subtasks are defined, test cases are defined, models are deployed.
Artifacts: Documentation (including evaluation criteria for subtasks)

MVP, 17.04.2024 Subtask Evaluation Criteria
Midway presentation MVP
Artifacts: Presentation: Application MVP, first test cases

M5, 30.04.2024 Evaluation
Test cases are run over all models and subtasks, results are documented.
Artifacts: Documentation (including test cases, results)

M6, 07.05.2024 Implementation of PoC, Interpretation
End of development of application, evaluation results are interpreted.
Artifacts: Documentation (including final PoC architecture, interpretation of results.), PoC codebase

M7, 21.05.2024 Scalability
Scalability of the application is tested, possible suggestions for scaling are documented.
Artifacts: Documentation (including scalability test cases, suggestions for scaling)

FF, 28.05.2024 Feature Freeze
All features in the planned project scope are implemented. From this day forward, only bugfixes are allowed.
Artifacts: None

FPS, 14.06.2024 Final Project Submission
Both the application and the documentation are finalized. Presentation, Abstract and Poster are finished.
Artifacts: Documentation, Codebase, Abstract (Poster) (10.06.2024), signed documents

FPS, 21.06.2024 Project Presentation
Slides are finished and prepared for presentation.
Artifacts: Presentation Slides

10.4 Tools

- **Python 3**[72] was used as the programming language for our evaluation workflows, the test environment and the graphical outputs. It was used as all team members were experienced in using it and it is supported by Jupyter Notebooks.
- **Jupyter Notebooks**[73] were used in order to present the workflow in a intuitive manner, making the workflow easier to understand, follow and debug. They allow easy sharing between the team members and allow simple duplication of similar notebooks. Matplotlib plots can be shown within the notebook, allowing for immediate visual feedback in the same notebook. Both team members had prior experience in their usage.
- **LangChain**[74] is a python library that was used to handle the connection to the LLMs' APIs, delivering them prompt and parsing the output. It was used based on prior experience of the team members.
- **LLaMA-Factory**[75] was used to finetune Mixtral. For further details see: Finetuning 5.3.2
- **Docker**[76] was used to make the prototype application portable and was used in model fine tuning
- **GitLab** was used as a collaboration platform, allowing for version controlled cooperation on the same repository and development in separate branches for each feature, allowing independent work. It was chosen over GitHub[77] to keep the repository in-house, as an instance of GitLab is hosted by OST[78]. Both team members had prior experience in its usage.

---

[72] https://www.python.org/
[73] https://jupyter.org/
[74] https://python.langchain.com
[75] https://github.com/hiyouga/LLaMA-Factory
[76] https://www.docker.com/
[77] https://github.com/
[78] https://gitlab.ost.ch/

- **LaTeX**[79] was used to build the project report. While typst[80] was used for the Studienarbeit, both team members felt more comfortable using LaTeXeven though it is more complex to use. At the start of the project, typst was still in beta. Some design features were very simple to implement, however the fine tuning and achieving more complex features were either not possible or not feasible to learn in the allotted time. LaTeXwas preferred for this Bachelor Thesis because both team members agreed to feel more confident reaching a good quality documentation design using it.
- **Draw.IO**[81] was used to create flow- and system-diagrams.
- **Overleaf**[82] was used a collaboration tool for writing the documentation using LaTeX. It's real time updates allow for more efficient parallel work. Having a commit based collaboration platform like GitLab or GitHub would increase the robustness against accidental data loss and would enforce the four-eyes model, merge conflicts, especially in the final stages of the documentation, while trying to adjust and update documentation files can lead to a lot of additional stress in the final stages. This could be preventable by having a rigorous and detailed documentation setup and commit/merge discipline. However the team members agreed that that standard will not be achievable for the entirety of the project. This led to the agreement to use Overleaf.
- **Jira**[83] was used to for workload planning and time tracking.

---

[79] https://www.latex-project.org/
[80] https://typst.app/
[81] https://www.drawio.com/
[82] https://www.overleaf.com/
[83] https://www.atlassian.com/software/jira

10.5 Time Recording

Time recording was done with Jira tasks. In the end we recorded a total of 728 hours on the project. 393 hours were recorded by Dominik Castelberg, 335 hours were recorded by Linus Flury.

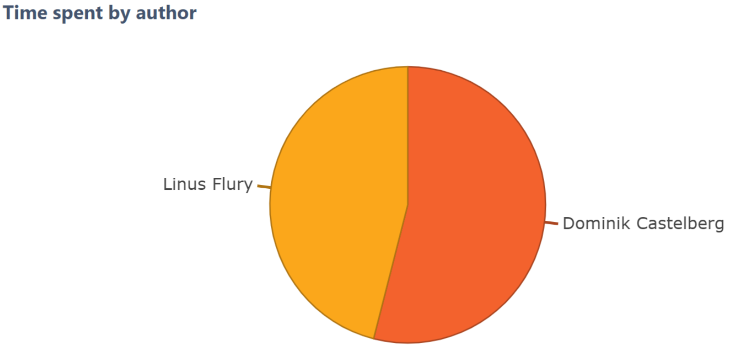


Figure 40: Time Recordings

11 Attachments

11.1 Evaluation Text Results

11.1.1 Use Case 1: Question Generation

User prompt:
Generiere 20 Fragen basierend auf dem bereitgestellten Dokument über politische Rechte in der Schweiz. Die Fragen sollen im Imperativ und mindestens auf der Ebene des Verstehens nach Blooms Taxonomie gestellt werden. Die Ausgabe beinhaltet nur die Fragen.

GPT 3.5
Vergleichen Sie fakultative und obligatorische Referenden hinsichtlich ihrer Auslöser und bindenden Wirkung.
Erläutern Sie den Unterschied zwischen aktivem und passivem Wahlrecht.
Diskutieren Sie die Bedeutung des Petitionsrechts in einer Demokratie.
Welche Vorteile bringt das Prinzip der Verhältniswahl im Vergleich zur Mehrheitswahl?
Beurteilen Sie die Bedeutung der Mindestanzahl an Unterzeichnenden für Wahlvorschläge.
Argumentieren Sie für die Einführung des Wahlrechts für Ausländerinnen und Ausländer in der Schweiz.
Beschreiben Sie die Bedeutung der Unterzeichnung von Initiativ- und Referendumsbegehren für die direkte Demokratie.
Vergleichen Sie die Anforderungen an die Mindestanzahl an Unterzeichnenden für Wahlvorschläge in verschiedenen Kantonen.
Diskutieren Sie die Bedeutung der Stabilität in einem Wahlsystem.
Warum ist die Unterscheidung zwischen Aktiv- und Passivwahlrecht in einer Demokratie wichtig?
Wie können verschiedene Wahlsysteme die Repräsentation der Wählerinnen und Wähler beeinflussen?
Erklären Sie den Unterschied zwischen obligatorischen und fakultativen Referenden anhand konkreter Beispiele.
Diskutieren Sie die Rolle von Wahlkreisen in einem Wahlsystem.
Welche Auswirkungen kann ein unverstandenes Wahlsystem auf die Wahlergebnisse haben?
Beurteilen Sie die Bedeutung der Wahl- und Abstimmungsfreiheit in einer Demokratie.
Vergleichen Sie die Rechte und Pflichten im Zusammenhang mit dem Verfassungsreferendum und dem Gesetzesreferendum.
Diskutieren Sie die Bedeutung der Repräsentation im Wahlsystem.
Bewerten Sie die Bedeutung von Volksinitiativen für politische Entwicklungen in der Schweiz.
Erklären Sie, warum das Petitionsrecht ein wichtiges demokratisches Instrument ist.
Diskutieren Sie, wie das Wahlrecht und das Wahlsystem die politische Landschaft eines Landes prägen können.

GPT 4

Vergleichen Sie die Auswirkungen von obligatorischen und fakultativen Referenden auf die politische Teilhabe der Bürger.

Erörtern Sie, welche Rolle direkte Gegenentwürfe in der Demokratie spielen und wie sie die Entscheidungsfindung im Volk beeinflussen kännten.

Diskutieren Sie die Vor- und Nachteile von Mehrheitswahlrecht und Verhältniswahlrecht unter besonderer Berücksichtigung der politischen Repräsentation kleiner Parteien.

Argumentieren Sie, welche Bedeutung die freien und starren Listen in Proporzwahlen für die politische Willensbildung haben.

Bewerten Sie die Bedeutung der geheimen Wahl im Kontext der Einheit der Materie und der korrekten Information der Stimmberechtigten.

Analysieren Sie, inwiefern das Wahlalter und Wahlberechtigungen in verschiedenen Ländern die politische Gleichheit beeinflussen können.

Stellen Sie dar, wie Änderungen in den Voraussetzungen für das Wahlrecht die politische Landkarte verändern könnten.

Beschreiben Sie die Auswirkungen von Wahlkreisgrenzen auf die politische Landschaft und Repräsentation in einem Mehrebenensystem.

Diskutieren Sie die potenziellen Auswirkungen der Einführung eines bedingten Rückzugs von Volksinitiativen auf die Gesetzgebung und politische Teilhabe.

Vergleichen Sie das aktive und passive Wahlrecht in Bezug auf ihre Bedeutung für die politische Gleichheit und Partizipation.

Bewerten Sie die Effektivität von Initiativ- und Referendumsrechten in der direkten Demokratie am Beispiel des Schweizer Systems.

Analysieren Sie, wie diverse Mehrheiten (Volks- und Ständemehr) in Verfassungsfragen die politischen Ergebnisse beeinflussen können.

Erörtern Sie, wie die Wahl- und Abstimmungsfreiheit die Legitimität politischer Entscheidungen sicherstellt.

Bewerten Sie, wie das Konsensprinzip in Wahlentscheidungen die politische Landschaft formt.

Diskutieren Sie, welche Rolle die Information der Wählerschaft durch staatliche Behörden spielt und welche Grenzen dabei beachtet werden müssen.

Argumentieren Sie, unter welchen Bedingungen konsultative Plebiszite eine wirkungsvolle Form der politischen Konsultation sein können.

Vergleichen Sie Ausländerwahlrecht auf kantonaler und kommunaler Ebene und dessen Einfluss auf politische Prozesse und Integration.

Diskutieren Sie die potenziellen Auswirkungen einer Reform des Wahlsystems auf die politische Stabilität und Parteienlandschaft.

Analysieren Sie die Rolle von Volksmehr und Ständemehr bei der Annahme von Verfassungsänderungen in einer föderalistischen Struktur.

Bewertet die Wichtigkeit und mögliche Probleme der korrekten Ermittlung von Wahl- und Abstimmungsergebnissen.

Mixtral

Vergleichen Sie die Voraussetzungen für die politischen Rechte auf Bundesebene mit denen auf kantonaler oder kommunaler Ebene. Gibt es Unterschiede? Wenn ja, welche?

Diskutieren Sie die Bedeutung des aktiven und passiven Wahlrechts in einer demokratischen Gesellschaft. Welches ist wichtiger? Begründen Sie Ihre Meinung.

Vergleichen Sie die Möglichkeiten des aktiven Wahlrechts auf Bundes- und Kantonsebene. Welche Unterschiede und Gemeinsamkeiten finden Sie?

Nehmen Sie eine Stellungsposition dazu ein, ob das Petitionsrecht ein wirksames Mittel der politischen Partizipation ist oder nicht. Unterstützen Sie Ihre Position mit Argumenten und Beispielen aus der Schweiz oder aus anderen Ländern.

Erörtern Sie die Vor- und Nachteile eines verbindlichen (dezisiven) Plebiszits und eines nicht-bindenden (konsultativen) Plebiszits. Geben Sie Beispiele an, in welchen Situationen jede dieser Varianten sinnvoll sein könnte.

Vergleichen Sie Mehrheits- und Verhältniswahlsysteme hinsichtlich Repräsentation, Stabilität und Einfachheit. Geben Sie jeweils ein oder zwei Beispiele aus verschiedenen Ländern an.

Anhand des Schweizer Beispiels erläutern Sie kurz die Begriffe Majorz- und Proporzsysteme. Erklären Sie anschließend, wie die Nationalratswahlen in der Schweiz durchgeführt werden, und bewerten Sie, inwieweit das Wahlsystem die Ziele Repräsentation, Stabilität und Einfachheit erfüllt.

Welche Bedeutung haben Wahlkreise bei Wahlen und wie variieren sie in der Schweiz bei den Nationalrats- und Kantonsratswahlen?

Nennen Sie Vor- und Nachteile von Einerwahlkreisen (Wahlkreise mit nur einem Sitz) und Mehrpersonenwahlkreisen bei Wahlen. Welche Art von Wahlkreis wird in der Regel gebraucht, um die Repräsentations- und Stabilitätsziele eines Wahlsystems zu erfüllen?

Beschreibe das "Two-round system" (absolutes Mehrheitswahlrecht in Einerwahlkreisen) und gib ein Beispiel an, in dem dieses System verwendet wird.

Vergleiche und erkläre die Unterschiede zwischen absolutem Mehr und relativen Mehr in Mehrheitswahlrecht in kleinen Mehrpersonenwahlkreisen.

Vergleiche starre Listen und freie Listen im Hinblick auf das Maß an Flexibilität und Einfluss, das Wählerinnen und Wählern geboten wird.

Beschreibe das Kumulieren und Panaschieren im Rahmen des Prinzips der freien Listen. Erkläre, wie diese Praktiken dem Wähler Einfluss auf die Zuteilung von Sitzen geben.

Vergleichen Sie die "natürliche Wahlhürde" in Proporzwahlsystemen mit Sperrklauseln als Wahlhürden festgelegten Staaten. Was sind die Vor- und Nachteile dieser verschiedenen Wahlhürden?

Diskutieren Sie die Auswirkungen der Sperrklausel von 5% in Wahlen zum Deutschen Bundestag. Warum wurde diese Sperrklausel eingeführt und wie wirkt sie sich auf die politische Landschaft in Deutschland aus?

Analysieren Sie den Unterschied zwischen dem "mechanischen Effekt" und dem "psychologischen Effekt" von Wahlsystemen. Wie beeinflussen diese Effekte die Wahlchancen neuer Parteien?

Diskutieren Sie die Vor- und Nachteile von Listenverbindungen in Bezug auf die politische Repräsentation von Minderheiten. Wie können Listenverbindungen die Einflussnahme kleinerer Parteien fördern oder behindern?

Anhand konkreter Beispiele erklären Sie, wie die Listenverbindung die Chancen auf einen Sitzgewinn und die Auswertung von Reststimmen beeinflussen kann.

Vergleichen Sie das obligatorische Referendum mit dem fakultativen Referendum in der Schweiz. Welche Bedingungen müssen erfüllt sein, damit ein Referendum automatisch ausgelöst wird?

Diskutieren Sie die Mehrheitsverhältnisse bei Volksabstimmungen in der Schweiz. Warum sind bei einigen Vorlagen nur das Volksmehr, bei anderen aber das Volks- und Ständemehr erforderlich? Wie würden Sie die aktuelle Regelung beurteilen?

GPT 3.5 finetuned

Nenne die vier Kategorien von Menschenrechten!
Nenne die Rechte, die in der ersten Kategorie zusammengefasst werden!
Nenne die Rechte, die in der zweiten Kategorie zusammengefasst werden!
Nenne die Rechte, die in der dritten Kategorie zusammengefasst werden!
Nenne die Rechte, die in der vierten Kategorie zusammengefasst werden!
Definiere individuelle Menschenrechte!
Definiere kollektive Menschenrechte!
Was ist der Unterschied zwischen individuellen und kollektiven Menschenrechten?
In welcher Dimension können individuelle und kollektive Menschenrechte unterschieden werden?
Nenne Beispiele für individuelle Menschenrechte!
Nenne Beispiele für kollektive Menschenrechte!
Erkläre den Begriff der sozialen Menschenrechte!
Nenne Beispiele für soziale Menschenrechte!
Was sind bürgerliche und politische Rechte laut der vierten Kategorie?
Kannst du die verschiedenen Generationen von Menschenrechten erklären?
Was sind die Unterschiede zwischen diesen Generationen von Menschenrechten?
Können individuelle und kollektive Menschenrechte gleichzeitig gelten?
Was sind die sogenannten doppelten Menschenrechte?
Welche Rechte bezeichnet man als reine Bürgerrechte, die in der ersten Kategorie enthalten sind?
Was sind indirekte Menschenrechte und wie werden sie auch genannt?

Mixtral finetuned

Benennen Sie die direktdemokratischen Rechte auf Bundesebene gemäss Abbildung! (Quelle fehlt)
Erläutern Sie das passive Wahlrecht!
Was ist das aktive Wahlrecht?
Was ist der Grundsatz der Einheit der Materie bei Abstimmungen und Wahlen?
Was ist das Recht der Stimmberechtigten, eine Petition einzureichen?
Nennen Sie die Voraussetzungen zur Ausübung politischer Rechte auf Bundesebene!
Was ist das fakultative Gesetzesreferendum?
Was bedeutet die Wahl- und Abstimmungsfreiheit?
Was ist das Stimmrecht?
Darf der Bund fakultative Gesetzesinitiativen einreichen?
Was versteht man unter der Einheit der Materie?
Welches spezifische politische Recht wurde in der Schweiz 1892 auf Bundesebene eingeführt?
Welche Schwierigkeit kann beim bedingten Rückzug einer Volksinitiative auftreten?
Was sind obligatorische Referenden?
Was bedeutet das passive Wahlrecht im Bund?
Wann werden Plebisziten konsultativ und wann devisiv genannt?
Welche politischen Rechte stehen Schweizerinnen und Schweizern laut Art. 136 Abs. 1 BV in Bundesangelegenheiten zu?
Wann wird die Einheit der Materie gewahrt?
Was ist das Stimmrecht?
Was regelt Art. 139b BV?

11.1.2 Use Case 2: Answer Evaluation

User prompt:

Frage A: Was sind die formalen Voraussetzungen für die Ausübung politischer Rechte auf Bundesebene in der Schweiz?
Antwort A1: Zu den Voraussetzungen gehören das Schweizer Bürgerrecht, das Erreichen des 18. Lebensjahres und das Fehlen verfassungsmäßig definierter Ausschlussgründe.
Antwort A2: Die Voraussetzungen umfassen das Schweizer Bürgerrecht und das 18. Lebensjahr.
Antwort A3: Alle Einwohner in der Schweiz, die mindestens 16 Jahre alt sind, können an Wahlen teilnehmen.
Frage B: Was versteht man unter dem aktiven und passiven Wahlrecht?
Antwort B1: Das aktive Wahlrecht ermöglicht es, an Wahlen teilzunehmen, während das passive Wahlrecht das Recht beinhaltet, gewählt zu werden.
Antwort B2: Aktives Wahlrecht bedeutet, für politische Ämter zu kandidieren, und passives Wahlrecht erlaubt die Teilnahme an der Wahl.
Antwort B3: Aktives Wahlrecht bedeutet, wählen zu dürfen.
Frage C: Was beschreibt ein obligatorisches Referendum?
Antwort C1: Ein obligatorisches Referendum kann von 50.000 Bürgern ausgelöst werden, wenn sie dies fordern.
Antwort C2: Ein obligatorisches Referendum findet statt, wenn die Regierung es bestimmt.
Antwort C3: Ein obligatorisches Referendum tritt bei wichtigen rechtlichen Änderungen,wie Verfassungsänderungen, automatisch in Kraft und muss vom Volk bestätigt werden.

GPT 3.5

Frage A: Was sind die formalen Voraussetzungen für die Ausübung politischer Rechte auf Bundesebene in der Schweiz?

Antwort A1: Richtig. Die formalen Voraussetzungen für die Ausübung politischer Rechte auf Bundesebene in der Schweiz umfassen das Schweizer Bürgerrecht, das Erreichen des 18. Lebensjahres und das Fehlen verfassungsmäßig definierter Ausschlussgründe.
Antwort A2: Falsch. Diese Antwort ist unvollständig, da sie nicht alle formalen Voraussetzungen für die Ausübung politischer Rechte auf Bundesebene in der Schweiz erwähnt. Es fehlt das Fehlen verfassungsmäßig definierter Ausschlussgründe.
Antwort A3: Falsch. Diese Antwort ist nicht korrekt, da nicht alle Einwohner in der Schweiz automatisch an Wahlen teilnehmen können. Es sind spezifische Voraussetzungen wie das Schweizer Bürgerrecht und das Erreichen des 18. Lebensjahres erforderlich.

Frage B: Was versteht man unter dem aktiven und passiven Wahlrecht?

Antwort B1: Richtig. Das aktive Wahlrecht ermöglicht es, an Wahlen teilzunehmen, während das passive Wahlrecht das Recht beinhaltet, gewählt zu werden.
Antwort B2: Falsch. Diese Antwort ist teilweise korrekt, da sie das aktive Wahlrecht korrekt beschreibt, aber das passive Wahlrecht unvollständig wiedergibt. Das passive Wahlrecht bezieht sich darauf, gewählt zu werden, nicht nur an der Wahl teilzunehmen.
Antwort B3: Falsch. Diese Antwort ist nicht korrekt, da das aktive Wahlrecht mehr beinhaltet als nur das Recht zu wählen. Es beinhaltet auch das Recht, für politische Ämter zu kandidieren.

Frage C: Was beschreibt ein obligatorisches Referendum?

Antwort C1: Richtig. Ein obligatorisches Referendum tritt bei wichtigen rechtlichen Änderungen, wie Verfassungsänderungen, automatisch in Kraft und muss vom Volk bestätitritt bei wichtigen rechtlichen Änderungen, wie Verfassungsänderungen, automatisch in Kraft und muss vom Volk bestätigt werden.gt werden.
Antwort C2: Falsch. Ein obligatorisches Referendum wird nicht von der Regierung bestimmt, sondern tritt automatisch bei bestimmten Bedingungen in Kraft.
Antwort C3: Falsch. Diese Antwort beschreibt nicht korrekt ein obligatorisches Referendum. Es bezieht sich nicht nur auf Verfassungsänderungen, sondern auch auf andere spezifische Fälle, wie Bundesbeschlüsse über bestimmte Angelegenheiten.

Falls Sie weitere Fragen haben oder zusätzliche Informationen benötigen, lassen Sie es mich bitte wissen!

GPT 4

Frage A: Was sind die formalen Voraussetzungen für die Ausübung politischer Rechte auf Bundesebene in der Schweiz?

Antwort A1: Zu den Voraussetzungen gehören das Schweizer Bürgerrecht, das Erreichen des 18. Lebensjahres und das Fehlen verfassungsmäßig definierter Ausschlussgründe.
Bewertung: Korrekt. Diese Antwort ist vollständig und stimmt mit den dokumentierten Anforderungen überein, die das Schweizer Bürgerrecht, die Vollendung des 18. Lebensjahres und das Fehlen von Ausschlussgründen durch Geistesschwäche und ganze Beistandschaft erfordern.

Antwort A2: Die Voraussetzungen umfassen das Schweizer Bürgerrecht und das 18. Lebensjahr.
Bewertung: Teilweise richtig, jedoch unvollständig. Diese Antwort erfasst die zentralen Voraussetzungen, lässt aber das wichtige Kriterium der Ausschlussgründe weg.

Antwort A3: Alle Einwohner in der Schweiz, die mindestens 16 Jahre alt sind, können an Wahlen teilnehmen.
Bewertung: Falsch. Diese Antwort ignoriert sowohl die Erfordernisse des Schweizer Bürgerrechts als auch des korrekten Alters von 18 Jahren sowie die verfassungsmäßigen Ausschlussgründe.

Frage B: Was versteht man unter dem aktiven und passiven Wahlrecht?

Antwort B1: Das aktive Wahlrecht ermöglicht es, an Wahlen teilzunehmen, während das passive Wahlrecht das Recht beinhaltet, gewählt zu werden.
Bewertung: Korrekt. Diese Antwort definiert sowohl das aktive als auch das passive Wahlrecht korrekt gemäß den dargelegten Definitionen.

Antwort B2: Aktives Wahlrecht bedeutet, für politische Ämter zu kandidieren, und passives Wahlrecht erlaubt die Teilnahme an der Wahl.
Bewertung: Falsch. Diese Antwort vertauscht die Definitionen von aktivem und passivem Wahlrecht.

Antwort B3: Aktives Wahlrecht bedeutet, wählen zu dürfen.
Bewertung: Teilweise richtig, aber unvollständig. Die Antwort adressiert korrekt das aktive Wahlrecht, ignoriert jedoch eine Erklärung des passiven Wahlrechts.

Frage C: Was beschreibt ein obligatorisches Referendum?

Antwort C1: Ein obligatorisches Referendum kann von 50.000 Bürgern ausgelöst werden, wenn sie dies fordern.
Bewertung: Falsch. Dies beschreibt die Bedingungen für ein fakultatives Referendum, nicht ein obligatorisches.

Antwort C2: Ein obligatorisches Referendum findet statt, wenn die Regierung es bestimmt.
Bewertung: Falsch. Ein obligatorisches Referendum wird nicht von der Regierung bestimmt, sondern tritt bei spezifischen gesetzlichen oder verfassungsrechtlichen Änderungen automatisch in Kraft.

Antwort C3: Ein obligatorisches Referendum tritt bei wichtigen rechtlichen Änderungen, wie Verfassungsänderungen, automatisch in Kraft und muss vom Volk bestätigt werden.
Bewertung: Korrekt. Diese Antwort beschreibt korrekt die Bedingungen und Umstände, unter denen ein obligatorisches Referendum in Kraft tritt.

Folgefragen und Korrekturen:

Zu Antwort A2: Wie werden die Ausschlussgründe für das Wahlrecht auf Bundesebene genau definiert? Sie könnten sich auf die verfassungsmäßig bedingten Ausschlussgründe in Zukunft beziehen, um so eine umfassende Kenntnis über die Wahlberechtigung in der Schweiz zu erlangen.

Zu Antwort B2: Können Sie die korrekten Definitionen von aktivem und passivem Wahlrecht erwähnen? Dies hilft, das grundsätzliche Verständnis beider Begriffe zu klären, wie im Dokument beschrieben.

Zu Antwort C1: Was sind die Hauptunterschiede zwischen einem obligatorischen und einem fakultativen Referendum? Eine vertiefte Betrachtung dieser beiden Referendumstypen könnte die fundamentalen Unterschiede verdeutlichen, die für das politische System der Schweiz von Bedeutung sind.

Mixtral

Frage A: Was sind die formalen Voraussetzungen für die Ausübung politischer Rechte auf Bundesebene in der Schweiz?

Antwort A1: Zu den Voraussetzungen gehören das Schweizer Bürgerrecht, das Erreichen des 18. Lebensjahres und das Fehlen verfassungsmäßig definierter Ausschlussgründe. Diese Antwort ist korrekt, weil sie alle relevanten Voraussetzungen für die Ausübung politischer Rechte auf Bundesebene in der Schweiz enthält.

Antwort A2: Die Voraussetzungen umfassen das Schweizer Bürgerrecht und das 18. Lebensjahr. Diese Antwort ist teilweise korrekt, da sie zwei der drei Voraussetzungen nennt, aber die verfassungsmäßig definierten Ausschlussgründe nicht erwähnt.

Antwort A3: Alle Einwohner in der Schweiz, die mindestens 16 Jahre alt sind, können an Wahlen teilnehmen. Diese Antwort ist falsch, weil die Voraussetzungen für die Ausübung politischer Rechte auf Bundesebene in der Schweiz das Schweizer Bürgerrecht und nicht einfach nur den Wohnsitz in der Schweiz erfordern.

Frage B: Was versteht man unter dem aktiven und passiven Wahlrecht?

Antwort B1: Das aktive Wahlrecht ermöglicht es, an Wahlen teilzunehmen, während das passive Wahlrecht das Recht beinhaltet, gewählt zu werden. Diese Antwort ist korrekt, weil sie die beiden Bedeutungen von aktivem und passivem Wahlrecht korrekt wiedergibt.

Antwort B2: Aktives Wahlrecht bedeutet, für politische Ämter zu kandidieren, und passives Wahlrecht erlaubt die Teilnahme an der Wahl. Diese Antwort ist falsch, weil sie die beiden Bedeutungen von aktivem und passivem Wahlrecht verwechselt.

Antwort B3: Aktives Wahlrecht bedeutet, wählen zu dürfen. Diese Antwort ist teilweise korrekt, da das aktive Wahlrecht tatsächlich das Recht bedeutet, an Wahlen teilzunehmen, aber sie lässt das passives Wahlrecht unerwähnt.

Frage C: Was beschreibt ein obligatorisches Referendum?

Antwort C1: Ein obligatorisches Referendum kann von 50.000 Bürgern ausgelöst werden, wenn sie dies fordern. Diese Antwort ist falsch, weil ein obligatorisches Referendum nicht von Bürgern ausgelöst werden kann, sondern automatisch stattfindet, wenn bestimmte Bedingungen erfüllt sind.

Antwort C2: Ein obligatorisches Referendum findet statt, wenn die Regierung es bestimmt. Diese Antwort ist falsch, weil die Regierung kein obligatorisches Referendum auslösen kann.

Antwort C3: Ein obligatorisches Referendum tritt bei wichtigen rechtlichen Änderungen, wie Verfassungsänderungen, automatisch in Kraft und muss vom Volk bestätigt werden. Diese Antwort ist korrekt, weil sie die Bedeutung und den Zweck eines obligatorischen Referendums korrekt wiedergibt.

Follow-up question: Was ist der Unterschied zwischen einem obligatorischen Referendum und einem fakultativen Referendum? Erklären Sie auch, unter welchen Bedingungen ein obligatorisches Referendum in der Schweiz stattfindet.

Erklärung: Ein obligatorisches Referendum ist eine Form der direkten Demokratie in der Schweiz, bei der eine Abstimmung über eine bestimmte Vorlage automatisch stattfindet, wenn bestimmte Bedingungen erfüllt sind, wie zum Beispiel bei Verfassungsänderungen. Ein fakultatives Referendum hingegen kann auf Begehren von mindestens 50'000 Stimmberechtigten oder von mindestens acht Kantonen durchgeführt werden und bezieht sich auf Bundesgesetze und gewisse Staatsverträge. Im Gegensatz zum obligatorischen Referendum erfordert das fakultative Referendum in der Schweiz also ein aktives Begehren seitens der Bürger oder der Kantone.

GPT 3.5 finetuned

Frage A: Was sind die formalen Voraussetzungen für die Ausübung politischer Rechte auf Bundesebene in der Schweiz?

Antwort A1: Richtig. Die Antwort enthält die korrekten formalen Voraussetzungen: Schweizer Bürgerrecht, das Erreichen des 18. Lebensjahres und das Fehlen verfassungsmäßig definierter Ausschlussgründe.
Antwort A2: Falsch. Diese Antwort ist nicht vollständig, da sie nicht das Fehlen der verfassungsmäßig definierten Ausschlussgründe erwähnt.
Antwort A3: Falsch. Diese Antwort ist inkorrekt, da nicht alle Einwohner in der Schweiz, die mindestens 16 Jahre alt sind, an Wahlen teilnehmen können. Es sind spezifische Voraussetzungen wie das Schweizer Bürgerrecht und das Erreichen des 18. Lebensjahres erforderlich.

Frage B: Was versteht man unter dem aktiven und passiven Wahlrecht?

Antwort B1: Richtig. Das aktive Wahlrecht ermöglicht es, an Wahlen teilzunehmen, während das passive Wahlrecht das Recht beinhaltet, gewählt zu werden.
Antwort B2: Falsch. Diese Antwort ist nicht korrekt, da sie die Begriffe des aktiven und passiven Wahlrechts vertauscht.
Antwort B3: Falsch. Diese Antwort beschreibt nicht korrekt das aktive Wahlrecht.

Frage C: Was beschreibt ein obligatorisches Referendum?

Antwort C1: Falsch. Diese Antwort ist nicht korrekt. Ein obligatorisches Referendum tritt bei wichtigen rechtlichen Änderungen, wie Verfassungsänderungen, automatisch in Kraft und muss vom Volk bestätigt werden.
Antwort C2: Falsch. Diese Antwort ist nicht korrekt. Ein obligatorisches Referendum wird nicht von 50.000 Bürgern ausgelöst, sondern tritt automatisch bei bestimmten wichtigen rechtlichen Änderungen in Kraft.
Antwort C3: Richtig. Ein obligatorisches Referendum tritt bei wichtigen rechtlichen Änderungen, wie Verfassungsänderungen, automatisch in Kraft und muss vom Volk bestätigt werden.

Mixtral finetuned

Die korrekten Antworten auf die Fragen sind wie folgt:

Frage A:

Antwort A1 ist korrekt: Das Schweizer Bürgerrecht, das Erreichen des 18. Lebensjahres und das Fehlen verfassungsmäßig definierter Ausschlussgründe sind Voraussetzungen für die Ausübung politischer Rechte auf Bundesebene in der Schweiz.
Antwort A2 ist teilweise korrekt: Das Schweizer Bürgerrecht und das Erreichen des 18. Lebensjahres sind tatsächlich Voraussetzungen.
Antwort A3 ist falsch: Nicht alle Einwohner in der Schweiz, die mindestens 16 Jahre alt sind, können an Wahlen teilnehmen. Das aktive Wahlrecht im Bund erfordert das Schweizer Bürgerrecht, das Erreichen des 18. Altersjahres und das Fehlen der verfassungsmässig definierten Ausschlussgründe.

Frage B:

Antwort B1 ist korrekt: Das aktive Wahlrecht ermöglicht es, an Wahlen teilzunehmen, während das passive Wahlrecht das Recht beinhaltet, gewählt zu werden.
Antwort B2 ist falsch: Das aktive Wahlrecht bedeutet nicht, für politische Ämter zu kandidieren. Es beinhaltet das Recht, an einer Wahl teilzunehmen, also Kandidaten oder Parteien zu wählen.
Antwort B3 ist falsch: Das aktive Wahlrecht bezieht sich nicht darauf, wählen zu dürfen. Es bedeutet, das Recht zu haben, an einer Wahl teilzunehmen, also Kandidaten oder Parteien zu wählen.

Frage C:

Antwort C1 ist korrekt: Ein obligatorisches Referendum tritt bei wichtigen rechtlichen Änderungen, wie Verfassungsänderungen, automatisch in Kraft und muss vom Volk bestätigt werden. Es kann unter bestimmten Bedingungen automatisch ausgelöst werden.
Antwort C2 ist falsch: Ein obligatorisches Referendum wird nicht von der Regierung bestimmt, sondern tritt bei bestimmten Bedingungen automatisch in Kraft.
Antwort C3 ist falsch: Ein obligatorisches Referendum bezieht sich nicht nur auf Verfassungsänderungen, sondern auch auf einige andere Fälle. Es wird nicht automatisch ausgelöst, wenn die Regierung es bestimmt, sondern unter festgelegten Bedingungen.

11.1.3 Use Case 3: Question Generation

User prompt:

Frage1: Weshalb dürfen im Kanton Glarus 16-Jährige an den Ständeratswahlen teilnehmen?
Antwort1: Weiss ich nicht.
Frage2: Was sind politische Rechte?
Antwort2: Keine Ahnung.
Frage3: Welche politischen Rechte gibt es in der Schweiz auf Bundesebene?
Antwort3: Frei zu leben.
Frage4: Begründe warum eine Listenverbindung bei Proporzwahlen für eine grosse Partei mit kleineren Parteien Sinn machen könnte.
Antwort 4: Bessere Ausstrahlungskraft.
Frage5: Erläutere den Grundsatz der Einheit der Materie.
Antwort5: Die Einheit der Materie besagt, dass in einer Vorlage nur Fragen umfasst werden dürfen, welche einen inneren Zusammenhang haben.
Frage 6: Benenne den Unterschied zwischen folgenden Situationen: A) Ein Wähler streicht ein Kandidat von seiner Liste B) Ein Wähler streicht ein Kandidat von seiner Liste und ersetzt die Person durch einen anderen Kandidaten derselben Liste.
Antwort6: Für die Liste macht dies kein Unterschied, die leere Stimme wird dennoch für die Liste gezählt. Für den Kandidaten macht dies aber ein Unterschied, da er so eine Stimme mehr bekommen hat und so bessere Chancen auf ein Sitz hat.

GPT 3.5

Der Student hat Schwierigkeiten, grundlegende Aspekte der schweizerischen Politiknovellen zu verstehen. Hier ist die Auswertung der Antworten:

1. Zur Frage nach dem Stimmrecht von 16-Jährigen im Kanton Glarus: Die Antwort "Weiß ich nicht" zeigt, dass der Student dieses spezifische Thema nicht verstanden hat.

2. Zu politischen Rechten: Die Antwort "Keine Ahnung" deutet darauf hin, dass ein grundlegendes Verständnis von politischen Rechten fehlt.

3. Zu politischen Rechten auf Bundesebene: Die Antwort "Frei zu leben" ist falsch, da sie keine spezifischen politischen Rechte wie Wahlrecht oder Referendumsrecht erwähnt.

4. Zur Listenverbindung bei Proporzwahlen: Die Antwort "Bessere Ausstrahlungskraft" ist ungenau und unvollständig. Es fehlt eine Erklärung darüber, wie Listenverbindungen die Sitzverteilungen beeinflussen können.

5. Zum Grundsatz der Einheit der Materie: Diese Antwort ist korrekt und gut formuliert.

6. Zum Unterschied zwischen Wahlhandlungen: Die Antwort ist im Großen und Ganzen zutreffend, aber etwas verwirrend formuliert. Eine klarere Differenzierung wäre hilfreich.

Mögliche Folgefragen zur Vertiefung und Überprüfung des Verständnisses:

1. Warum dürfen im Kanton Glarus 16-Jährige an den Ständeratswahlen teilnehmen? Diese Frage wurde nicht korrekt beantwortet und kann erneut hervorgebracht werden, um das Verständnis zu vertiefen.

2. Was sind einige Beispiele für politische Rechte in der Schweiz und wie unterscheiden sie sich auf kantonaler und nationaler Ebene? Diese Frage zielt darauf ab, ein besseres Verständnis der politischen Rechte und deren Unterschiede auf verschiedenen Ebenen zu entwickeln.

3. Warum können Listenverbindungen bei Proporzwahlen strategisch vorteilhaft sein, besonders in Bezug auf die Vertretung im Parlament? Diese Frage soll das Verständnis dafür fördern, wie Listenverbindungen die Wahlstrategien beeinflussen können.

4. Was sind die Vorteile und eventuellen Nachteile von Listenverbindungen aus der Perspektive kleiner Parteien? Diese Frage hilft, die Perspektive kleiner Parteien zu verstehen und deren Rolle im politischen System der Schweiz zu beleuchten.

Diese Folgefragen zielen darauf ab, das Verständnis des Schülers über die Funktionsweise politischer Rechte und Wahlen in der Schweiz zu vertiefen und etwaige Lücken im Verständnis zu schließen.

GPT 4

Der Student zeigt ein mäßiges Verständnis einiger politischer Konzepte, hat jedoch offensichtliche Lücken in grundlegenden Verständnissen und spezifischen Fakten im Zusammenhang mit dem politischen System der Schweiz.

Der Student konnte die Frage, warum 16-Jährige im Kanton Glarus an den Ständeratswahlen teilnehmen dürfen, nicht beantworten. Diese Information könnte spezifische rechtliche oder historische Gründe haben, die besprochen werden sollten.

Bei der Frage nach den politischen Rechten hat der Student zugegeben, keine Kenntnisse zu haben. Dies ist ein fundamentales Konzept, das grundlegend verstanden werden muss.

Die Antwort auf die Frage nach den politischen Rechten auf Bundesebene war ungenau und zeigte ein Missverständnis in den Konzepten politischer Rechte. Politische Rechte umfassen meist Themen wie Wählen oder politische Teilnahme, nicht "frei zu leben", was eher zu allgemeinen Menschenrechten zählt.

Die Begründung zur Listenverbindung war unzureichend. Der tatsächliche Nutzen solcher Verbindungen bei Proporzwahlen geht über bloße "Ausstrahlungskraft" hinaus und betrifft technische und strategische Wahlvorteile.

Der Grundsatz der Einheit der Materie wurde korrekt erklärt, was zeigt, dass der Student dieses Konzept verstanden hat.

Der Unterschied zwischen den beschriebenen Wahlverhalten-Szenarien wurde zutreffend erläutert und zeigt ein gutes Verständnis der Auswirkungen von verschiedenen Stimmgebungen.

Folgefragen könnten sein:

Welche gesetzlichen Bestimmungen oder historischen Gründe erlauben es 16-Jährigen im Kanton Glarus, an den Ständeratswahlen teilzunehmen?

Was genau sind politische Rechte und welche Beispiele dafür gibt es in der Schweizer Verfassung?

Können Sie erläutern, wie eine Listenverbindung bei Proporzwahlen technisch funktioniert und warum diese insbesondere für kleinere Parteien von Vorteil sein könnte?

Weitere Fragen, die den Rest des Textes betreffen und noch nicht behandelt wurden:

Was sind die wesentlichen Unterschiede zwischen einem Proporz- und einem Majorz-Wahlsystem?

In welchen Kantonen der Schweiz gibt es Besonderheiten in Bezug auf das Wahlalter und warum?

Welche besonderen Rechte haben Ausländer in Bezug auf die politische Teilnahme in der Schweiz?

Erläutern Sie den Begriff und die Bedeutung der politischen Initiativen und Referenden in der Schweizer Demokratie.

Mixtral
Verständnisniveau:

Der Schüler hat ein grundlegendes Verständnis einiger Konzepte des schweizerischen Wahlsystems, wie zum Beispiel den Unterschied zwischen aktiven und passiven Wahlrechten und den Begriff obligatorischer und fakultativer Referenden. Es gibt jedoch mehrere Bereiche, in denen das Verständnis des Schülers unzureichend oder falsch ist.

Begründung: Der Schüler konnte nicht beantworten, warum 16-Jährige im Kanton Glarus an Bundeswahlen teilnehmen dürfen, was auf ein Mangel an Wissen in diesem Bereich hinweist. Der Schüler konnte politische Rechte nicht definieren, was ein grundlegender Begriff in diesem Text ist. Die Antwort des Schülers auf die Frage nach politischen Rechten auf Bundesebene in der Schweiz war falsch. Das Recht, frei zu leben, ist kein politisches Recht. Stattdessen gehören zu den politischen Rechten das Wahlrecht, das Recht, für ein Amt zu kandidieren, und das Recht, an Volksinitiativen und Referenden teilzunehmen. Die Antwort des Schülers darauf, warum eine grössere Partei eine Listenverbindung mit kleineren Parteien in proportionalen Wahlen eingehen möchte, war teilweise korrekt. Zwar kann erhöhte Sichtbarkeit ein Vorteil sein, doch ist der Hauptgrund für eine solche Listenverbindung die Erhöhung der Chancen, die Mehrheit der Sitze zu erhalten. Die Antwort des Schülers auf die Frage zum Prinzip der Einheit der Materie war korrekt. Die Antwort des Schülers auf die Frage nach dem Unterschied zwischen dem Streichen eines Kandidaten von einer Liste und dem Ersetzen durch einen anderen Kandidaten derselben Liste war korrekt.

Follow up Fragen:

Können Sie näher erläutern, was das aktive Wahlrecht beinhaltet und was das passive Wahlrecht ist?
Können Sie einige Beispiele für politische Rechte auf Bundesebene in der Schweiz nennen, abgesehen vom Wahlrecht und dem Recht, für ein Amt zu kandidieren?
Können Sie den Unterschied zwischen obligatorischen und facultativen Referenden näher erläutern?
Können Sie einige konkrete Beispiele dafür nennen, wann in der Schweiz ein obligatorisches Referendum abgehalten wird?

Neue Frage auf der Grundlage der Datei: Was ist der Unterschied zwischen den Wahlrechten von Schweizer Bürgern und Nicht-Schweizer Bürgern in der Schweiz?

GPT 3.5 finetuned

Die Antworten sind noch nicht korrekt. Der Studierende hat sich weiterhin ungenügend mit den behandelten Themen auseinandergesetzt. Der Studierende konnte bisher keine der Fragen korrekt beantworten. Es besteht Unklarheit über grundlegende Konzepte wie politische Rechte, und die gegebenen Antworten zeigen ein geringes Verständnis der behandelter Themen, z.B. den Unterschied zwischen Listenverbindung und Kumulation sowie den Grundsatz der Einheit der Materie. Der Studierende sollte die grundlegenden Konzepte der politischen Rechte und der Wahlverfahren überprüfen sowie das Prinzip der Einheit der Materie erneut betrachten, um ein besseres Verständnis zu entwickeln.

Mixtral finetuned

Es scheinen noch Wissenslücken zu bestehen bezüglich der Definition politischer Rechte und den verschiedenen Ebenen der politischen Rechte in der Schweiz. Zudem sind die bisherigen Antworten teilweise unpräzise. Weitere Aspekte sind noch nicht behandelt. Folluw up: Nennen Sie die verschiedenen Ebenen politischer Rechte in der Schweiz.

References

[1] Patricia Armstrong. Bloom's taxonomy. https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/, 2010. (accessed: 09.06.2024).

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[3] Dominik Castelberg and Linus Flury. Llm assisted development. https://eprints.ost.ch/id/eprint/1187, 2023. (accessed: 09.06.2024).

[4] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023.

[5] Eduard Frankford, Clemens Sauerwein, Patrick Bassner, Stephan Krusche, and Ruth Breu. Ai-tutoring in software engineering education. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training*, ICSE-SEET '24, page 309–319, New York, NY, USA, 2024. Association for Computing Machinery.

[6] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[7] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

[8] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. *CoRR*, abs/2005.11401, 2020.

[9] Chien-Chang Lin, Anna Y. Q. Huang, and Owen H. T. Lu. Artificial intelligence in intelligent tutoring systems toward sustainable education: a systematic review. *Smart Learning Environments*, 10(1):41, 2023.

[10] OpenAI. Six strategies for getting better results with prompt engineering. https://platform.openai.com/docs/guides/prompt-engineering/six-strategies-for-getting-better-results, 2024. (accessed: 12 June 2024).