Enhancing Student Support Services with Large Language Models: Developing a Chatbot and Virtual Assistance System for the University of Padova Named Galileo

Mohammad Khosravi
mohammad.khosravi.1@studenti.unipd.it

SINA DALVAND
sina.dalvand@studenti.unipd.it

LORENZO ROSALES VASQUEZ
lorenzo.rosalesvasquez@studenti.unipd.it

NAZANIN GHORBANI
nazanin.ghorbani@studenti.unipd.it

MOHAMMAD HOSSEINIPOUR
mohammad.hosseinipour@studenti.unipd.it

Abstract

In this paper, we explore the development of a chatbot for the University of Padova called Galileo, aimed at enhancing the student experience by providing accurate information on courses. To achieve this, we evaluated five state-of-the-art pre-trained Large Language Models (LLMs): Llama3[1], Gemma[2], Qwen[3], Mixtral[4], and Mistral[5]. Our methodology involved a comprehensive comparison based on metrics such as RAG score, hallucination, and toxicity, as well as human evaluation, multilingual capabilities, and usage rights. After selecting the most suitable model, we conducted experiments using two methods— Fine-tuning and Retrieval Augmented Generation (RAG) pipeline —to feed the selected language model with university data. We then evaluated the viability of each method based on our specific use case and requirements. Then we chose the most appropriate method, fed the LLM and tested it thoroughly. The results and insights gained from these experiments are summarized in the conclusion of this report. Future work will focus on scaling the model, refining the RAG architecture, and addressing practical deployment challenges.

1 Introduction

The rapid advancement of AI and natural language processing (NLP) technologies offers new opportunities to enhance education. In this project we aim to develop an intelligent chatbot for the University of Padova to improve the student experience by providing accurate and timely information. The project's first goal is to identify and implement the most suitable large language model (LLM) from five evaluated options: Llama3, Gemma, Qwen, Mixtral, and Mistral. A key challenge is maintaining the accuracy and relevance of information due to the dynamic nature of university resources. To address this, the project employs a Retrieval-Augmented Generation (RAG) pipeline, a novel approach that ensures continuous updates to the LLM's knowledge base.

2 Related Work

Several universities and educational platforms have integrated chatbots to enhance student engagement and streamline services. For example, Stanford University uses a chatbot[6] for course selection and scheduling, while the Georgia Institute of Technology employs an AI teaching assistant named Jill Watson[7] to support students in online courses. These implementations have demonstrated the potential of chatbots to significantly reduce administrative workload and improve information dissemination efficiency.

At the University of Padova, an investigation into existing chatbot services revealed several initiatives aimed at improving student services, though with varying degrees of success:

Department of Economics Management Chatbot[8]: This chatbot operates with predefined questions but struggles with complex or relevant student queries, often lacking depth and specificity in its responses.

Career Services Chatbot[9]: Available only in Italian, this chatbot has issues with accuracy and comprehension, reducing its utility for non-Italian speaking students or those seeking precise career information.

The University of Padova's current chatbot solutions suffer from low accuracy, poor user comprehension, and limited functionality. This project aims to address these issues by implementing a state-of-the-art LLM-based chatbot using advanced NLP models and the Retrieval-Augmented Generation (RAG) pipeline to enhance the student experience at the University of Padova.

3 Methodology

Our methodology focuses on identifying and implementing the most suitable large language model (LLM) for developing a chatbot that addresses the specific needs of the University of Padova. To select the best model, we evaluated five state-of-the-art pre-trained LLMs:

- Llama3 8b, 70b
- Gemma 2b, 7b

- Qwen/Qwen1.5-0.5B-Chat
- Mixtral 8×22b, 8×7b
- Mistral 7b

These models were chosen based on their high performance in the Hugging Face open LLMs leader board [10] and their potential to meet the requirements of our application. In the "Models Comparison Factors" section, we introduce the criteria used to evaluate the LLMs under consideration. Subsequently, in the "Model Selection" section, we discuss our final decision on the most suitable LLM for training and fine-tuning, based on the comprehensive evaluation of these factors.

3.1 Models Comparison Factors

The following factors have been employed as our measurement criteria to select the final LLM to proceed with:

3.1.1 Metrics

We initially considered traditional automated metrics like ROUGE, METEOR, BERTScore, and GLEU for evaluating our models. While these metrics are widely used for language generation tasks, they have significant limitations in our context. They rely on word-by-word comparisons between the model's output and the expected answers, leading to misleadingly low scores when responses do not match exactly. This approach fails to account for the semantic correctness or relevance of the responses, which is crucial for our chatbot's effectiveness.

Given these limitations, we employed a novel approach better suited to our requirements, which is detailed in the following sections. Other evaluation methods exist, but we found them less effective and omitted them for brevity. Plus, according to our findings they are not as effective as the following methods:

- **LLM Evaluations(RAGAS Scores):** With the advancements in LLMs such as GPT-4 and GPT-4o, one of the most accurate ways to evaluate a model is to use another LLM (the leading and most accurate) as a measurement criterion to judge other new LLMs. This method, used by Yang Liu et al[11]. from Microsoft, led to the development of a framework called G-Eval, which uses LLMs to evaluate LLM outputs (LLM-Evals).

  However, even using LLM-Eval as a judge posed challenges, as the models did not share the same knowledge base to answer identical questions. To address this, we used Retrieval-Augmented Generation (RAG) to provide a consistent knowledge base for our LLMs.

  Retrieval-Augmented Generation (RAG) serves as a method to supplement LLMs with extra context to generate tailored outputs, making it ideal for building chatbots. It consists of two components: the retriever and the generator.

  The RAG system receives an input. The retriever uses this input to perform a vector search in our knowledge base (often a vector database). The generator receives the retrieval context and the user input to generate a tailored output. High-quality LLM outputs depend on a proficient retriever and generator. Consequently, effective RAG metrics focus on evaluating either the RAG retriever or generator reliably and accurately. RAG metrics are designed to be reference-less metrics, meaning they do not require ground truths, making them usable even in a production setting. We will cover RAG architecture in more depth later in the experiments section.

  We fed our data to our LLM (the University of Padova's "how to apply" page[12]) using Langchain Document Loader and designed a reasonable number of questions to assess the LLMs based on the same criteria and data. We then used GPT-4o with the help of DeepEval to judge whether the outputs of our LLMs matched the expected answers, considering the retrieval context provided.

  We utilized the RAGAS (Retrieval Augmented Generation Assessment) [13Shahul Es et al] score from the DeepEval framework, which is the average of four distinct metrics:

  – Answer Relevancy: Measures the quality of the RAG pipeline's generator by evaluating the relevance of the actual output compared to the input. The score is calculated as:

  $$AR = \frac{NumberofRelevantStatements}{TotalNumberofStatements} \quad (1)$$

  – Faithfulness: Measures whether the actual output factually aligns with the contents of the retrieval context. The score is calculated as:

  $$F = \frac{NumberofTruthfulClaims}{TotalNumberofClaims} \quad (2)$$

  – Contextual Precision: Measures whether relevant nodes in the retrieval context are ranked higher than irrelevant ones. The score is calculated based on the ranking accuracy of relevant context nodes.

  $$CP = \frac{\sum_{k=1}^{n} \left(\frac{NumberofRelevantNodesUptoPositionk}{k} \times r_k \right)}{NumberofRelevantNodes} \quad (3)$$

  * **k**: This is the index of the node in the retrieval context. It runs from 1 to n.
  * **n**: This is the total number of nodes in the retrieval context.

* $\mathbf{r}_k$: This is a binary relevance score for the k-th node. It is 1 if the node is relevant to the input and 0 if it is not.

– Contextual Recall: Measures the extent to which the retrieval context aligns with the expected output. The score is calculated as:

$$CR = \frac{Number of Attributable Statements}{Total Number of Statements} \tag{4}$$

Figures 1 and 2 shows the results of our LLMs based on these metrics.

Model	Contextual Precision	Contextual Recall	Faithfulness	Answer Relevancy	RAGAS Overall Score
Llama3 8b	0.9999999999	0.8333333333	0.6810101010	0.6783961932	0.7981849069
Gemma 2B	0.9999999999	0.8333333333	0.6433333333	0.6807886884	0.7893638388
Llama3 70b	0.9999999999	0.8333333333	0.6116666667	0.5973696629	0.7605924157
Qwen/Qwen1.5-0.5B-Chat	0.9999999999	0.8333333333	0.1902941176	0.6758084369	0.6748589720
Mixtral-8x7B-Instruct-v0.1	0.9999999999	0.8333333333	0.0599060150	0.5217392026	0.6037446377
Mixtral-8x7B-Instruct-v0.1	NA	NA	NA	NA	NA
Gemma 7B	NA	NA	NA	NA	NA
Mistral 7b	NA	NA	NA	NA	NA

Figure 1: Models Comparison Table

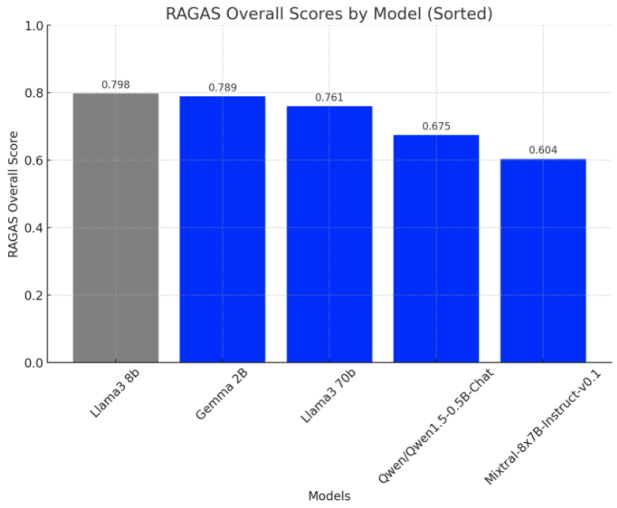

Figure 2: Models RAGAS Score Comparison chart

In our evaluation of various models using the RAGAS metric, we found that Llama3 8b achieved the highest overall RAGAS score of 0.798, indicating superior performance in contextual precision, recall, faithfulness, and answer relevancy. Gemma 2B and Llama3 70b followed with scores of 0.789 and 0.761, respectively, showing strong but slightly lower performance. Qwen/Qwen1.5-0.5B-Chat and Mixtral-8x7B-Instruct-v0.1 scored lower at 0.675 and 0.604, respectively, indicating room for improvement, especially in faithfulness.

Contextual precision and recall were consistently high across all models, indicating a strong understanding and retention of context. Faithfulness varied significantly, with Llama3 8b (0.681) and Gemma 2B (0.643) performing better, showing higher accuracy in factual information. Answer relevancy scores were more dispersed, with Gemma 2B (0.681) and Llama3 8b (0.678) leading, highlighting differences in response relevance.

Despite significant computational resources, inadequate GPU memory (peaking at 22 GB on the L4 GPU) limited the comprehensive evaluation of some models. Mixtral-8x7B-Instruct-v0.1, Gemma 7B, and Mistral 7b could not be fully evaluated due to these hardware constraints and are marked with ”NA” in the table.

- **Human Evaluation** The evaluation process involved human judges assessing the accuracy and relevance of the LLM-generated answers. Five evaluators compared the given answers of the models with the expected answers, assigning scores based on correctness and relevance. The average score for each model was then calculated to provide a comprehensive evaluation of performance from a human perspective. Figure 3 illustrates the results of our human evaluation on the models which clearly proves the superiority of Llama3 models:

Model	Average Score
Llama3 70b	0.785
Llama3 8b	0.7825
Qwen/Qwen1.5-0.5B-Chat	0.39325
Mixtral 8×7b	0.3525
Gemma 2b	0.345

Figure 3: Human Evaluation of the Models

- **Hallucination:** Hallucination refers to the generation of responses by the model that contain information not present in the input or the training data. In other words, it occurs when the model generates incorrect or fabricated information. This metric is essential for evaluating the model's ability to generate accurate and reliable responses. Hallucination can be quantified by assessing the extent to which the model generates responses that deviate from factual information or contextually appropriate content. Figure 4 compares the hallucination rates of our models as reported on the Hugging Face leader board[14] and clearly shows that Mixtral, Mistral and Llama3 models have the lowest Hallucination rates.

Model	Hallucination Rate	Factual Consistency Rate	Answer Rate	Average Summary Length (Words)
Mixtral 8x22B	3.80%	96.20%	99.90%	92
Mistral 7B Instruct-v0.2	4.50%	95.50%	100.00%	106.1
Llama 3 70B	4.50%	95.50%	99.20%	68.5
Llama 3 8B	5.40%	94.60%	99.80%	79.7
Google Gemma-1.1-7b-it	6.30%	93.70%	100.00%	64.3
Mixtral 8x7B	9.30%	90.70%	99.90%	90.7
Mistral 7B Instruct-v0.1	9.40%	90.60%	98.70%	96.1
Google Gemma-1.1-2b-it	11.20%	88.80%	100.00%	66.8
Qwen 1.5-0.5b chat	N/A	N/A	N/A	N/A

Figure 4: Hallucination Rate Comparison

- **Toxicity:** Toxicity refers to harmful, offensive, or inappropriate content generated by the model, encompassing abusive, derogatory, or disrespectful language towards individuals or groups. Assessing toxicity involves identifying and quantifying such content to ensure respectful and safe responses. The toxicity rate is evaluated using the TrustLLM Benchmark[15] website, with scores ranging from -1 to 1. Scores close to -1 indicate highly non-toxic responses, suggesting positive, friendly, and polite language. Scores around 0 indicate neutral responses, meaning the content is neither harmful nor particularly positive. Scores close to 1 indicate highly toxic responses, suggesting harmful, offensive, or inappropriate language. This scoring helps ensure that the model's outputs are suitable for all users.

Model	Mixtral	Llama 3 8B	Llama 3 70B	Mistral 7B	GPT-4	Gemma	Qwen
Toxicity	-1.0	0.059	0.189	0.262	0.386	N/A	N/A

Figure 5: Toxicity Rate Comparison

Figure 5 demonstrates the toxicity rate of our models[16Lichao Sun et al]. According to it Mixtral and LLama3 models have the lowest toxicity rates.

3.1.2 Benchmarks

This section provides an overview of several prominent LLM benchmarks, essential for evaluating various LLM models' performance across different tasks. We used the following benchmarks to assess our models, focusing on reasoning and correct answer ability:

MMLU (Massive Multitask Language Understanding): Assesses LLM performance across 57 tasks in multiple domains, using 14,000 questions from academic and professional tests. The metric is accuracy in answering multiple-choice questions correctly [17Hendrycks et al., 2020].

HellaSwag: Focuses on commonsense reasoning by selecting plausible continuations of given contexts, derived from ActivityNet and WikiHow. The dataset includes 70,000 examples, and accuracy is the primary metric [18Zellers et al., 2019].

BIG-Bench Hard: A subset of the BIG-Bench project, targeting tasks challenging for LLMs but easy for humans, covering domains like mathematics and logic. Performance is evaluated using task-specific metrics [19BIG-Bench Collaboration, 2022].

DROP (Discrete Reasoning Over Paragraphs): Evaluates reading comprehension with discrete reasoning tasks over text from Wikipedia, including arithmetic, counting, and sorting. Metrics include exact match accuracy and F1 score, with a dataset of 96,000 questions [20Dua et al., 2019].

TruthfulQA: Tests the truthfulness of LLM-generated answers to questions that might prompt false information, focusing on factually accurate responses. The dataset has 817 questions, and the metric is the percentage of correct answers [21Lin et al., 2021].

GSM8K (Grade School Math 8K): Evaluates mathematical reasoning through grade school-level math problems. Performance is measured by the accuracy of answers, with a dataset of 8,000 problems [22Cobbe et al., 2021].

Figure 6 outlines the evaluation of our LLMs and also GPT-4 as a comparison on these benchmarks.

Model	MMLU	HellaSwag	BIG-Bench Hard	DROP	TruthfulQA	GSM8K	Overall
GPT-4	85.0	90.2	65.4	88.1	60.0	85.5	79.03
Qwen 1.5-0.5b chat	68.9	81.0	57.8	78.5	47.3	72.4	67.65
Llama3 70b	66.8	79.1	55.4	75.2	44.6	70.3	65.23
Mixtral 8×22b	64.1	77.3	53.1	73.1	42.0	67.5	62.85
Gemma 7b	64.3	81.2	51.8	72.3	52.8	53.2	62.60
Mistral 7b	62.5	81.0	47.0	66.3	52.2	54.9	60.65
Mixtral 8×7b	60.5	74.2	50.7	70.5	38.7	64.8	59.90
Llama3 8b	53.2	69.8	46.8	66.0	33.0	58.1	54.48
Gemma 2b	42.3	71.4	49.7	65.4	47.8	42.1	53.11

Figure 6: Models Benchmarks Comparison[23]

3.1.3 Multilingual Capability

The analysis of multilingual large language models (LLMs) compares Gemma, Llama, Mistral, and Mixtral models across different languages. Mistral models, such as Mistral-8B-Instruct-v0.1 and Mistral-7B-Instruct-v0.2, demonstrate superior multilingual capabilities with efficient language processing characterized by lower token lengths. Mixtral shows balanced performance similar to Mistral models, indicating strong multilingual optimization. In contrast, Gemma-7b-it, while reasonably balanced, does not achieve the same level of optimization as Mistral and Mixtral. Overall, Mistral and Mixtral models lead in multilingual efficiency, while Llama requires significant improvements in this area. The mentioned models supports Italian language as it is one of our requirement for the future work.

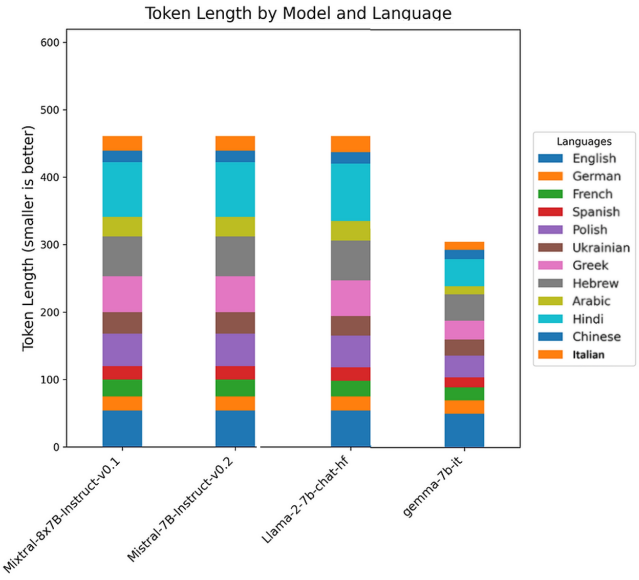

Figure 7: Models Multilingual Capabilities Comparison[24] [25]

3.1.4 usage rights

usage rights distinguish between commercial and research uses of models. Commercial use allows entities to deploy models in profit-generating activities, while research use limits applications to non-commercial, academic, or experimental contexts. Figure 8 compares the usage rights of the open-source LLMs we worked on:

Model Name	Commercial Use	Research Use	References
Gemma 2b	Yes	Yes	Gemma - Google Blog
Gemma 7b	Yes	Yes	Gemma - Google Blog
Mistral 7b	Yes	Yes	Mistral Models
Llama3 8b	No	Yes	Llama - Meta AI
Llama3 70b	Yes	Yes	Llama - Meta AI
Mixtral 8×22b	Yes	Yes	Mixtral - Mistral
Mixtral 8×7b	Yes	Yes	Mixtral - Mistral
Qwen 72b	Yes	Yes	Qwen - Hugging Face

Figure 8: Models Multilingual Capabilities Comparison[26] [27][28] [29] [30]

3.2 Model Selection

The Metrics section adopts an innovative approach using Retrieval-Augmented Generation Assessment (RAGAS) scores to evaluate LLMs. This method leverages RAG for generating accurate outputs and assesses Answer Relevancy, Faithfulness, Contextual Precision, and Contextual Recall. Human assessments complement automated evaluations, focusing on accuracy and relevance while considering factors like hallucination,and toxicity.

Llama3 8B emerges as the preferred choice due to its strong overall RAGAS score of 0.798, excelling in contextual precision, recall, faithfulness, and answer relevancy. It also ranks high in human evaluation despite not leading in benchmarks, making it suitable for a university chatbot application. Gemma 2B, despite a score of 0.789, suffers from low reasoning capacity and high hallucination rates, leading to its exclusion.

Models like Llama3 70B and Mixtrall 8*22B show promise but are hindered by high computational requirements, prompting the selection of Llama3 8B for its balanced performance and manageable resource demands and considering its usage rights and acceptable multilingual capabilities.

4 Experiments

In this section, we describe the technologies and methodologies employed to integrate new data into our selected model, Llama 3 8b. Our experiment involved feeding the model with data not included in its pre-existing knowledge base. We used the course catalogs of the master's degree programs from the School of Science at the University of Padova[31], a 778-page document.

4.1 Feeding Llama3 8B (Galileo)

There are two primary methods for incorporating new data into an LLM: fine-tuning and Retrieval-Augmented Generation (RAG). The choice of method depends on the specific use case.

4.1.1 Fine-tuning

Fine-tuning involves retraining the LLM on new data, a widely-used technique since the inception of LLMs. It is particularly useful for relatively small datasets that are not frequently updated. Fine-tuning requires substantial computational resources and time. Although it can achieve high performance, it is computationally expensive and slow, which may not be ideal for all use cases.

To experiment with fine-tuning, we prepared a small dataset from the course catalogs of the master's degree in computer science[32]. The data was formatted to train the Llama 3 8b model effectively. Dataset typically involves three categories: instruction, input, and output. The instruction is the directive given to the LLM , the input is the user's query (in the context of a chatbot), and the output is the raw data providing the answer.

However, we encountered several challenges during fine-tuning. The model struggled with questions not explicitly covered in the training data, particularly when questions were paraphrased. Iterative training, rather than a single extended session, was recommended to improve performance. Despite using an L4 GPU, the process remained time-consuming and resource-intensive. Given the continuous updates to university data and our limited resources, fine-tuning proved unsatisfactory for our use case. The dataset prepared and fine-tuned models is uploaded at the Hugging-face website[33] for further inspections.

4.1.2 Retrieval-Augmented Generation (RAG)

Our subsequent experimentation focused on the RAG approach, which offered greater flexibility and required fewer computational resources. Previously, in the Methodology section, we used RAG to provide a consistent context for different language models, allowing us to evaluate them on the same data and select the final models. In this section, we employ RAG to enhance our language model with our own data. RAG optimizes the LLM's output by referencing an external, authoritative knowledge base before generating responses. This method is particularly effective for large, frequently updated datasets which is exactly our use case for given the nature of our university data which is constantly being updated.

For this experiment,to give a general overview, we utilized "LangChain[34]", an open-source developer framework for building LLM applications. We prepared the data, used an embedding model to persist the embedded data in a vector database, and implemented a retrieval system. When a query is made by a user (e.g., a student), the system performs a similarity search based on "Euclidean Distance". The top k relevant retrieved data points are then provided to the LLM to generate the response. In figure 9 the architecture used to implement RAG is depicted

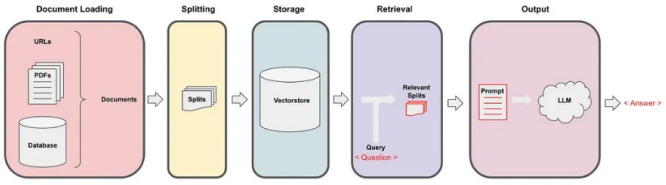

Figure 9: RAG architecture used for feeding Galileo

This approach proved more efficient and aligned better with our use case, offering a practical solution given our resource constraints. By leveraging RAG, we could maintain up-to-date information and provide accurate responses without the extensive computational demands of fine-tuning.

4.1.3 RAG Architecture Implementation

- **Data Preparation:** We initiated our process by scraping the course catalog for all master's programs within the School of Science at the University of Padova from the university's website and structured them by preparing two CSV files: one containing the degree programs and the other detailing all the courses within each program, including information such as courses lecturer, start of activity, course materials, etc. After cleaning both files and removing redundant information, we joined them. Additionally, we prepared a metadata document providing detailed explanations for each column, clarifying their meaning for the LLM. Another document containing instructions (prompts) for the LLM on how to handle the data was also created. Ultimately, we merged all four documents, preparing them for further processing.
- **Document Splitting:** In the next phase, we utilized LangChain's "RecursiveSplitting" text splitter to divide the merged documents, enabling them to fit within the content window of the LLM. Splitting documents into smaller chunks is crucial yet challenging, as maintaining meaningful relationships between chunks is essential. The input text was split based on a defined chunk size with a specified chunk overlap. The chunk size is a measure of the chunk's length, often in characters or tokens. Chunk overlap allows for some consistency between two chunks. We employed a chunk size of 1500 with an overlap size of 150.
- **Vector Store and Embeddings:** After splitting the documents into small chunks, we indexed these chunks for easy retrieval when answering questions about the document. We used embeddings and vector stores for this purpose. Embeddings convert a piece of text into a numerical representation, ensuring that semantically similar content has similar vectors in embedding space. By comparing the embeddings, we can find text that is similar. The pipeline starts with documents, which are split into smaller segments, and embeddings of these segments are created. These embeddings are then stored in a vector store. A vector store is a database that allows for easy lookup of similar vectors later on, which is useful for finding relevant documents in response to a query. For this project, we used the All MiniLM embedding model[35]to create these embeddings and Chroma[36] as our vector database. Chroma is lightweight and in-memory, making it easy to start with. Additionally, we implemented caching to reduce the overall process time. When embedding a new document, the method first checks the cache for existing embeddings. If not found, it uses the underlying embedder to embed the documents and stores the results in the cache. This approach reduces computational overhead and costs by avoiding recalculating embeddings.
- **Retrieval:** Retrieval is the core of our Retrieval-Augmented Generation (RAG) flow. It is one of the biggest challenges in question-answering over documents. Retrieval errors are often the cause of failure in question answering. Retrieval is crucial at query time to retrieve the most relevant document splits.
- **Question Answering:** For question answering, we retrieved documents and the original question, passing both to a language model to generate the answer. We employed Chain-of-Thought prompting to

enhance the LLM's performance on tasks requiring logic, calculation, and decision-making by structuring the input prompt to mimic human reasoning. This allowed the LLM to use its reasoning capabilities to provide answers that could be inferred from the given information. Additionally, we incorporated chat history, enabling Galileo to remember previous questions and answers, providing a stateful interaction experience. This prevents the LLM from answering questions in a stateless format, thereby improving response coherence and relevance. Figure 10 demonstrates the whole procedure again as a wrap up.

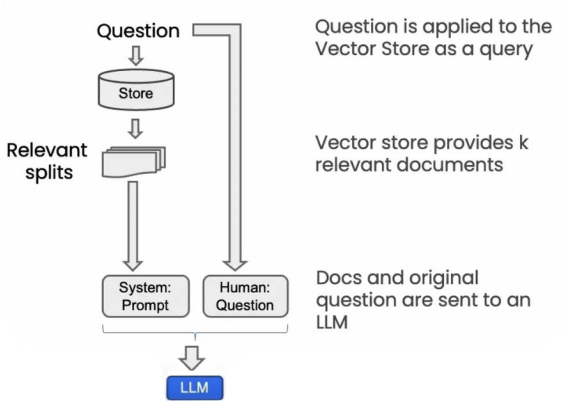

Figure 10

4.2 Final Evaluation and testing

In this section, we present the evaluation of the Llama 3 8b model, conducted following the methodology outlined in the previous section.

4.2.1 LLM Evaluation (RAGAS Scores)

For the automated evaluation, we again adopted the previous approach explained before in the Methodology section. Initially, we designed a set of questions derived from the comprehensive dataset of course catalogs encompassing all master's degree programs offered by the School of Science. Subsequently, we employed GPT-40 as a benchmark to assess the accuracy of Llama 3 8b's responses based on the provided information.

The evaluation metrics yielded the following results for Galileo:

- Contextual Precision: Average score of 0.9999
- Contextual Recall: Average score of 0.85
- Faithfulness: Average score of 0.4033
- Answer Relevancy: Average score of 0.5243
- **RAGAS: Average overall score of 0.6944**

The evaluation of the Galileo RAG pipeline shows exceptional contextual precision (0.9999) and solid contextual recall (0.85), but it struggles with faithfulness (0.403) and answer relevancy (0.524). This indicates that while Galileo maintains context well, it often produces inaccurate or irrelevant information. Two main factors contribute to this: the chunk size of retrieved documents limits the ability to fully explain questions, and the context from the course catalog was cleaned of verbosity, reducing relevancy and faithfulness. Despite these issues, Galileo's performance remains satisfactory and above the 60 percent threshold we defined for passing the test, with human evaluation we can ensure confidence in the results.

4.2.2 Human Evaluation

Despite recognizing the limitations of any LLM, including GPT-4O, we conducted a human evaluation to thoroughly assess Galileo's performance. Five developers evaluated Galileo's answers to questions from the previous phase. The human evaluation yielded a satisfactory **score over 90 percent**, with Galileo answering all questions correctly. Some answers were brief, which was expected as the chatbot was designed to provide concise responses unless detailed explanations were requested.

Based on both automated and manual testing, we conclude that the Llama 3 8b model, referred to as Galileo, performed satisfactorily in generating responses from the university course catalog dataset. While there is room for improvement in faithfulness and answer relevancy, Galileo meets the project's requirements within the given timeframe.

5 Conclusion

In this report, we assessed various pretrained LLMs based on metrics, benchmarks, and other factors, ultimately selecting Llama3 8B as the final model due to its strong performance, relatively low size, and modest computational requirements. We explored two strategies for utilizing the Llama3 model: fine-tuning and Retrieval-Augmented Generation (RAG). The RAG strategy was preferred for its flexibility in constant data feeding and lower computational resource needs.

After implementing the RAG pipeline, we evaluated our model, now referred to as Galileo, through both automated and manual testing, with satisfactory results. Due to time constraints of this class project, future work will focus on expanding Galileo's data to include the entire university dataset.

6 Future Work

For the future work, several advanced techniques could be implemented to enhance the robustness of the chatbot, such as Maximum Marginal Relevance (MMR) [Carbonell et al.37], Contextual Compression, Self Query [38], and Recursive Abstractive Processing for Tree Organized

Retrieval (RAPTOR)[39]. Implementing these techniques will ensure Galileo effectively enhances student support services across the University of Padova.

By following these steps and incorporating advanced techniques, we can ensure Galileo fully meets the needs of the university community.

7 References

References

[1] Llama 3 LLM official webpage https://llama.meta.com/llama3/

[2] Gemma LLM official webpage https://ai.google.dev/gemma

[3] Qwen LLM official webpage https://qwenlm.github.io/blog/qwen1.5/

[4] Mixtral 8x22b official webpage https://mistral.ai/news/mixtral-8x22b/

[5] Mistral official webpage https://mistral.ai/news/announcing-mistral-7b/

[6] Standford University Chatbot https://financialaid.stanford.edu/cardy/

[7] Georgia Institute of Technology AI teaching assistant named Jill Watson https://emprize.gatech.edu/

[8] Department of Economics Management of the University of Padova Chatbot https://www.economia.unipd.it/en/

[9] Career Service,University of Padova https://www.unipd.it/career-service

[10] Hugging face Open LLM Leaderboard https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard

[11] Yang Liu,Dan Iter,Yichong Xu,Shuohang Wang,Ruochen Xu,Chenguang Zhu,"G-EVAL: NLG Evaluation using GPT-4 with Better Human Alignment",Microsoft Cognitive Services Research, May2023 https://arxiv.org/pdf/2303.16634

[12] UNIPD how to apply webpage https://www.unipd.it/en/how-apply

[13] Shahul Es , Jithin James , Luis Espinosa-Anke, Steven Schockaert,"RAGAS: Automated Evaluation of Retrieval Augmented Generation",Exploding Gradients ,CardiffNLP, Cardiff University, United Kingdom ,AMPLYFI, United Kingdom, September 2023 https://arxiv.org/pdf/2309.15217

[14] Huggingface Hallucination Leaderboard https://huggingface.co/spaces/vectara/Hallucination-evaluation-leaderboard

[15] TRUSTLLM Leaderboard https://trustllmbenchmark.github.io/TrustLLM-Website/leaderboard.html

[16] Lichao Sun, Yue Huang,Haoran Wang et al,"TRUSTLLM: TRUSTWORTHINESS IN LARGE LANGUAGE MODELS", March 2024, https://arxiv.org/pdf/2401.05561

[17] Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., Steinhardt, J. (2020). Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1903.12261

[18] Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., Choi, Y. (2019). Hellaswag: Can a machine really finish your sentence? *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*. https://arxiv.org/abs/1905.07830

[19] BIG-Bench Collaboration (2022). "Beyond the Imitation Game Benchmark (BIG-Bench)."Dua, D., et al. (2019). https://arxiv.org/pdf/2206.04615

[20] "DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs.",Lin, B., et al. (2021). https://arxiv.org/pdf/1903.00161

[21] "TruthfulQA: Measuring How Models Mimic Human Falsehoods."Chen, M., et al. (2021). "Evaluating Large Language Models Trained on Code." https://arxiv.org/pdf/2107.03374

[22] Cobbe, K., et al. (2021). "Training Verifiers to Solve Math Word Problems." https://arxiv.org/pdf/2110.14168

[23] he data is gathered from the following website https://paperswithcode.com

[24] Wang, Y., Bao, Y., Chen, B., Peng, W., Sun, L., Ma, W., Wu, J., Zhou, W., Chen, W., Zhang, Z. (2024). GPT-4 Technical Report. arXiv. https://arxiv.org/abs/2403.03814

[25] Zhang, W., Aljunied, M., Gao, C., Chia, Y. K., Bing, L. (2023). M3Exam: A Multilingual, Multimodal, Multilevel Benchmark for Examining Large Language Models. In Advances in Neural Information Processing Systems 36 (NeurIPS 2023) Datasets and Benchmarks Track. Retrieved from , https://proceedings.neurips.cc/paper_files/paper/2023/hash/117c5c8622b0d539f74f6d1fb082a2e9-Abstract-Datasets_and_Benchmarks.html

[26] Gemma LLM Usage rights https://blog.google/technology/developers/gemma-open-models/

[27] Llama LLM Usage rights https://ai.meta.com/static-resource/responsible-use-guide/

[28] Mistral LLM Usage rights https://mistral.ai/technology/#models

[29] Mixtral LLM Usage rights https://mistral.ai/technology/#models

[30] Qwen LLM Usage rights https://huggingface.co/Qwen/Qwen-72B

[31] Data used to feed Llama3 8B on https://en.didattica.unipd.it/off/2024/LM/SC

[32] Data used to fine-tune Llama3 8B on https://en.didattica.unipd.it/off/2024/LM/SC/SC2598

[33] Dataset and models fine-tuned available at Hugging-face website https://huggingface.co/MohammadKhosravi

[34] Langchain opensource framwork https://python.langchain.com/v0.1/docs/get_started/introduction

[35] all-MiniLM-L6-v2 https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[36] Chroma Vector DB https://api.python.langchain.com/en/latest/vectorstores/langchain_community.vectorstores.chroma.Chroma.html

[37] Carbonell, J., Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 335-336. https://www.cs.cmu.edu/~jgc/publication/The_Use_MMR_Diversity_Based_LTMIR_1998.pdf

[38] LangChain's Self Query https://python.langchain.com/v0.1/docs/modules/data_connection/retrievers/self_query/

[39] Pang, R. Y., Parrish, A., Joshi, N., Nangia, N., Phang, J., Chen, A., Padmakumar, V., Ma, J., Thompson, J., He, H., Bowman, S. (2024). RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval. arXiv preprint arXiv:2401.18059. https://arxiv.org/pdf/2401.18059

8 Work Plan

8.1 Searching for Available and Similar Unipd Services

The Task Is Completed

The results are explained in the section "Related Works." Members involved and time spent: All members **Each member spent 2 to 3 hours** searching different sections of the university.

8.2 Research

The Task Is Completed

Members involved: **Sina Dalvand**: Researched the capabilities of state-of-the-art LLMs such as Llama2 and Llama3. **3 hours**

Lorenzo Rosales Vasquez: Assessed pre-made chatbots like DialogFlow. **4 hours**.

Mohammad Khosravi: Investigated metrics for LLM evaluation, focusing on semantic evaluation metrics such as the DeepEval metrics collection rather than just exact similarity metrics like ROUGE. **5 hours**.

All members: Increased general knowledge about LLMs, studied similar projects **5 hours**.

Participated in webinars (e.g., the Nebius.ai webinar on May 16). **1.5 hours**.

8.3 Choosing, Feeding, Testing, and Gathering All Required Information of the Pre-trained Model

The Task Is Completed

The task is explained in detail in the methodology section.

Members involved: All members studied their own LLM documentation listed below to check the LLM capabilities, implemented the code and different modules such as chain of thought (LangChain), feeding through RAG, and testing via the DeepEval library. They gathered the information related to their LLM on the metrics and benchmarks as explained in the methodology section. Human evaluation of all the models and obtaining the average.

- **Sina Dalvand**: Gemma 2b, 7b; **24 hours**
- **Nazanin Ghorbani**: Mistral 7b + gathering information about multilingual capabilities, and usage rights of all models; **30 hours**
- **Mohammad Khosravi**: Llama3 8b, 70b; **26 hours**
- **Lorenzo Rosales Vasquez**: Qwen/Qwen1.5-0.5B-Chat; **24 hours**
- **Mohammad Hosseinipour**: Mixtral 8×22b, 8×7b; **24 hours**

8.4 Dataset Preparation and Fine-tuning the Selected LLM

The Task Is Completed

The results of the task are explained in section 4.1.1 (Fine-tuning).

Members Involved: **All team members**: The course catalog of the Master's degree in Computer Science was divided into 5 parts. Each member prepared the dataset as explained in the fine-tuning section. Each member worked on 8 courses, designed the original questions and answers, and paraphrased them in 6 different ways for data augmentation. **Each member: approximately 7 hours**

In addition to the task above, the following members completed the following tasks:

Sina Dalvand: Cleaning, merging, and preparing all the datasets received from other members. **7 hours**.

Mohammad Khosravi: Training and fine-tuning the Llama3 model on the prepared dataset in an iterative format. **6 hours**

8.5 Feeding the Selected LLM via RAG Pipeline and Testing the Model

The Task Is Completed

Members Involved: **All team members** searched for the most suitable architecture with respect to our use case and technologies to implement the RAG pipeline, including VectorDB, embedding models, retrieval methods, etc., as explained in the experiment section. **Each member: approximately 4-6 hours**

In addition to the mentioned work, **Sina Dalvand spent 5 to 8 hours** merging, running, and cleaning the code snippets sent by other members.

8.6 Report and Work Plan Preparation

The Task Is Completed

Preparing different sections of the report and work plan.

Members Involved: All team members: The report was divided among the 5 members. Each member spent approximately 5 to 6 hours preparing their own section of the report and work plan.

In addition to the mentioned work, **Mohammad Khosravi spent 5 to 7 hours** cleaning and unifying the explanations in the report sent by other members.

Approximate aggregation of the work done by each member:

- **Sina Dalvand : 60 to 65 hours**
- **Mohammad Khosravi: 60 to 65 hours**
- **Nazanin Ghorbani: 55 to 60 hours**
- **Lorenzo Rosales Vasqez: 45 to 50 hours**
- **Mohammad Hosseinpour : 45 to 50 hours**

8.7 Defining Requirements

Planned as the future activities

Gathering requirements from both students and university staff. Understanding what kind of questions are commonly asked and what features would be most helpful will guide our development process.

8.8 Data Collection for All Services of the UNIPD

Planned as the future activities

Gathering relevant data from various sources such as university websites, FAQs, course catalogs, etc.

8.9 Pre-processing the Raw Data for All Services of the UNIPD

Planned as the future activities

Pre-processing this data to ensure consistency and clean formatting.

8.10 Feeding the Entire Collected and Prepared Data Through RAG

Planned as the future activities

Training our NLP model on the collected data and fine-tuning it on our specific university data using techniques like transfer learning, RAPTOR, MMR, etc.

8.11 Development of Chat Interface

Planned as the future activities

Designing and developing a user-friendly chat interface where students can interact with our NLP model. This can be a web application, mobile app, or a chatbot integrated into existing platforms like Slack or Microsoft Teams.

8.12 Integration with Backend Systems

Planned as the future activities

Integrating our NLP service with these systems to provide seamless assistance to students.

8.13 Testing and Evaluation

Planned as the future activities

Testing our NLP model extensively to ensure it handles a wide range of questions accurately. Collecting feedback from users and iterating on our model and interface design accordingly.

8.14 Deployment and Maintenance

Planned as the future activities

Once everything is tested and polished, deploying our service for use by students. Continuously monitoring its performance and updating the model as needed to keep up with changing information and user needs.

8.15 Legal and Ethical Considerations

Planned as the future activities

Ensuring compliance with privacy regulations and university policies regarding data handling and user privacy. Implementing measures to prevent misuse or abuse of the service.

8.16 Documentation and Training

Planned as the future activities

Providing documentation and training materials for university staff who will be managing and maintaining the system. Also, creating user guides for students to help them make the most of the service.