Generative AI-Based Virtual Assistant Using Retrieval-Augmented Generation:
An evaluation study for bachelor projects

Dumitru Verşebeniuc[1 [0009−0004−4660−9636]], Martijn Elands[1 [0009−0004−1296−5413]], Sara Falahatkar[1], Chiara Magrone[1], Mohammad Falah[1], Martijn Boussé[1 [0000−0003−2090−0682]], and Aki Härmä[1 [0000−0002−2966−3305]]

Department of Advanced Computing Sciences, Maastricht University, Maastricht 6200 MD, The Netherlands

d.versebeniuc, m.elands, c.magrone, s.falahatkar, m.falah@student.maastrichtuniversity.nl
m.bousse, aki.harma@maastrichtuniversity.nl

Abstract. Large Language Models have been increasingly employed in the creation of Virtual Assistants due to their ability to generate human-like text and handle complex inquiries. While these models hold great promise, challenges such as hallucinations, missing information, and the difficulty of providing accurate and context-specific responses persist, particularly when applied to highly specialized content domains. In this paper, we focus on addressing these challenges by developing a virtual assistant designed to support students at Maastricht University in navigating project-specific regulations. We propose a virtual assistant based on a Retrieval-Augmented Generation system that enhances the accuracy and reliability of responses by integrating up-to-date, domain-specific knowledge. Through a robust evaluation framework and real-life testing, we demonstrate that our virtual assistant can effectively meet the needs of students while addressing the inherent challenges of applying Large Language Models to a specialized educational context. This work contributes to the ongoing discourse on improving LLM-based systems for specific applications and highlights areas for further research.

Keywords: Natural Language Processing · Retrieval-Augmented Generation · Information Retrieval · Educational Technology · AI Evaluation Metrics · Interactive AI

1 Introduction

In this paper, we propose a virtual assistant (VA) that combines recent techniques from Natural Language Processing (NLP) and Retrieval-Augmented Generation (RAG). The VA contains a multi-query and self-reflection mechanism and was tested with students from the Department of Advanced Computing Sciences (DACS) of Maastricht University as its goal was to address common challenges faced by students when seeking information about academic rules and regulations related to bachelor projects. For example, if a student cannot attend meeting

X due to reason Y, the student needs to review the rules and regulations for meeting X to determine if reason Y is acceptable. This process can be time-consuming/unsuccessful for the student and may result in contacting the project coordinator. The VA aims to provide accurate and contextually relevant responses to student inquiries, improving the student experience and addressing information overload challenges. Even though we limited ourselves to the application of a VA within bachelor projects, our approach can be generalized easily to various academic settings.

Pre-trained language models have revolutionized NLP, enabling efficient information storage and retrieval with impressive accuracy [1]. However, Large Language Models (LLMs) often struggle with precise and contextually sensitive knowledge manipulation [2]. In this paper, we address this limitation by using the RAG framework, integrating retrieval mechanisms that access up-to-date and domain-specific information from external sources, and enhancing the generative capabilities of LLMs [3].

As students might struggle with phrasing their questions precisely, we implemented a multi-query mechanism, thereby improving the retrieval process and ensuring the most relevant answers are provided. Moreover, generative LLMs can hallucinate while generating responses. In such cases, we have a self-reflection system that evaluates the response and attempts to correct it if the information is incorrect or if the answer does not align with the question. The combination of these features creates a novel approach that underscores our VA's potential to enhance the field of academic assistance tools. Moreover, rather than relying solely on predefined testers, we conducted trials with actual bachelor students, gathering their feedback to refine the system's performance in real-world scenarios. (A demonstration of the VA's functionality can be found here.)

Building upon foundational work in RAG and knowledge graph-based systems like the GRAPE model [4] and Amazon's VA for financial reconciliation [5], this VA aspires to deliver contextual information to the student query in an academic environment. Our approach is aligned with the broader trends in RAG, as surveyed extensively by Wang et al. [6], who provide a detailed overview of how RAG techniques are enhancing various NLP tasks. However, our VA distinguishes itself by specifically addressing the challenges faced by students in an academic setting, by incorporating novel strategies to better interpret and respond to student inquiries.

For this showcase, the goal of the VA is to alleviate increased workload of staff due to the addition of a new bachelor program in the Department of Advanced Computing Sciences (DACS) at Maastricht University. In particular, this caused a doubling of the number of project groups (from 36 to 72) with 6-7 students per group, putting immense pressure on project coordinators and tutors in managing student inquiries about project organization, rules and regulations, and examination details. For example, coordinators were contacted multiple times with similar, yet subtlety different, questions such as "What are the criteria for

X in case of Y?" or "What happens if I miss X, but did do Y". By focusing on project-specific information and leveraging an LLM, the VA can assist students by providing immediate answers to these common questions, reducing the burden on staff while still offering a personalized experience.

Project Preview: Bachelor projects at Maastricht University involve students working in small groups of about six to seven persons, guided by a fixed tutor, on a project divided into three subtasks across three periods. Students work part-time and full-time at different stages, concluding each period with presentations and submissions, culminating in a final report and product examination.

We conducted a pre-survey to assess student needs for a system like this before developing the VA. From 386 first-year bachelor students, 27 participants responded, with 75% indicating they had previously consulted project coordinators about project organization and rules. Participants showed a strong preference for such a system, with an average rating of 4.2 and a median of 5 on a 5-point Likert scale.

We aim to answer the following questions regarding the VA's performance:

1. How accurately can the VA retrieve relevant content to the student's queries?
2. How precisely can the VA generate a relevant response to the student's queries given the retrieved content?
3. What is the fallback mechanism employed by the VA when unable to retrieve a suitable answer to a student's question?
4. What is the average response time of the VA in providing answers to student queries while ensuring the quality and comprehensiveness of the response content?

This paper is organized as follows. In Section 2, we explain our methodology, including retrieval and generation pipeline as well as self-reflection. In Section 3 and 4, we outline the evaluation process and the experiments. Results are reported in Section 5. Discussion and conclusion can be found in Section 6 and Section 7.

2 Methods

The following section details the architecture, processes, and methodologies employed in the VA pipeline (see Figure 1). The VA pipeline receives a user question as input to the retrieval pipeline Figure 1.1. The retrieval pipeline collects all necessary information such as relevant document chunks and similar Question and Answer (Q&A) examples to the user question, and then sends all collected information to the generation pipeline Figure 1.2. The generation pipeline organizes the information from the retrieval phase and generates the response with a set of instructions, which it then sends to the self-reflection part Figure 1.3.

The self-reflection part is our fallback mechanism, which evaluates the generated response for hallucinations and relevance. If one part of self-reflection fails, it

attempts to correct itself or asks for clarifications from the user. The detailed mechanisms of the VA pipeline are discussed later in this paper, with numerical references as illustrated in Figure 1.

Retrieval-Augmented Generation: RAG is a technique that combines the strengths of both retrieval-based and generation-based methods. This hybrid method involves retrieving relevant documents from a large corpus and using this information to generate more accurate and contextually enriched responses [7]. The integration of RAG allows our VA to access up-to-date information, thus overcoming the temporal limitations of static pre-trained models.

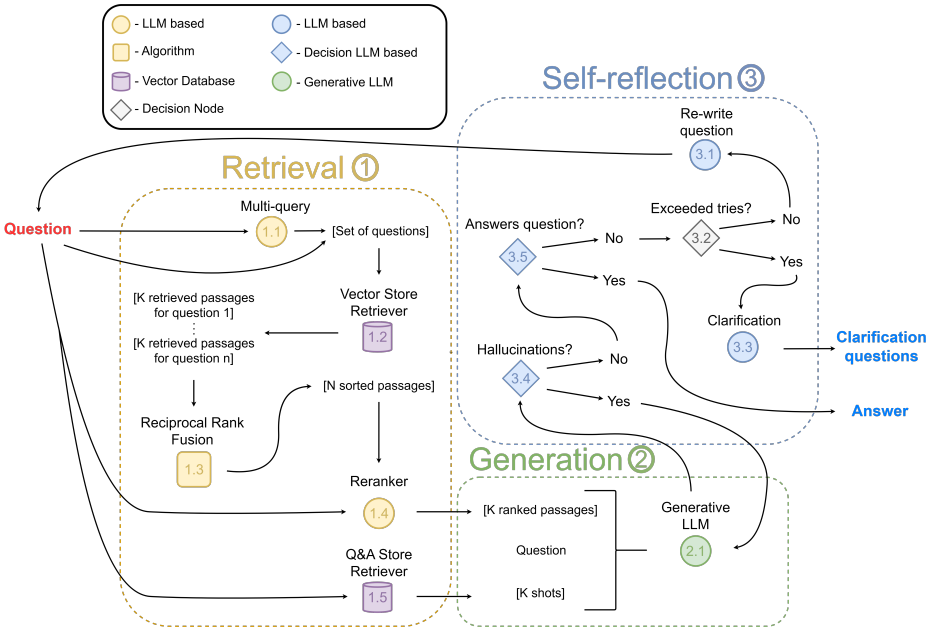

Fig. 1: The VA architecture consists of retrieval, generation, and self-reflection parts. Retrieval ① collects the relevant information to the user question. Generation ② processes the retrieval information and structures it to generate the response. Self-reflection ③ is the fallback mechanism that ensures the correctness of the response and understanding of the user question.

Manual Parsing: Since we are dealing with a not large amount of data, we consider manual parsing, where we split documents into relevant chunks, which helps avoid including noise during the retrieval process.

2.1 Retrieval Pipeline ①

The retrieval pipeline is a fundamental component of our VA system, designed to efficiently retrieve relevant information from a corpus of documents.

Multi-Query Retrieval (1.1) Multi-query retrieval enhances the retrieval process [8] by generating multiple versions of a user query, capturing different perspectives, and ensuring a broader range of relevant documents. This approach leverages LLMs to create alternative queries, improving the accuracy and comprehensiveness of the VA's responses by considering diverse and contextually relevant documents.

Vector Database (1.2) The vector database is the whole basis of our retrieval system, enabling efficient storage and retrieval of document embeddings [9]. The embedding model transforms documents and queries into dense vector representations, ensuring quick and accurate retrieval of relevant documents based on similarity metrics such as cosine similarity, maximal marginal relevance (MMR), etc.

Embedding Techniques Embedding techniques are crucial for converting text into numerical vectors that models can understand and manipulate. We can use advanced embedding models from Google [10], OpenAI [11], Mistral [12], or BGE [13] to generate high-quality numeric text representation. These embeddings capture semantic meanings and relationships within the text [14], which is essential for both the retrieval and generation phases.

Reciprocal Rank Fusion (1.3) Reciprocal Rank Fusion (RRF) is a method used in RAG to combine and rerank documents from multiple retrieval queries. By assigning a score to each document based on its ranking across various result lists, RRF effectively consolidates different retrieval outputs. The formula for RRF is given by:

$$\text{score}_{\text{RRF}}(D) = \sum_{i=1}^{k} \frac{1}{k + \text{rank}_i(D)}$$

where $\text{rank}_i(D)$ is the rank of document D in the i-th result list, and k is a small constant, often set to 60. This method enhances retrieval accuracy by emphasizing higher-ranked documents while maintaining robustness against discrepancies in individual list rankings.

Reranker (1.4) The reranker model is used to obtain a better relevance score between a question and a document. There are two different types of reranked models:

- *Cross-encoder model* uses both the question and the document as inputs and directly outputs a similarity score, rather than generating embeddings. [13]

- *LLM-Embedder*, unlike embeddings, which primarily assess semantic similarity between a document and a query, the LLM-Embedder can provide precise scores for how well a document answers a given query using fine-tuned generative LLMs. [15]

Few-shot retriever (1.5) Since we have already received a Q&A dataset[1] from the bachelor's project coordinator, storing them in the Vector Store allows us to retrieve similar questions and example truth answers to provide few-shot examples to our generative LLM, thereby giving example answers to the question.

2.2 Generation Pipeline ②

In the generation process of a VA, using low-temperature 0.2 and structured XML prompts (shared on our GitHub) helps to improve answer accuracy and factual grounding. Low temperatures reduce randomness, leading to more conservative and reliable responses, while structured XML prompts facilitate better input understanding and contextual relevance by clearly defining different elements of a prompt. This combination ensures that the LLM (2.1) generates responses that are consistent and closely aligned with the provided facts. Moreover, this technique optimizes the VA's functionality in handling detailed and complex queries [16].

2.3 Self-Reflection ③

The self-reflection process is about assessing and refining the response generated by the VA to obtain accuracy and relevance [17]. Here's how it functions:

- *Re-write Question* (3.1): If the generated answer is not satisfactory, or if it does not directly address the user's query, the process includes a mechanism to rewrite the query. This might involve rephrasing, correcting, or breaking down the question into more manageable parts to match the available data better. By rephrasing the question, the VA can better match the query with relevant documents in the database, leading to more accurate responses.
- *Exceeded Tries* (3.2): This checks whether the process of rewriting the question has exceeded a set number of tries. If so, it indicates that the system is struggling to understand the query or to find relevant information, and a different approach might be needed such as clarification questions.
- *Clarification Questions* (3.3): If necessary, the system can pose clarification questions to the user. This step is especially important when the query is ambiguous or lacks specific details needed for an accurate response. By engaging the user in a dialogue, the VA can gather additional information to better address the user's needs.
- *Hallucinations Check* (3.4): This step involves checking whether the generated responses include hallucinated information, which is a common issue with generative models. This is achieved using a generative LLM as a decision-maker to evaluate the generated answer against the retrieved documents.

  The hallucination detection mechanism works by comparing the response to the set of facts obtained during the retrieval phase. The generative LLM

[1] The Q&A dataset consists of a set of questions posed by students and answers provided by the project coordinator. The questions are typically inquiries about project-related matters, such as assessment, attendance, or deliverables, while the answers provide feedback or clarification on those inquiries.

uses a specific prompt to determine whether the answer is grounded in the provided facts.

If the LLM determines that the answer contains information not supported by the facts, it requests the Generative LLM (2.1) to regenerate the response until the Hallucination Checker (3.4) produces a satisfactory result.

– *Answer Check* (3.5): This involves verifying whether the generated response adequately addresses the user's question. The system employs a generative LLM as a decision-maker to determine if the answer resolves the query. If the answer is found lacking, the system may trigger the query rewriting mechanism (3.1) or ask for clarification (3.3) from the user.

3 Evaluation

We evaluated the VA's performance in the first-year second-semester bachelor projects. In particular, we assessed its ability to retrieve and generate accurate and relevant responses to student queries, as well as its overall effectiveness and user satisfaction. The vector search evaluation involves assessing the effectiveness of different embedding models and retrieval configurations. The model's performance was evaluated based on its ability to retrieve relevant documents for a given query. The evaluation parameters included search type (similarity, MMR, or similarity score threshold) and various keyword arguments to fine-tune the retrieval process.

In addition to the experiments, we used an offline dataset to evaluate both parts of the pipeline. For this, we use metrics from the RAGAS evaluation framework [18] and a custom precision3 metric. The evaluation process was automated using a script that iterated through the dataset of question-answer pairs, retrieved relevant documents, and generated responses to calculate the metrics. The next two subsections outline the evaluation metrics.

Retrieval evaluation To measure the performance of the retrieval part, we processed each question to retrieve the most relevant documents. The metrics used for this part are Context Precision and Context Recall.

Context precision is a measure used to see how well a system ranks important pieces of information. Ideally, the most relevant pieces of information (called chunks) should be in a high rank [19]:

$$\text{Context Precision@K} = \frac{\sum_{k=1}^{K}(\text{Precision@k} \times v_k)}{\text{total number of relevant items in top } K \text{ results}} \quad (1)$$

where $\text{Precision@k} = \frac{\text{true positives@k}}{\text{true positives@k+false positives@k}}$

Context Recall measures the extent to which retrieved context aligns with the ground truth (annotated answer) [19]:

$$\text{Context Recall} = \frac{|\text{GT sentences that can be attributed to context}|}{|\text{Number of GT sentences}|} \quad (2)$$

where Ground Truth (GT) refers to an annotated or expected correct answer.

Furthermore, we also used a customized precision metric (see Equation 3) to evaluate the proportion of correctly retrieved documents by the system with the manually labeled correct documents.

$$\text{Custom precision} = \frac{|\text{correctly retrieved documents}|}{|\text{manually considered relevant documents}|} \tag{3}$$

where relevant documents are manually incorporated into the dataset.

Generation evaluation Once the relevant documents were retrieved, the next step involved generating responses using the retrieved content(s). The evaluation metrics for this are Answer Relevancy and Faithfulness. These metrics guarantee the relevancy, accuracy, and fidelity of the generated responses. The generation process was repeated for different generative LLMs [20].

Answer Relevancy assesses how pertinent the generated answer is to the given prompt [19]. It uses artificial questions based on the generated answer. Hence, it uses the mean cosine similarity between the original question and a number of artificial questions.

$$\text{Answer Relevancy} = \frac{1}{N} \sum_{i=1}^{N} \cos(\mathbf{E}_{g_i}, \mathbf{E}_o) \tag{4}$$

where

- E_{g_i} represents the embedding of the generated question i.
- E_o represents the embedding of the original question.
- N is the number of generated questions, typically 3 by default.

Faithfulness evaluates whether the generated answer accurately represents the information in the retrieved context [19]:

$$\text{Faithfulness} = \frac{|\text{Inferred claims from the given context in generated answer}|}{|\text{Total claims in the generated answer}|} \tag{5}$$

4 Experiments

A series of tests with DACS bachelor students were performed to ensure that the system was user-friendly and to identify any other improvements. Furthermore, we also wanted to gather quantitative measurements about the system's performance. These tests were conducted on June 20 and June 21, 2024.

The students who participated were divided into two groups: A and B. Each group received eight scenarios accompanied by a multiple-choice question to assess their knowledge of the rules and regulations and examination details. An "I don't know" option was added to limit guessing behavior. For the first four scenarios, participants in Group A could not use the VA, while participants in

Group B were encouraged to use the VA. For the last four scenarios, the roles were reversed, making the tests complementary. We ensured no leakage occurred.

At the start of the test, participants received a list of basic background questions. During the test, the first four scenarios also featured a question asking if the participant would prefer to ask the VA or the coordinator first. The last four scenarios are accompanied by a question if the VA helped answer the scenario on a Likert scale where 1 was not helpful at all and 5 was very helpful. At the end of the test, they were asked some general questions, for example, about the response time.

For example, scenario 3 is: "Luca carefully planned his day to arrive on time for the final product and report examination. However, the bus he took left the stop earlier than scheduled, causing him to miss the exam. Now he is worried about his project grades. What are the consequences for his project grades?"

Answers:

- **He will receive a NG for the project**
- He will receive the same grade as everyone else in the team
- He will receive a lowered individual grade for the project

The full survey is shared online on our GitHub page.

5 Results

5.1 Automated Retrieval Augmented Generation Assessments

Table 1 shows the summary of the metrics presented in section 3 for the system with different generative LLMs. While Gemini 1.0 Pro slightly outperforms GPT-3.5 in context precision, GPT-3.5 excels in other key areas like context recall, answer relevancy, and faithfulness. This suggests that GPT-3.5 generates more relevant and accurate content, making it better suited for our VA.

Table 1: Evaluation metrics for different generative LLMs

Metric	GPT3.5	Gemini1.0Pro
Context precision	88%	**89%**
Context recall	**42%**	41%
Answer relevancy	**57%**	37%
Faithfulness	**43%**	32%
Customized precision	**77%**	**77%**

5.2 Survey Results

In total, 64 participants completed the survey: 34 in group A and 30 in group B. As described in Section 4, participants were asked about their enrolled program,

year of study, and whether they had ever read the rules and regulations. We found that 55% of the participants had read the rules and regulations.

Participants were also asked how they would respond to the first four presented scenarios: whether they would contact the project coordinator or a VA for assistance. On average, 71% of participants preferred contacting the project coordinator before the VA. This preference was consistent across all scenarios, even without prior use of the VA.

Students' Knowledge of Project Rules: The survey revealed varied levels of understanding among participants regarding the rules for skipping project meetings. However, based on responses, both groups demonstrated similar prior knowledge. This suggests that misunderstandings about the rules and regulations were widespread and not confined to any particular group. The correct answer is that one meeting can be skipped in phases 1 and 2 combined, one meeting in phase 3 without consequence, two meetings result in a lower project grade, and three meetings result in an NG. Out of the respondents, only 7 participants had the correct understanding of these rules.

Additionally, 28 participants believed that one meeting could be skipped per phase, which shows a partial understanding of the rules but lacks specificity regarding phase combinations and consequences. Another 11 participants thought that one meeting could be skipped without consequences, with the second and third meetings leading to a grade reduction and NG respectively, which again is partially correct but not entirely accurate.

There was also a group of 5 participants who admitted to not knowing the rules, indicating a clear gap in knowledge. Furthermore, some participants provided detailed answers that did not fully align with the correct rules, showing a mixture of partial knowledge and misconceptions about the consequences of missing project meetings. This data suggests that while most participants have a general idea about the meeting policies, there is significant room for improvement in ensuring that all participants have a precise and thorough understanding of the rules.

Scenario Results: In Table 2, we show the results for all scenarios with and without VA assistance. The data indicates an overall improvement in performance when participants used the VA. Performance improvement when using the VA can be seen based on the data collected from participants' responses to the scenarios.

The results indicate that, in general, the performance of participants improved when using the VA, as evidenced by the higher percentages of correct answers in most scenarios. For instance, in Scenario 1, the percentage of correct answers increased from 32.4% without the VA to 46.7% with the VA. Similarly, in Scenario 6, the correct answers jumped from 50% to 73.5%. These improvements suggest that the VA can assist students in answering organizational questions related to rules and regulations. However, there are notable exceptions, such as Scenarios 5 and 7, where the performance did not improve as expected. In Scenario 5, the

Table 2: Impact of VA on Correct Answers and "I Don't Know" Responses showing improvements in accuracy and reducing uncertainty in student responses across most scenarios.

Scenarios	Correct Answers		"I Don't Know"	
	Without VA	With VA	Without VA	With VA
1	32.4%	46.7%	29.4%	16.7%
2	17.6%	26.7%	32.4%	6.7%
3	32.4%	56.7%	50.0%	16.7%
4	2.9%	26.7%	41.4%	26.7%
5	53.3%	35.3%	30.0%	2.9%
6	50.0%	73.5%	30.0%	5.9%
7	63.3%	50.0%	3.3%	2.9%
8	36.7%	32.4%	30.0%	32.4%

percentage of correct answers decreased from 53.3% without the VA to 35.3% with the VA. Similarly, in Scenario 7, the correct answers decreased from 63.3% to 50%. The reasons why that might have happened are explained in 5.3.

Additionally, the "I don't know" responses generally decreased with the use of the VA, indicating that the assistant helped reduce uncertainty among participants. For example, in Scenario 2, the "I don't know" responses dropped from 32.4% without the VA to 6.7% with the VA. This further supports the utility of the VA in providing clearer guidance and information to the students.

When scenarios 5 and 7 are left out-of-scope for reasons mentioned in section 5.3, we can state that we have significantly improved the scenarios with a 95% confidence interval for the correct answers and the reduction of "I don't know" submissions over the usage without a VA.

Scenario Feedback: Table 3 presents the distribution of helpfulness scores received by the VA for various scenarios. Participants rated the VA on a Likert scale ranging from 1 (not helpful) to 5 (extremely helpful). This table allows us to analyze how the VA's perceived helpfulness varies across different tasks it was asked to perform.

Scenarios 5 and 6, show a significant portion of participants (over 40%) rating the VA as extremely helpful (score 5). This suggests the VA effectively assisted users in those specific situations. In contrast, scenarios with a wider range of scores, such as Scenario 1 and 8, indicate a more diverse range of user experiences. This implies a mixed perception of the VA's usefulness in those Scenarios, which might be caused by a difference of opinion between students and the rules and regulations.

Table 3: Distribution of Helpfulness Scores of VA for each Scenario

Scenarios	**On a scale of 1 to 5, how helpful was the VA?**				
	1	2	3	4	5
1	6.67%	16.67%	30%	20%	26.67%
2	10%	16.67%	33.33%	26.67%	13.33%
3	10%	16.67%	13.33%	26.67%	33.33%
4	6.67%	16.67%	40%	30%	6.67%
5	11.76%	5.88%	8.82%	29.41%	44.12%
6	0.00%	8.82%	2.94%	47.06%	41.18%
7	0.00%	8.82%	17.65%	35.29%	38.24%
8	2.94%	20.59%	20.59%	32.35%	26.67%

Overall Feedback Experience: Figure 2 shows the overall satisfaction of students with the VA system. Each bar represents a summary of the Likert scale responses for three different questions:

- "Do you think that we developed a valuable VA? By valuable, we mean that it can reduce staff members' workload and increase response time for students." This is also on a Likert scale where 1 is "Not valuable at all" and 5 is "Extremely valuable".
- "How was the response time?" This is on a Likert scale where 1 is "Worse than expected" and 5 is "Better than expected".
- In the third case, we summarized the responses to the question posed to students after each scenario, "Was the VA helpful in answering this question?" This was also rated on a Likert scale.

The chart shows the distribution of responses for each category, indicating a generally positive reception to the VA system. For response time, students predominantly rated it as "Neutral", "Satisfied", and "Very satisfied", reflecting a positive view of the VA's promptness. The perceived value of the VA was also highly rated, with most students finding it valuable for reducing staff workload and improving response times. In evaluating the VA's helpfulness in answering specific questions, the majority of responses were "Satisfied" and "Very satisfied", demonstrating the VA's effectiveness across various scenarios. Overall, the feedback highlights a positive reception, appreciating both the response time and added value of the VA, while also suggesting areas for further improvement.

Response-Time Analysis: The analysis of response times indicated that the VA achieved an average response time of 10.045 seconds across all queries, with a standard deviation of 2.39 seconds. While this performance was satisfactory for the purposes of initial testing, there is room for improvement to ensure more consistent and reliable response times in real-life deployment scenarios.

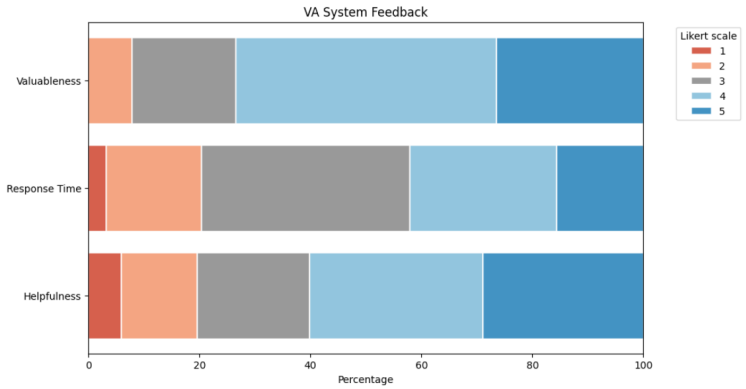

Fig. 2: Overall student satisfaction with the VA system showing a generally positive reception, based on students' rating

Table 4: The table categorizes the most asked topics related to group projects, revealing the frequency of specific concerns among students. It identifies the most discussed issues, focusing on areas that may require more attention.

Topic	**Representative Words**	**Counts**
0	[Missed pre-examination and final presentation consequences]	97
1	[Dealing with Inactive Group Members in a Project]	23
2	[help, context, new update]	12
3	[Force Majeure and Attendance Issues]	9
4	[Factors affecting individual grades in projects]	8
5	[Communication with Tutors and Examiners]	4
6	[What happens when the whole group misses something]	4
7	[Consequences of a given case]	4
8	[Phase 3 Consequences]	4

For instance, during periods of high demand, such as when multiple queries are submitted simultaneously, the VA exhibited increased response times, suggesting the need for enhanced handling of concurrent requests to maintain efficiency.

5.3 Refinements

Student testing revealed areas for improvement in the VA's performance. For example, Scenario 5 highlighted the need for more comprehensive training data, as the VA struggled to distinguish between a project meeting and an exam due to missing information in the reference document. Similarly, Scenario 7 showed the limitations of the VA in handling unforeseen scenarios not covered by the rules. These findings suggest a need to refine the training data and potentially develop mechanisms for the VA to handle situations outside its current knowledge base. Additionally, the feedback received from students provides valuable insights for

improving the system to better align with their needs. This feedback will be shared in detail on our GitHub page.

6 Discussion

While our VA shows promising results, we identified several limitations during its development, testing, and design phases.

Firstly, we had to disable the copy-pasting functionality on the client side after the second day of testing. This decision was driven by our observation that students were not following the instructions for paraphrasing or formulating their own questions about the given scenarios.

Another challenge we faced was determining the appropriate evaluation metrics to use in this setting. In our study, we decided to utilize the RAGAS framework, along with a customized precision metric, to assess the system's performance. However, the field is rapidly evolving, with numerous metrics available, which makes it difficult to choose the most suitable ones.

Additionally, we found that traditional Large Language Models (LLMs) struggle with processing long input texts, as noted in previous research [21]. This limitation can result in a loss of detail, which is particularly problematic when evaluating the system's responses. The coordinator's general answers, which tend to be shorter, are often less detailed than the generated responses, leading to potential inconsistencies in evaluation.

During testing, we also observed that the system occasionally misidentifies courses as skill classes. This issue suggests that the VA may benefit from being provided with a predefined list of skill class names for verification purposes. Such an enhancement could reduce the frequency of these misidentifications and improve the system's overall accuracy.

Finally, we encountered a limitation related to our inability to access personal information, such as students' years or their respective coordinators. This lack of information reduces our ability to fully tailor the system to individual student cases. However, this challenge could potentially be overcome by integrating the VA with Maastricht University's Learning Management System or by adding a feature to the User Interface that allows users to input their contact information directly.

7 Conclusions

The development of a VA for DACS students has shown potential in alleviating the workload of staff and providing a significant impact on fast and accurate information to students. By leveraging advanced NLP techniques and integrating RAG systems, the VA can effectively support students with queries related to rules, regulations, and examination details. The retrieval pipeline, incorporating various embedding models and advanced retrieval techniques, ensures the accuracy and relevance of the information provided.

Our evaluation metrics showed that we can accurately retrieve documents with a context precision of 88% and a context recall of 42%. Generating relevant answers based on these documents is done at around 57% (answer relevancy) and faithfulness of 43%. All of this is done by the system in around 10 seconds which the students seem to be satisfied with.

By integrating self-reflection, the system can evaluate its own outputs and decision-making processes, allowing it to identify inefficiencies or errors and adjust its methods accordingly. This process is essential for maintaining the reliability of the system; if uncertainties or ambiguities arise, the system can request clarifications, thereby preventing potential errors and refining its responses.

Testing with students has provided valuable insights into the practical application of the assistant. This has led to improvements in the system and indicators for further possible enhancement. Despite its limitations, the VA represents a significant step forward in educational technology, enhancing the student experience and reducing the administrative burden on academic staff. Future work will focus on addressing the identified limitations and exploring additional functionalities to further improve the system's performance and user satisfaction.

Future VA iterations should integrate feedback from the testing participants to align more closely with their expectations, including adjustments like incorporating FAQs and relevant contact information. An example of this is a force majeure template. Additionally, the approach to verify model hallucinations using semantic entropy [22] and employing techniques like Reverse HyDE [23] and contrastive learning [15] will refine the retrieval accuracy and relevance of information provided by the VA.

References

[1] Regina Gubareva and Rui Lopes. Virtual assistants for learning: A systematic literature review. In *Proceedings of the 12th International Conference on Computer Supported Education - Volume 1: CSEDU,*, pages 97–103. Institute for Systems and Technologies of Information, Control and Communication, SciTePress, 2020.

[2] Patrick 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, 2020.

[3] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Knowledge Discovery and Data Mining '24, page 6491–6501, New York, NY, USA, 2024. Association for Computing Machinery.

[4] Mingxuan Ju, Wenhao Yu, Tong Zhao, Chuxu Zhang, and Yanfang Ye. Grape: Knowledge graph enhanced passage reader for open-domain question answering. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: The 2022 Conference on Empirical Methods in Natural Language Processing*, pages 169–181, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[5] Daksha Yadav, Sabrina Zhang, Tom Jin, Prakash Krishnan, and Des Clarke. Generative ai based virtual assistant for reconciliation research. In *The Association for the Advancement of Artificial Intelligence 2024 Workshop on AI for Financial Services*, 2024.

[6] Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, and Chun Jason Xue. Retrieval-augmented generation for natural language processing: A survey, 2024.

[7] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

[8] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Lin-*

guistics: The 2023 Conference on Empirical Methods in Natural Language Processing, pages 9248–9274, Singapore, December 2023. Association for Computational Linguistics.

[9] Yikun Han, Chunjiang Liu, and Pengfei Wang. A comprehensive survey on vector database: Storage and retrieval technique, challenge. *Computing Research Repository*, abs/2310.11703, 2023.

[10] Text embeddings API, Generative AI on Vertex AI, Google Cloud. `https://cloud.google.com/vertex-ai/generative-ai/docs/model-reference/text-embeddings-api`. [Accessed 11-07-2024].

[11] OpenAI. Open AI Text Embedding Model. `https://platform.openai.com/docs/guides/embeddings`. [Accessed 11-07-2024].

[12] Embeddings — Mistral AI Large Language Models — docs.mistral.ai. `https://docs.mistral.ai/capabilities/embeddings/`. [Accessed 11-07-2024].

[13] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024.

[14] Rajvardhan Patil, Sorio Boit, Venkat Gudivada, and Jagadeesh Nandigam. A survey of text representation and embedding techniques in nlp. *Institute of Electrical and Electronics Engineers Access*, 11:36120–36146, 2023.

[15] Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. Retrieve anything to augment large language models, 2023.

[16] OpenAI. Prompt engineering. `https://platform.openai.com/docs/guides/prompt-engineering/strategy-provide-reference-text`. [Accessed 12-07-2024].

[17] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7036–7050, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[18] Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. RAGAs: Automated evaluation of retrieval augmented generation. In Nikolaos Aletras and Orphee De Clercq, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta, March 2024. Association for Computational Linguistics.

[19] Ragas Documentation. Metrics — component-wise evaluation. `https://docs.ragas.io/en/stable/concepts/metrics/index.html`, 2024. [Accessed July 2024].

[20] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17754–17762, 03 2024.

[21] Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning. *Computing Research Repository*, abs/2404.02060, 2024.

[22] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, Jun 2024.

[23] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada, July 2023. Association for Computational Linguistics.