UNIVERSIDAD CATÓLICA DE CUENCA

# UNIVERSIDAD CATÓLICA DE CUENCA

*Comunidad Educativa al Servicio del Pueblo*

## UNIDAD ACADÉMICA DE INFORMÁTICA, CIENCIAS DE LA COMPUTACIÓN E INNOVACIÓN TECNOLÓGICA

## CARRERA DE SOFTWARE

## TÍTULO

# Large Language Model (LLM) implementada en la documentación de la Universidad Católica de Cuenca

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SOFTWARE**

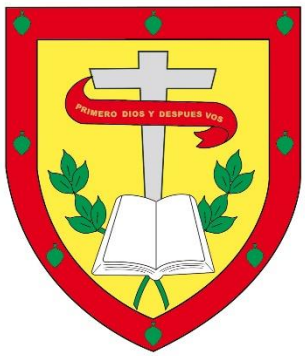
**AUTOR: PEDRO FERNANDO ÁLVAREZ SARMIENTO**

**DIRECTOR: ING. ANDRÉS SEBASTIÁN QUEVEDO SACOTO, MGS**

**CUENCA – ECUADOR**

**2024**

**DIOS, PATRIA, CULTURA Y DESARROLLO**




# UNIVERSIDAD CATÓLICA DE CUENCA

*Comunidad Educativa al Servicio del Pueblo*

## UNIDAD ACADÉMICA DE INFORMÁTICA, CIENCIAS DE LA COMPUTACIÓN E INNOVACIÓN TECNOLÓGICA

## CARRERA DE SOFTWARE

## TÍTULO

Large Language Model (LLM) implementada en la documentación de la Universidad Católica de Cuenca.

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SOFTWARE**

**AUTOR: PEDRO FERNANDO ÁLVAREZ SARMIENTO**

**DIRECTOR: ING. ANDRÉS SEBASTIAN QUEVEDO SACOTO MGS**

**CUENCA - ECUADOR**

**2024**

**DIOS, PATRIA, CULTURA Y DESARROLLO**

## Declaratoria de Autoría y Responsabilidad

**Pedro Fernando Álvarez Sarmiento** portador(a) de la cédula de ciudadanía Nº **0106620222**. Declaro ser el autor de la obra: **"Large Language Model (LLM) implementada en la documentación de la Universidad Católica de Cuenca"**, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Cuenca, **24 de junio de 2024**

F: ……..

**Pedro Fernando Álvarez Sarmiento**

C.I. **0106620222**

**CERTIFICO:**

Certifico que el presente Trabajo de titulación fue desarrollado por **Pedro Fernando Álvarez Sarmiento** con el título "LARGE LANGUAGE MODEL (LLM) IMPLEMENTADA EN LA DOCUMENTACION DE LA UNIVERSIDAD CATÓLICA DE CUENCA", bajo mi supervisión.

Firmado electrónicamente por:
ANDRES SEBASTIAN QUEVEDO SACOTO

F: .....

Mgtr. Andrés Sebastian Quevedo Sacoto

# DEDICATORIA

Dedico este logro a mis padres y hermano por formar parte de este proceso y ser clave en la obtención en el alcance de mis metas.

# RESUMEN

En la actualidad, la inteligencia artificial (IA) ha realizado progresos impresionantes. Un avance notable es la aparición de los Modelos de Lenguaje Amplio (LLM), capaces de generar e interpretar datos de lenguaje natural. Entre estos modelos han acaparado una gran atención por su notable capacidad de generación de textos y su interfaz de usuario mejorada. Las instituciones académicas se enfrentan al reto de acceder a grandes cantidades de información de manera eficiente. Este problema se ve agravado por la creciente cantidad de documentos académicos, la dispersión de la información en distintos repositorios y el tiempo y los recursos necesarios para buscar y filtrar esta información, lo que representa una importante carga de trabajo para profesores y estudiantes. Para abordar este problema, este trabajo propone un asistente impulsado por IA integrado con LLM y un sistema de software basado en una arquitectura de microservicios. El asistente ofrece respuestas claras y contextualmente relevantes, haciendo más eficientes los procesos de recuperación de información académica. Este artículo propone un asistente potenciado por IA, que cubre aspectos de integración tanto de modelos de IA como de software. Además, utiliza asistentes inteligentes para gestionar información académica, sirviendo de modelo para futuras implementaciones.



**Palabras Clave:** Asistente; API; LLM; Recuperación de Conocimiento; NLP

# ABSTRACT

Currently, artificial intelligence (AI) has made impressive progress. One notable development is the emergence of Large Language Models (LLMs), capable of generating and interpreting natural language data. Among these models have gained widespread attention for their remarkable text generation capabilities and improved user interface. Academic institutions face challenges in accessing vast amounts of information efficiently. This problem is compounded by the increasing amount of academic documents, the dispersion of information in different repositories, and the time and resources required to search and filter this information, representing a significant workload for professors and students. To address the issue, this paper proposes an AI-powered assistant integrated with LLMs and a software system based on a microservices architecture. The Assistant offers clear and contextually relevant answers, making academic information retrieval processes more efficient. This article proposes an AI-powered assistant, covering integration aspects of both AI and software models. Moreover, it uses intelligent assistants to manage academic information, serving as a model for future implementations.



**Keywords:** Assistant; API; LLM; Knowledge Retrieval; NLP

**LARGE LANGUAGE MODEL (LLM) IMPLEMENTED IN THE DOCUMENTATION OF THE UNIVERSIDAD CATÓLICA DE CUENCA.**

**LARGE LANGUAGE MODEL (LLM) IMPLEMENTADA EN LA DOCUMENTACIÓN DE LA UNIVERSIDAD CATÓLICA DE CUENCA.**

JCSE Journal of Computing Science and Engineering

Regular Paper

Journal of Computing Science and Engineering
Vol.00, No.0, Jan 0000, pp. 000-000

# Large Language Model (LLM) implemented in the documentation of the Universidad Catolica de Cuenca

**Pedro Alvarez, Sebastian Quevedo**
Unidad Académica de Informática, Ciencias de la Computación, e Innovación Tecnológica, Grupo de Investigación Simulación, Modelado, Análisis y Accesibilidad (SMA$^2$), Universidad Católica de Cuenca, 010107, Cuenca, Ecuador
**pedro.alvarez.22@est.@ucacue.edu.ec, asquevedos@ucacue.edu.ec**

**Abstract**
Currently, artificial intelligence (AI) has made impressive progress. One notable development is the emergence of Large Language Models (LLMs), capable of generating and interpreting natural language data. Among these models have gained widespread attention for their remarkable text generation capabilities and improved user interface. Academic institutions face challenges in accessing vast amounts of information efficiently. This problem is compounded by the increasing amount of academic documents, the dispersion of information in different repositories, and the time and resources required to search and filter this information, representing a significant workload for professors and students. To address the issue, this paper proposes an AI-powered assistant integrated with LLMs and a software system based on a microservices architecture. The Assistant offers clear and contextually relevant answers, making academic information retrieval processes more efficient. This article proposes an AI-powered assistant, covering integration aspects of both AI and software models. Moreover, it uses intelligent assistants to manage academic information, serving as a model for future implementations.

**Category:** Smart and Intelligent Computing

**Keywords:** Assistant; API; LLM; Knowledge Retrieval; NLP

## I. INTRODUCTION

Recent advances in artificial intelligence (AI) have marked the beginning of a new digital era [1, 2, 3, 4]. A particular AI model known as Large Language Models (LLMs) has stood out prominently, offering incredible capabilities in generating and interpreting natural language data [5].

LLMs can develop a significant new text based on small input requests [6]. The broad public release of ChatGPT by OpenAI in November 2022 marked a substantial increase in the fundamental ability of the software to create new text using refined models (GPT-3.5), along with an improved user interface [7]. This release has increased public conversation about how LLMs can impact educational integrity [6].

Advanced languages, such as LLM, have the potential to provide professional assistance in various fields, including education [8]. They offer a range of benefits, such as enhancing the user's interaction with digital platforms, particularly in areas related to customer service. These languages have sophisticated capabilities to improve the user experience by providing efficient and effective solutions.

According to [9], using AI, especially LLMs, in software engineering education is not just a trend but a necessity. The introduction of LLMs has posed crucial questions about their impact on the process and implementation of software engineering. This AI-powered Assistant is geared towards simplifying the task of retrieving academic information. It is anticipated that this technology is expected to improve the academic staff.

Application Programming Interface (API) integration holds significant implications for developers seeking to harness the power of external services and functionalities within their serverless applications. As explored in the study [10], the effectiveness of API integration mechanisms becomes paramount in enabling

Open Access yy.5626/JCSE.2011.5.2.xxx http://jcse.kiise.org



Received 00 Month 2011, Accepted 00 Month 2011, Revised 00 Month 2011

* Corresponding Author



pISSN: 1976-4677 eISSN: 2093-8020

developers to leverage a wide range of services and data sources in their serverless workflows.

Educational institutions must have an efficient system that ensures quick access to information and prompt responses to any queries related to administrative and academic documents. This problem is solved by AI-powered assistants making use of technology and automation. This can facilitate document management and enable students and administrative staff to access information smoothly. However, finding immediate information can be challenging due to the many academic and administrative documents available. To overcome this challenge, a modern LLM-based assistant was developed to leverage documents as a single source of information to respond to academic staff queries effectively. Institutions store a vast amount of valuable information in their documents, which contain relevant details related to their functioning. However, it can be challenging for students and administrative staff to access this information due to the scattered documentation, leading to long wait times.

The Assistant is powered by OpenAI's advanced natural language processing models. Its purpose is to provide quick and accurate answers to user queries related to accessing academic information, thereby eliminating the need for manual scanning of multiple documents. This project aims to solve the problem of long wait times for students and staff when seeking answers to their questions or concerns.

The article presents the comprehensive process of developing an innovative assistant integrated with AI technology and microservices software technology. It covers Related work, Methodology, Results, Conclusions, and Future Work.

## II. RELATED JOBS

Numerous studies in the field of AI have explored various professional domains such as medicine, education, and public administration. These studies have investigated the potential of modern technologies in these fields. For example, AI can assist with image interpretation [11] and the segmentation of tumors [12] in the medical field. In public administration, question-answering systems are being developed with the help of AI [2]. Similarly, in the field of education, AI-powered tools such as ChatGPT [3] are being used to support students in decision-making [4].

[1] discusses the potential of an OpenAI chatbot to transform medical fields, including diagnostics, treatment planning, and healthcare delivery, complementing the idea of increased performance in professional and administrative fields.

Pham Duy [2] describes the development of a Question-Answering (QA) system for public administrative services in Vietnam. The focus is on building a legal QA system that provides answers related to passages in law documents. Retrieval models are used for query handling related to documents. This assistant is not only designed to use files as a source of information, but it can also guide students based on their interests.

Mhlanga [3] emphasizes the potential of ChatGPT, which utilizes LLMs to revolutionize the educational sector. One key advantage of this research is its accessibility, as it is capable of handling multiple languages, making it more accessible to people all over the world.

Abu et al. [4] proposed a chatbot that integrates LLMs to mediate between students, assisting them in decision-making during the learning process. The study exhibited positive feedback on the chatbot's functionality. This research aims to leverage the latest advancements in AI, particularly the development of LLMs, to optimize communication, simplify processes, and enhance support in professional and educational settings. Furthermore, this study intends to demonstrate the potential of AI-driven assistants to boost human capabilities.

[11] describe the development of a software system that uses a microservice architecture to ensure scalability, maintainability, and efficient AI model integration. In this paper, microservices facilitate easier deployment and management of assistant components, as each service can be developed, deployed, and scaled independently.

The limitations highlighted in the studies [10, 2, 3, 4, 1] emphasize the need to address various areas to improve related to AI-powered academic query management systems, such as reliance on virtual machines, lack of direct engagement with individual learners' needs, heavy dependency on language-specific models, and potential ethical concerns like bias and privacy breaches.

Incorporating these insights into academic query management systems can lead to significant improvements in various aspects of the proposed Assistant. For instance, addressing the reliance on previously assigned virtual machines can ensure efficient query handling, avoiding overloads and slow startups [10]. Similarly, by considering the distinctive needs of individual users and avoiding over-reliance on language-specific models, chatbots can offer more personalized and accessible support to users from diverse backgrounds [3, 2].

Moreover, integrating ethical considerations and safeguards, such as bias mitigation techniques and privacy protection measures, can ensure responsible and trustworthy assistance [1]. Furthermore, the weaknesses identified in the study [4] suggest opportunities for future research to expand sample sizes, incorporate multiple LLMs, and integrate user-profile data, which can enhance the effectiveness

and reliability of AI-driven assistants in supporting academic query management.

By considering these limitations and incorporating the insights from the studies, academic query management systems can be more robust, reliable, and ethical. This, in turn, can improve the academic experience for both students and staff.

## III. METHODOLOGY

This section presents a detailed overview of the proposed Architecture, including effective prompting engineering techniques, key aspects to create an assistant, compatible files for knowledge retrieval, and user interface design. This document has been thoughtfully structured into subsections, each focusing on a specific aspect of the Assistant.

### *A. Overview of the proposed Assistant*

Fig. 1 presents the overview of the proposed Assistant. It consists of different stages to get a final response. We will thoroughly explore each step of creation and implementation, showing the efficacy of LLMs in retrieving information and document management.

- **Front-End:** Displays the answers issued by the Assistant and serves as an intermediary for sending queries from the user.
- **Back-End:** Manages the existence of threads and is responsible for retrieving a response once the request has been processed.
- **External Services:** Processes user queries through the Assistant that works with instructions to send answers.

### *B. External Services*

Practical prompt engineering is essential in developing an intelligent assistant. It aims to furnish precise and comprehensive instructions that enable the artificial intelligence model to produce relevant and reliable responses that meet the user's expectations. A skillfully crafted prompt can significantly enhance the accuracy and thoroughness of the model's answers, ensuring the user's satisfaction. This study employed various techniques to optimize the interactions with the artificial intelligence model. The instructions to be followed by the Assistant to solve any academic query can be seen in Fig.3.

#### *1) Define a goal and audience:*

The goal will determine the structure of the prompt to be designed in the following step and assist in evaluating the quality of the system's response before further iterations. Moreover, there is a clearly defined and described target audience. Defining the prompt in the function of the audience lets us adjust the tone, complexity, detail, and content to be provided by the assistant [13, 14]. An example can be seen in Sections 1 and 4 in Fig.3.

#### *2) Context as a guide:*

It is essential to provide context to help to comprehend the situation. However, more information does not mean better quality of response; it is necessary to omit data that does not contribute significantly to the understanding of the context. Providing a specific role to the Assistant ensures that its responses are aligned with the desired outcome [15, 16]. An example can be seen in Sections 5 and 6 in Fig.3.

#### *3) Clear Instructions:*

Providing clear and descriptive instruction is essential to effectively direct the course of artificial intelligence. This includes specifying whether seeking more creative or precision-focused responses, which will help the AI understand and execute your objectives. Providing a precise and detailed prompt is crucial in generating content more aligned with the unique requirements of a given scenario [15, 16].

#### *4) Evaluate and Adapt:*

Resampling is a continuous process based on the evaluation of the obtained output. "Iteration is beneficial for improving the instruction, making it possible to adjust the limitations or areas where the model presents problems, getting multiples outputs to select the best one [16].

#### *5) Validation of unnecessary prompts:*

It is essential to give concrete instructions to ensure the Assistant stays focused and does not give unnecessary answers, providing accurate and helpful answers, as shown in Sections 2 and 3 in Fig.3. It is also essential to ensure that the files or information provided to the Assistant are related to the query at hand, as this will help consolidate the process and avoid confusion or delays.

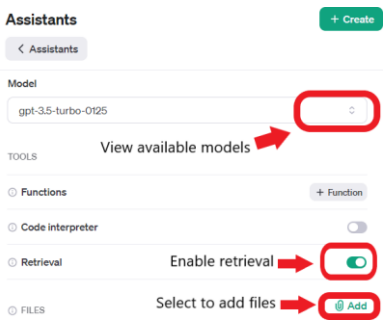



**Fig. 2.** Illustrates a list of available models, how to enable retrieval tool, and how to add files to the Assistant.

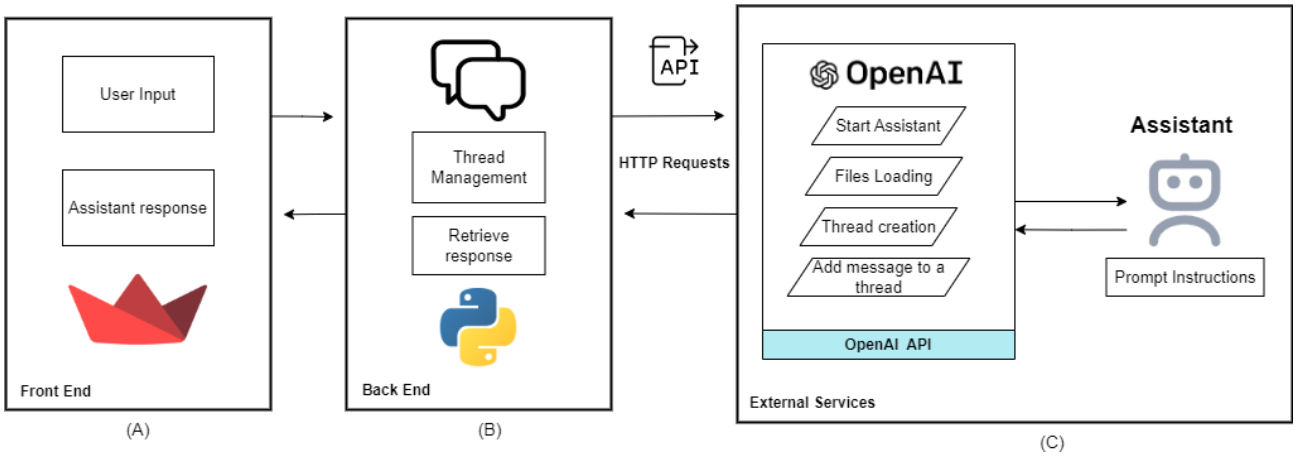



**Fig. 1.** Architectural of Proposed AI-Integrated Assistant: front-end in (A) Box, back-end in (B) Box, and External Services in (C) Box.

*6) Files Loading:*

Files can be uploaded easily from the development platform. These files are necessary to use the "knowledge retrieval" tool, which retrieves specific data from the uploaded files. As shown in Fig.2, the retrieval tool can be enabled through the OpenAI developer platform.

| |
|---|
| **Section A:** |
| **Instructions:** I am your intelligent university assistant, and I am designed to provide you with accurate and complete information based on the uploaded documents. My goal is to efficiently assist you in navigating the academic offerings, costs, and administrative processes of an Academic Institution. |
| **Section B:** |
| Specific Instructions for Service Improvement: |
| **Section C:** |
| Out-of-Context Questions: Whenever the user asks about topics that are not related to university information, uploaded documents, mathematical operations, scientific topics, or practical advice, you should politely respond, "I can't answer those types of questions. Is there another topic I can help you with?" |
| **Section D:** |
| Polite and Cordial Responses: Always maintain a cordial, polite, and helpful tone, reinforcing a positive user experience. Do not give the name of the documents you have uploaded. |
| **Section E:** |
| Handling of documents and responses: Before responding to the user, you should always research all the available documents to offer a satisfactory response. Each document has relevant information that can complement an answer. To answer what careers the university offers, you should list them by their "Area of Knowledge." |
| **Section 6:** |
| Career Costs or Prices: The careers are divided into sections; in each section, you will find the cost of all their semesters and tuition. To give information about the costs of semesters of a specific career, move to the "Knowledge Area." |

**Fig. 3.** The following instructions are segmented into sections to smoothly create and operate the Assistant. Each of these sections has an essential role in understanding every single query.

**Table I.** Available files for Knowledge Retrieval

| Category | Formats |
|---|---|
| Text Documents | .txt, .md, .docx, .html, .json, .py, .rb, .java, .pdf, .tex |
| Presentation Files | .pptx |
| Code Files | .cpp, .c, .php, .js, .css, .ts |

*7) Files Compatibility:*

The effectiveness and quality of assistant responses are directly related to the compatibility of file formats and their content. Therefore, an important aspect to consider is the simplicity and commonality of formats the Assistant can manage to ensure that it comprehends them accurately.

In this aspect, specific attributes in the content of widely used PDF files can affect the Assistant's ability to process retrieved information. Complex elements such as images, diagrams, and tables can be particularly challenging to understand, potentially resulting in unclear or incorrect responses.

These files should be implemented with a clear and straightforward structure to ensure a precise interpretation and a relevant response. This approach is critical to achieving optimal results.

According to data provided by OpenAI [17], supported documents are shown in Table I.

### *C. Back-End Development*

The OpenAI development platform is essential for creating an efficient AI-powered assistant. It features a range of tools to support specific needs. Users can write detailed instructions for smooth task execution. Additionally, the platform presents a selection of models to find the best fit for the system's requirements.

The "Retrieval" tool allows the assistant to retrieve knowledge outside its natural language processing model as user-supplied documents. Once files

**Table II.** LLM Model Prices

| Model | Input | Output |
|---|---|---|
| gpt-3.5-turbo [18] | 0.50 | 1.50 |
| gpt-4-turbo [18] | 10.00 | 30.00 |
| Mistral Large [19] | 8.00 | 24.00 |
| Mistral Small [19] | 2.00 | 6.00 |
| Gemini Pro [20] | 0.125 | 0.375 |
| Claude Instant [21] | 0.80 | 2.40 |
| Claude 2.1 [21] | 8.00 | 24.00 |

**Table III.** Run Steps and Concepts

| Category | Concept |
|---|---|
| in progress | Assistant is analyzing user prompt |
| completed | The request was fulfilled |
| failed | Internal error |
| canceled | The request was canceled expired |
| | The request took too long to respond |

are uploaded, OpenAI will automatically fragment the document, index and store the embeddings, and implement vector search to retrieve relevant information and answer the user's queries[17].

A list of available models can be seen and tested through assistant configuration as seen in Fig. 2.

*1) LLM Pricing and Models:*

Natural Language Processing (NLP) offers many possibilities for LLMs. Various LLMs are available, each with unique features that can be leveraged for NLP tasks. Table II shows a comparative analysis of the pricing of the most widely used LLMs to understand their capabilities relative to their cost.

The table II shows the cost per 1 million tokens for input and output prompts. There is an apparent reason why the "gpt-3.5-turbo" model was chosen, with cost being one of the most significant factors compared to the other available models.". It offers a cost-effective option for natural language processing, and the quality of response is not significantly compromised.

The assistant model was selected based on evaluating implementation costs, especially in academic settings where resources are often limited. The chosen model delivers a satisfactory balance between performance and price.

*2) Threads Management:*

Messages are assigned to a thread representing the conversation session between the Assistant and the user.

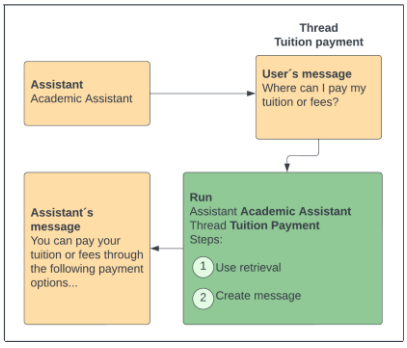



**Fig. 4.** Process from thread creation to retrieve a response.

There is no limit to the number of messages the thread can store [17]. Threads serve as a record of all the messages exchanged during a session, providing important context for future queries within that session. The creation of the thread is indicated in Fig. 4.

*3) Query Processing:*

Once the query is sent, the Assistant processes it using the "gpt-3.5-turbo" model and then enters an execution process representing different response stages. According to the data provided by OpenAI [17], the stages are shown in Table III.

### *D. Front-end Development*

To interact with the OpenAI API through HTTP requests, a unique "API KEY" is required for authorization to access all functions. In light of this, OpenAI offers an application programming interface called the "Assistants API" that allows developers to build powerful intelligent assistants within their applications. The API eliminates the need to manage conversation history and provides access to multiple tools such as "Code Interpreter," "Retrieval," and "Function Calling." [22].

*1) Integration to Streamlit app:*

Integrating OpenAI services for assistant creation requires several parameters to enable interaction between the user and the API. A unique API key and an available assistant identifier, which are declared in the code to authorize access to OpenAI services, are essential. Once these parameters are configured, the API can be called from the application.

*2) User input queries:*

The Assistant presents a simple, easy-to-understand interface, allowing users to perform academic queries efficiently. Streamlit was used for its implementation; this framework is an open-source Python library that facilitates the creation of customized web applications for machine learning and data science[23]. When the initial screen is displayed, the interface has a text field where users can type their queries. A button to send queries is next to the input field, starting the communication process with the OpenAI assistant, as shown in Fig. 5.

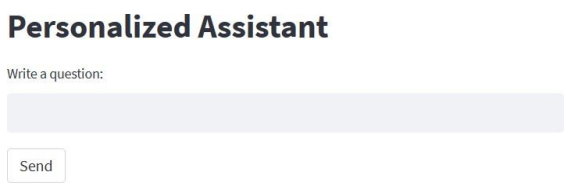



**Fig. 5.** Assistant interface developed through Streamlit

*3) Message Annotations:*

Assistants create messages with annotations embedded in the content array of the object. These annotations provide information on how to annotate the text in the message [17]. There is one type of annotation on this Assistant:

- File citation: File citation annotations refer to a specific quote in a particular file uploaded and used by the Assistant to generate the response.

## IV. RESULTS AND DISCUSSION

The proposed AI-powered Assistant for academic information retrieval is illustrated in Fig. 6, showcasing a historical chat session where it followed a set of features designed to streamline the access and management of academic documents. This system offers robust functionalities to facilitate efficient access to relevant information within document retrieval.

A series of comprehensive assessments were conducted to evaluate the AI-powered Assistant's efficiency in comprehending a broad range of queries related to academic and administrative documents. The results, as presented in Fig. 6, demonstrate that the Assistant can provide exact responses and is closely aligned with the specific context of the questions asked.

### *A. Evaluation of User Experience Using System Usability Scale*

User experience is crucial, which is why the System Usability Scale (SUS) was employed. It is a widely used tool to measure the usability of various systems and products. This scale provides a quick and reliable usability assessment from the user's perspective.[24]

For this project, we used the SUS tool to assess the usability of the system we implemented. The evaluation involved ten individuals from the Virtual Reality department at the Centro de Investigación, Innovación y Transferencia Tecnológica (CIITT) with substantial expertise in immersive technologies. Their feedback provided a comprehensive evaluation of the system's usability. The results are shown in Figure 7.

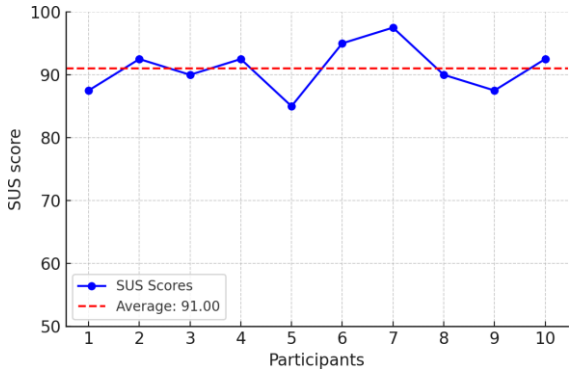



**Fig. 7.** Statistical illustration displays SUS scores from ten participants. The X-axis represents the participants numbered from one to ten, and the Y-axis shows the scores ranging from fifty to one hundred.

The system achieved an impressive average score of 91.00 on the SUS scale for usability and user satisfaction. The high SUS scores from participants indicate an exceptional level of user satisfaction, rapid adoption, and ease of use, equivalent to an A+ grade. This positive feedback strongly suggests that the system has significant potential for adoption and application in academic and research environments.

The integration of AI-powered assistants in academic environments presents numerous ethical considerations that necessitate thorough examination. While this technology has the capacity to transform the organization and availability of academic data, it also introduces substantial issues surrounding bias, privacy, and data accuracy.

### *B. Potential Biases*

One of the significant challenges associated with using AI, particularly LLMs, is the potential existence of inherent biases. These biases can result from the training data used in developing the models. If the training data contains biases, they could manifest in the responses generated by the assistant. For instance, an LLM primarily trained on data in various languages may not offer accurate responses for users. To address these biases, it is crucial to utilize representative datasets during model training and to implement adjustment techniques to the document content to enhance the assistant's comprehension.

### *C. Data Privacy*

Privacy is a major concern when using AI assistants in academic settings. These assistants have access to sensitive student and staff information, and ensuring this data is handled securely and confidentially is crucial. To mitigate these concerns, the assistant's responses are restricted to information found exclusively within academic documents, which must be exhausted and reviewed before being

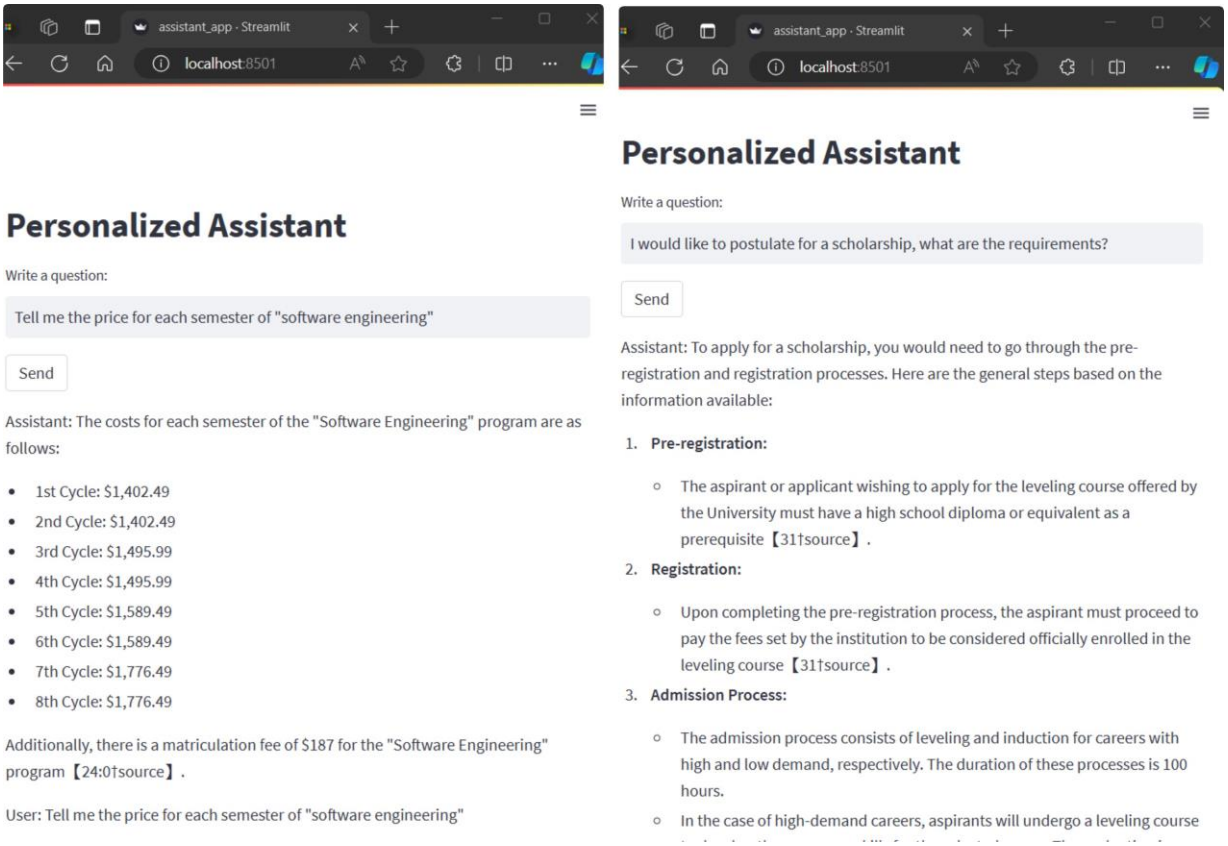



**Fig. 6.** Showing assistant history chat about academic information related to professional careers availability, tuition costs, payment options, and scholarship requirements. Assistant and user inputs are labeled on different fields

loaded to the assistant using sophisticated, prompt engineering techniques.

### *D. Disinformation and Information Accuracy*

The accuracy of the information provided by AI assistants is crucial to their effectiveness and reliability. LLMs, while powerful, are not infallible and can generate incorrect or misleading answers. This situation can be particularly problematic in academic contexts where information accuracy is vital. Implementing information verification and validation mechanisms is fundamental. Assistants from OpenAI are capable of citing verifiable sources. In addition, supervised learning techniques and continuous feedback can be used to improve the accuracy and reliability of models.

The comprehensive assessments' results underscore the efficacy and potential impact of AI-powered assistants in facilitating efficient access to academic information. Continuous refinement and innovation are vital to maximizing utility and addressing evolving user needs in academic environments.

### *E. Code Availability*

The code generated can be found in https://github.com/xr-lab-ucacue/Academic-Assistant.git

## V. CONCLUSION AND FUTURE WORK

The advancement of new advanced AI-powered assistants presents a potential shift in the landscape of academic support services, potentially replacing certain human tasks. However, this transition also opens up new opportunities for employment and innovation, emphasizing the dynamic nature of technological advancement in academic environments.

After performing this research on an AI-powered Assistant for academic information using a knowledge retrieval tool from OpenAI, it is clear that integrating NLP technologies can greatly enhance the accessibility and management of academic documents. The results show that this technology has the potential to revolutionize the way we interact with academic information.

It is necessary to explore and compare multiple LLMs that are publicly available and restricted to evaluate the scalability of the proposed methodology.

Prominent models include OpenAI's GPT-4, open models, and other closed LLMs such as Claude. This comparison will not only allow us to demonstrate the robustness of our methodology. However, it will also help to identify the most suitable models for various academic tasks based on their performance and cost.

The comparative approach will focus on implementing the same prompting across different LLMs, which will allow a direct assessment of each model's ability to handle specific academic tasks. This method will facilitate the identification of the relative strengths and weaknesses of each LLM, providing a solid basis for selecting the most efficient and effective model for academic applications.

In addition, the comparative analysis will include a thorough evaluation of performance factors, such as response accuracy, consistency in content generation, and the ability to handle complex and contextually rich queries. The costs of using each LLM, regarding computational resources required and licensing and model access, will also be considered.

Despite its implications for designing better LLM-powered Assistants, this Assistant has some limitations that point to future research directions. First, due to the Assistant API latency issues, because it is still in beta version, the integration of LLMs into a Personalized Assistant resulted in errors that would ideally be avoidable. Future works should explore alternative LLMs and their performance in solving user queries.

## CONFLICT OF INTEREST

The authors have declared that no competing interests exist.

## ACKNOWLEDGEMENTS

The XR LAB department in CIITT (Centro de Investigacion, Innovación y Tranferencia Tecnológica) fully supports this work, providing the necessary equipment to develop it.

## REFERENCES

[1] M. R. King, "The future of ai in medicine: a perspective from a chatbot," *Annals of Biomedical Engineering*, vol. 51, no. 2, pp. 291–295, 2023.

[2] A. Pham Duy and H. Le Thanh, "A question- answering system for vietnamese public administra- tive services," in *Proceedings of the 12th Interna- tional Symposium on Information and Communica- tion Technology*, 2023, pp. 85–92.

[3] D. Mhlanga, "Open ai in education, the responsible and ethical use of chatgpt towards lifelong learning," *Education, the Responsible and Ethical Use of ChatGPT Towards Lifelong Learning (February 11, 2023)*, 2023.

[4] H. Abu-Rasheed, M. H. Abdulsalam, C. Weber, and M. Fathi, "Supporting student decisions on learning recommendations: An llm-based chatbot with knowledge graph contextualization for conversational explainability and mentoring," *arXiv preprint arXiv:2401.08517*, 2024.

[5] J. Prather, P. Denny, J. Leinonen, B. A. Becker, I. Albluwi, M. E. Caspersen, M. Craig, H. Keuning, N. Kiesler, T. Kohn, A. Luxton-Reilly, S. MacNeil, A. Petersen, R. Pettit, B. N. Reeves, and J. Savelka, "Transformed by transformers: Navigating the ai coding revolution for computing education: An iticse working group conducted by humans," in *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 2*, ser. ITiCSE 2023. New York, NY, USA: Association for Computing Machinery, 2023, p. 561–562. [Online]. Available: https://doi.org/10.1145/3587103.3594206

[6] M. Perkins, "Academic integrity considerations of ai large language models in the post-pandemic era: Chatgpt and beyond," *Journal of University Teaching & Learning Practice*, vol. 20, no. 2, p. 07, 2023.

[7] "Introducing chatgpt," Mar. 2024, [Online; accessed 4. Mar. 2024]. [Online]. Available: https://openai.com/blog/chatgpt

[8] R. Tang, Y.-N. Chuang, and X. Hu, "The science of detecting llm-generated texts," 2023.

[9] V. D. Kirova, C. S. Ku, J. R. Laracy, and T. J. Marlowe, "Software engineering education must adapt and evolve for an llm environment," in *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, 2024, pp. 666–672.

[10] S. Quevedo, F. Merchán, R. Rivadeneira, and F. X. Dominguez, "Evaluating apache openwhisk-faas," in *2019 IEEE fourth ecuador technical chapters meeting (ETCM)*. IEEE, 2019, pp. 1–5.

[11] S. Quevedo, F. Domíngez, and E. Pelaez, "Detecting multi thoracic diseases in chest x-ray images using deep learning techniques," in *2023 IEEE 13th International Conference on Pattern Recognition Systems (ICPRS)*. IEEE, 2023, pp. 1–7.

[12] S. Pati, U. Baid, B. Edwards, M. Sheller, S.-H. Wang, G. A. Reina, P. Foley, A. Gruzdev, D. Karkada, C. Davatzikos *et al.*, "Federated learning enables big data for rare cancer boundary detection," *Nature communications*, vol. 13, no. 1, p. 7346, 2022.

[13] L. J. Jacobsen and K. E. Weber, "The promises and pitfalls of chatgpt as a feedback provider in higher education: An exploratory study of prompt engineering and the quality of ai-driven feedback," *OSF*, 2023.

[14] J. D. Velásquez-Henao, C. J. Franco-Cardona, and L. Cadavid-Higuita, "Prompt engineering: a methodology for optimizing interactions with ai- language models in the field of engineering," *Dyna*, vol. 90, no. 230, pp. 9–17, 2023.

[15] S. Ekin, "Prompt engineering for chatgpt: A quick guide to techniques, tips, and best practices," *Authorea Preprints*, Oct. 2023.

[16] B. Chen, Z. Zhang, N. Langrené, and S. Zhu, "Unleashing the potential of prompt engineering in large language models: a comprehensive review," *arXiv preprint arXiv:2310.14735*, 2023.

[17] "Openai platform," Mar. 2024, [Online; accessed 4. Mar. 2024]. [Online]. Available: https://platform.openai.com/docs/overview

[18] "Pricing," Mar. 2024, [Online; accessed 4. Mar. 2024]. [Online]. Available: https://openai.com/pricing

[19] "Pricing and rate limits | Mistral AI Large Language Models," Mar. 2024, [Online; accessed 4. Mar. 2024]. [Online]. Available: https://docs.mistral.ai/platform/pricing

[20] "Precios | IA generativa en Vertex AI | Google Cloud," Mar. 2024, [Online; accessed 4. Mar. 2024]. [Online]. Available: https://cloud.google. com/vertex-ai/generative-ai/pricing?hl=es

[21] "Claude API," Mar. 2024, [Online; accessed 4. Mar. 2024]. [Online]. Available: https://www.anthropic.com/api

[22] "Assistants API | OpenAI Help Center," Mar. 2024, [Online; accessed 5. Mar. 2024]. [Online]. Available: https://help.openai.com/en/articles/ 8550641-assistants-api

[23] "Streamlit Docs," Mar. 2024, [Online; accessed 5. Mar. 2024]. [Online]. Available: https://docs.streamlit.io

[24] J. Brooke *et al.*, "Sus-a quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.