



UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

ESCUELA DE CIENCIAS MATEMÁTICAS Y
COMPUTACIONALES

Integrating a LLaMA-based Chatbot with Augmented Retrieval Generation as a Complementary Educational Tool for High School and College Students

Trabajo de integración curricular presentada como requisito para la
obtención del título de Ingeniero en Tecnologías de la Información

Autor:

Darío Sebastián Cabezas Erazo

Tutor:

Manuel Eugenio Morocho Cayamcela, Ph.D.

Cotutor:

Rigoberto Salomón Fonseca Delgado, Ph.D.

Urcuquí, Diciembre de 2024

Autoría

Yo, **DARÍO SEBASTIÁN CABEZAS ERAZO**, con cédula de identidad 0402019749, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así como, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el/la autor/a del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urququí, Diciembre 2024.



Darío Sebastián Cabezas Erazo

CI: 0402019749

Autorización de publicación

Yo, **DARÍO SEBASTIÁN CABEZAS ERAZO**, con cédula de identidad 0402019749, cedo a la Universidad de Investigación de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación

Urcuquí, Diciembre 2024.



Darío Sebastián Cabezas Erazo

CI: 0402019749

Dedication

This research is dedicated to my father, whose constant support has been an example to follow in my life. His guidance, free of judgment, and encouragement at all times has shaped the way I am and my path. To my mother, who has always made sure there was a hot meal on the table and worked to make sure I never lacked for anything. To my brother, who has been an example of perseverance and resilience, showing me how to always keep going despite any adversity. To my grandparents and family in Ibarra, who have offered me a place to go, a listening ear, and constant support, both emotional and material. And to my girlfriend, Joseline, who showed up when I needed support the most and has helped me be a better person every day with her wise counsel and loving care.

Darío Sebastián Cabezas Erazo

Acknowledgment

I would like to express my deepest gratitude to my advisor, Professor Eugenio Morocho, who was the first to believe in my potential and capabilities, guiding me in the development of this thesis. I am equally grateful to Professor Rigoberto Fonseca, who provided me with guidance when I had doubts about my future at the university. My gratitude also extends to all the professors from the different departments who have supported me throughout different courses and research work, pushing me forward.

I would like to extend my gratitude to the Universidad Internacional del Ecuador, and specifically to Paulina Vizcaíno and Iván Reyes, who provided invaluable support during the writing of this work. I also wish to express my thanks for the funding received for the publication of this research at ICSOFT 2024.

Special thanks go to Jhoao Minango, my first and oldest friend at the university. His emotional support, whether it was sharing a meal or talking about our lives over a drink, has been a constant source of strength throughout these years.

Finally, I want to express my heartfelt thanks to my girlfriend, Joseline. Her presence has been a beacon of support when I needed it most. Her wise counsel and loving care have not only encouraged me, but have also played an important role in my personal growth. Joseline, your unwavering love and support have been instrumental in my journey, and I am deeply grateful for all you have done.

Darío Sebastián Cabezas Erazo

Resumen

En el panorama educativo actual, la transición de paradigmas tradicionales a experiencias de aprendizaje más interactivas y personalizadas se ha visto acelerada por los avances tecnológicos, particularmente en inteligencia artificial. Esta investigación presenta el desarrollo de un chatbot basado en el modelo largo de lenguaje (LLM) llamado LLaMA 7B chat desarrollado por Meta, que ha sido cuantizado para optimizar el uso de recursos, mejorado mediante técnicas de generación aumentada por recuperación con el fin de llevar a cabo respuestas en una asignatura centrada predominantemente en habilidades de resolución de problemas como son las matemáticas. Nuestra investigación ha dado como resultado un sistema de recuperación de documentos utilizando Pinecone como vectorstore, cuyos documentos almacenados fueron creados a partir del contenido de las páginas, que denominamos metadatos, del aclamado libro *Precálculo: Matemáticas para el Cálculo* que se procesaron utilizando el modelo all-MiniLM-L6-v2 y empleando la similitud coseno como métrica para realizar la búsqueda semántica. A continuación, el sistema de recuperación de documentos se une al modelo cuantizado utilizando el framework especializado en LLM denominado Langchain. De esta forma hemos obtenido un chatbot que siguiendo un prompt de sistema, el contexto de los documentos recuperados de Pinecone y la pregunta del usuario puede responder con ejercicios o material audiovisual de apoyo a preguntas de matemáticas para estudiantes de secundaria o cursos de nivel universitario.

Palabras clave: Grandes modelos de lenguaje, LLaMA 7B chat, Cuantización, Generación Aumentada por Recuperación, Base de Datos Vectorial, Pinecone, Langchain

Abstract

In the current educational landscape, the transition from traditional paradigms to more interactive and personalized learning experiences has been accelerated by technological advancements, particularly in artificial intelligence. This research presents the development of a chatbot based on the Large Language Model (LLM) called LLaMA 7B chat developed by Meta, which has been quantized to optimize the use of resources, enhanced using retrieval augmented generation techniques in order to carry out responses in a subject that is predominantly focused on problem solving skills such as mathematics. This research has resulted in a document retrieval system using Pinecone as our vectorstore whose stored documents were created from the page content, called metadata, of the acclaimed book *Precálculo: Matemáticas para el Cálculo* which were processed using the all-MiniLM-L6-v2 model and using cosine similarity as a metric to perform the semantic search. Document retrieval system is then linked to the quantized model using the specialized LLM framework called Langchain. In this way we have obtained a chatbot that following a system prompt, context of documents retrieved from Pinecone and the user's question can respond with exercises or audiovisual support material for mathematics questions for high school students or university level courses.

Keywords: Large Language Model , LLaMA 7B chat, Quantization, Retrieval Augmented Generation, Vector Database, Pinecone, Langchain

Contents

Dedication	iii
Acknowledgment	iv
Resumen	v
Abstract	vi
Contents	vii
List of Tables	x
List of Figures	xi
1 Introduction	1
1.1 Background	1
1.2 Problem statement	2
1.3 Objectives	3
1.3.1 General Objective	3
1.3.2 Specific Objectives	3
2 Theoretical Framework	4
2.1 Natural Language Processing: A Key Subfield of Artificial Intelligence . . .	4
2.1.1 Natural Language Processing	4
2.1.2 Transformer	4
2.2 Large Language Models	6
2.2.1 Large Language Model Major Families	7
2.2.1.1 Generative Pre-trained Transformers Family	7

2.2.1.2	Pathways Language Model Family	8
2.2.1.3	Large Language Model Meta AI Family	9
2.2.2	Limitations of LLM	10
2.2.2.1	Hallucinations	10
2.2.3	Improving Performance in LLM	11
2.2.3.1	Prompting	12
2.2.3.2	Fine-Tuning	13
2.2.3.3	RAG	13
3	State of the Art	15
3.1	Literature Review on LLMs in Education	15
3.2	Chatbots in Education	15
3.2.1	GPTs: OpenAI Custom Chatbot	16
3.2.2	MathGPT	16
3.2.3	NewtBot: LLM-as-tutor for Physics	16
3.2.4	Chatting Over Course Material	17
3.2.5	Knowledge Navigator	17
3.2.6	Large Language Model Intelligent Agent Tutoring in Higher Educa- tion Courses using a RAG Approach	17
3.2.7	Large language model-powered chatbots for internationalizing stu- dent support in higher education	18
4	Methodology	19
4.1	Methodology	19
4.1.1	Development of a Vector Database with Pinecone	19
4.1.2	Math Database Creation	20
4.1.3	Uploading embedding and metadata to Pinecone	21
4.2	Quantization and Optimization of LLM model	22
4.2.1	Download Model Weights	22
4.2.2	Local Deployment and Quantization	22
4.2.3	Large Language Model and LangChain	23
4.2.3.1	Quantized Model	24

4.2.3.2	Quantized Model and System prompt	24
4.2.3.3	Quantized Model, System prompt and RAG	25
4.3	Comparative Analysis of Model Configurations and Retrieval-Augmented Generation Techniques	25
4.3.1	Document Retrieval Process	26
4.3.2	Model Configuration Analysis	26
5	Results and Discussion	27
5.1	Vector Database in Pinecone	27
5.1.1	Math Database	27
5.1.2	Document Retrieval Process	28
5.2	Model Configuration Results	30
5.2.1	Quantized Model	30
5.2.2	Quantized Model and System Prompt	31
5.2.3	Quantized Model, System Prompt, and RAG	33
5.2.3.1	RAG Evaluation	34
6	Conclusions and Future Work	36
6.1	Limitations	37
6.2	Future Works	37
	Bibliography	38

List of Tables

2.1	LLM principal families.	7
2.2	Three types of LLM hallucinations	11
4.1	Structure of the mathematics database’s metadata	21
4.2	Quantization options available for the LLaMA-2-7B model.	23
5.1	Vector properties considered for different similarity metrics.	27
5.2	Top k elements associated with each query vector	29
5.3	Queries used in quantized LLaMA 7B chat model using $q4_0$ quantization .	31
5.4	Queries and responses of model with an integrated system prompt	32
5.5	Comparison of responses using a system prompt to provide guidelines for a quantized model.	34
5.6	Details of the response, including total character count, number of context characters generated, and the percentage of context characters.	35

List of Figures

2.1	Transformer architecture	5
2.2	Milestones in language models	6
2.3	Overview of the different generations of GPTs developed by OpenAI.	8
2.4	Overview of the different versions of PaLM family developed by Google	9
2.5	Overview of the different variants of the LLaMA family	9
2.6	Retrieval Augmented Generation process	14
4.1	Metadata elements for our vectorial database	20
4.2	Projection of math vectorial database using principal components analysis method	22
4.3	LLMChain using system prompt Workflow.	24
4.4	Educational chatbot workflow	25
5.1	Comparison of similarity metrics used in the Pinecone system.	28
5.2	Document Retrieval Process	29

Chapter 1

Introduction

1.1 Background

In the era of Education 1.0, students primarily obtained information from books and web pages, yet these traditional mediums needed to have the essential element of interactivity. The accurate perspective offered by [1], on the 3 Rs—Receiving, Responding, and Regurgitating—many educational institutions remain entrenched in this outdated paradigm. Education 1.0 is firmly grounded in the passive acquisition of knowledge from instructors to students [1, 2, 3]. Consequently, education 2.0 marked a crucial turning point in the evolution of the educational process, as underscored by Huk [4].

Education in the “online world” provided the opportunity for users to comment and interact with content [1, 4, 2, 3], reflecting a significant shift toward a more engaging and participatory learning environment. Hence, education 3.0 revolves around personalized learning tailored to students’ interests, fostering innovation and creativity [4, 2, 3]. In this paradigm, the role of the teacher has transformed, evolving into that of an organizer facilitating an educational environment where students collaboratively cultivate their knowledge [5, 2]. In this way, we are moving from a passive to an active learning process through technology and the internet.

Education 4.0 marks the forefront of technology integration in learning. Central to Education 4.0 is the ongoing enhancement of artificial intelligence (AI) [4, 3], which catalyzes driving educational transformation [3]. This transformation encompasses personalized and adaptive learning, intelligent learning assistants, predictive analytics for early intervention,

integration of augmented reality and virtual reality, gamification and simulation, lifelong learning and skill development, and the ethical considerations of AI in fostering inclusive education [3].

Natural language processing (NLP), a subset of AI, empowers machines to comprehend, interpret, and generate human language, offering numerous educational advantages, especially in facilitating personalized language-learning experiences [6]. The ascendancy of deep learning in NLP [7], the availability of vast public datasets [8], and the capabilities of powerful computing devices [9] to handle substantial data through sophisticated algorithms have catalyzed the emergence of large language models, known as LLM.

LLM such as large language model Meta AI [10], known as LLaMA, developed by Meta, generative pre-trained transformer (GPT) [11] in its version 3.5 developed by OpenAI, bidirectional encoder representations from Transformers [12], among others. LLM are renowned for their proficiency in capturing semantic relationships between words and phrases [13]; these models, alongside chatbots, have become a formidable force, ushering in an unprecedented revolution in education [6, 14]. However, LLMs face critical issues such as hallucinations and knowledge obsolescence; at the same time, retrieval-augmented generation (RAG) has emerged as an innovative approach to overcoming these limitations by integrating knowledge from external databases [15].

1.2 Problem statement

Despite the long-standing integration of information technology in education, the past four decades have shown the most significant potential for transformative change [16]. Simultaneously, an educational technology industry has emerged, dedicating its endeavors to the development and dissemination of educational content, learning management systems, language applications, as well as augmented and virtual reality tools, personalized tutoring, and assessment platforms [16, 3]. However, this transformation varies widely based on socioeconomic and educational contexts [17, 16]. At the same time, in Latin America and the Caribbean, the challenges persist. According to the Programme for International Student Assessment [17] (PISA) 2022 results, the average rate of low performance in the region was 75% in mathematics, 55% in reading, and 57% in science. According to [18],

chatbots are utilized across various learning domains, including science, language, and mathematics. However, there needs to be more research on their application in mathematics, where courses predominantly focus on calculus and problem-solving skills [18]. It makes the implementation of chatbots in mathematics more challenging, requiring advanced AI technologies for their development [18, 19].

1.3 Objectives

1.3.1 General Objective

Design, develop and evaluate a mathematics-specific educational chatbot using an LLM and RAG techniques using a vector database to enhance personalized learning and problem solving in mathematics education.

1.3.2 Specific Objectives

- Develop an educational chatbot using the LLaMA 2 7B chat model.
- Perform a quantization process to optimize model size and random access memory requirements.
- Integrate a RAG process using a vectorstore to perform semantic search.
- Analyze and compare chatbot text generation with and without RAG.

Chapter 2

Theoretical Framework

2.1 Natural Language Processing: A Key Subfield of Artificial Intelligence

AI has undergone rapid advancements. Progress in theoretical research, coupled with significant improvements in computational power, has paved the way for developing more sophisticated AI models [20]. AI spans sub-fields such as computer vision, speech recognition, robotics, machine learning, and natural language processing.

2.1.1 Natural Language Processing

NLP explores how computers can be harnessed to process and understand human language to execute a range of useful tasks [21, 22, 23, 24]. In this way, NLP involves designing and implementing models, systems, and algorithms to solve practical problems in understanding human languages [21]. Typical NLP applications span spoken language understanding, sentiment analysis, machine translation, natural language generation, natural language summarization, and question answering, among others [21, 22, 25, 24].

2.1.2 Transformer

In 2017, the transformer architecture, shown in Fig. 2.1, revolutionized NLP, particularly for LLMs [26]. This groundbreaking innovation was introduced in the paper “Attention is All You Need” by Vaswani et al. [27].

The original transformer is an encoder-decoder model that uniquely relies entirely on

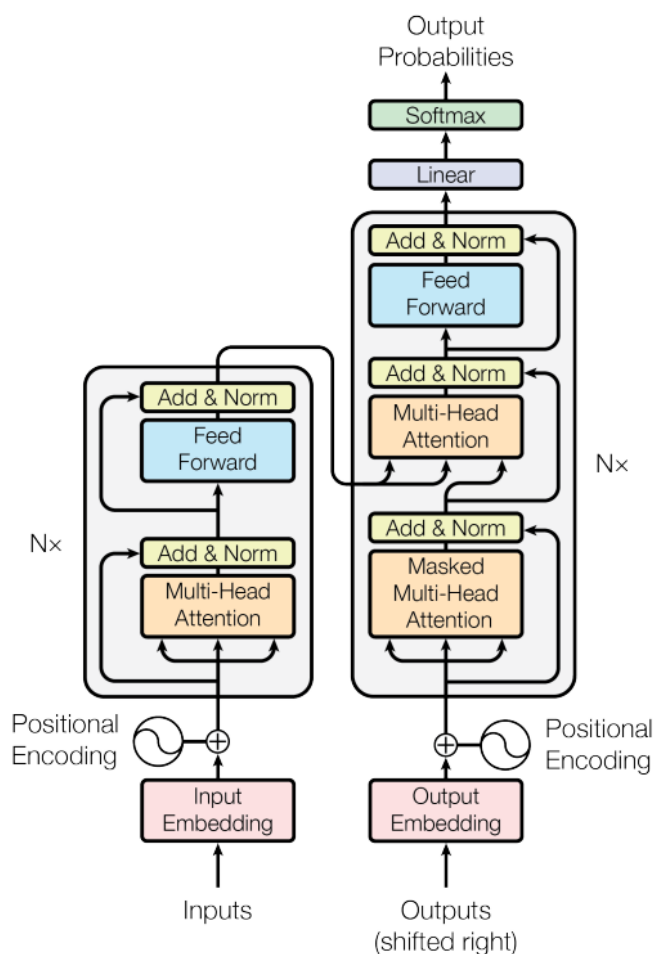


Figure 2.1: The transformer architecture proposed in “Attention is All You Need” [27], highlighting the use of self-attention mechanisms.

self-attention mechanisms [28], unlike previous models that were primarily based on long short term memory [29, 30] and other recurrent neural networks [29, 31] variations. This shift significantly impacted NLP research and development, as transformers excel at capturing long-range dependencies within language. Furthermore, their parallelization ability across multiple graphic processing units (GPUs) allows the training of significantly larger models [32, 26].

At the heart of the transformer is the self-attention mechanism, which captures long-term contextual information more effectively using GPUs compared to recurrence and convolutional mechanisms [26, 28]. In this mechanism, the attention function maps a query and a set of key-value pairs to output, with the query, keys, values, and output all represented as vectors [28, 26].

2.2 Large Language Models

Language models (LMs) have revolutionized natural language processing, demonstrating impressive performance across a wide range of NLP tasks [33, 34]. These models have evolved through several generations and have been marked by significant advancements in research shown in Fig 2.2. The stages of development include statistical LMs (1990s) [33, 29, 35, 33, 36, 26], neural LMs (2013) [33, 35, 33, 36, 26], pre-trained LMs (2018) [29, 35, 33, 36, 26], and LLMs (2020s to present) [29, 35, 33, 36, 26].

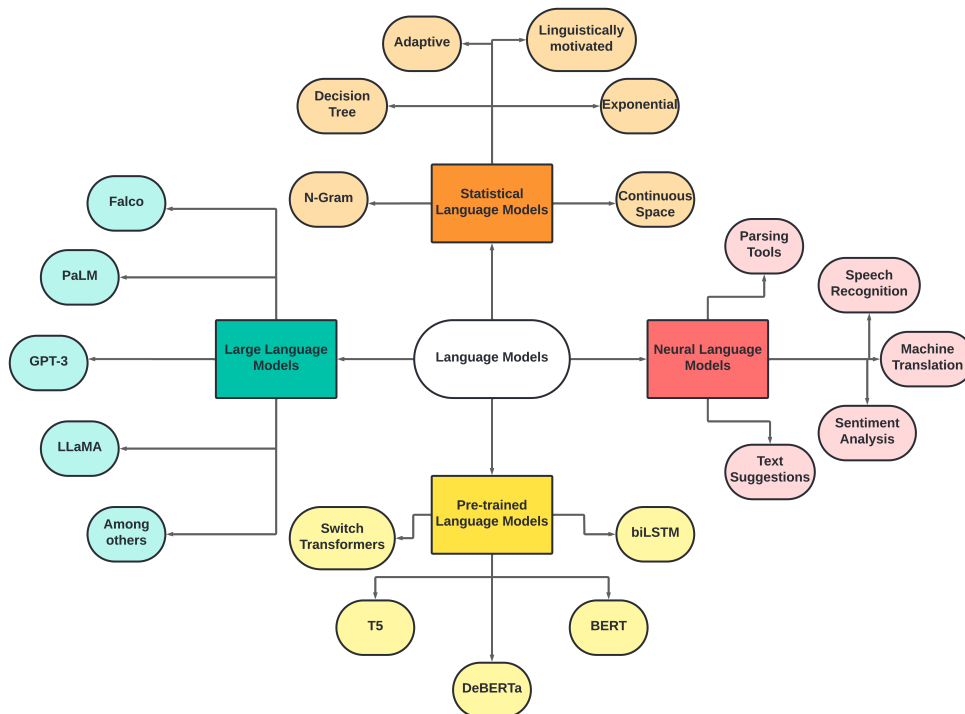


Figure 2.2: Major milestones in language model development over the years [33], illustrating key model advancements and breakthroughs.

Transformer-based LLM like GPT-3 [37], pathways language model (PaLM) [38], and LLaMA [10, 39], leverage extensive parameterization and vast training datasets [29, 40]. These models exhibit robust capabilities in natural language understanding and perform complex tasks such as text generation [36, 26]. Originating from the transformer architecture [27], LLMs have expanded significantly in model size, data volume, and computational requirements [33, 12]. Research indicates that when trained on diverse and extensive datasets, LLMs can effectively generalize across domains due to their immense capacity [26, 37, 41, 38].

2.2.1 Large Language Model Major Families

LLM exhibit superior language understanding and generation capabilities and emergent abilities absent in smaller-scale models [26]. According to Minaee et al., [26], there are three major families of transformer-based LLMs: the GPT family, the PaLM family, and the LLaMA family. Table 2.1 summarizes the most important models and characteristics of these families.

Type	Model Name	#Parameters	Release	Base Models	Open Source
GPT Family	GPT-3	125M, 350M, 760M, 1.3B, 2.7B, 6.7B, 13B, 175B	2020		X
	CODEX	12B	2021	GPT	✓
	WebGPT	760M, 13B, 175B	2021	GPT-3	X
	GPT-4	1.76T	2023	-	X
LLaMA Family	LLaMA 1	7B, 13B, 33B, 65B	2023	-	✓
	LLaMA 2	7B, 13B, 34B, 70B	2023	-	✓
	Alpaca	7B	2023	LLaMA1	✓
	Vicuna-13B	13B	2023	LLaMA1	✓
	Koala	13B	2023	LLaMA1	✓
	Mistral-7B	7.3B	2023		✓
	Code LLaMA	34B	2023	LLaMA2	✓
	LongLLaMA	3B, 7B	2023	OpenLLaMA	✓
	LLaMA-Pro-8B	8.3B	2024	LLaMA2-7B	✓
TinyLLaMA-1.1B	1.1B	2024	LLaMA1.1B	✓	
PaLM Family	PaLM	8B, 62B, 540B	2022	-	X
	U-PaLM	8B, 62B, 540B	2022	-	X
	PaLM-2	340B	2023	-	✓
	Med-PaLM	540B	2022	PaLM	X
	Med-PaLM 2	-	2023	PaLM 2	X

Table 2.1: A summary of the principal families of LLM, detailing their types, model names, parameter counts, release years, and open source status [26].

2.2.1.1 Generative Pre-trained Transformers Family

The GPT family, illustrated in Fig. 2.3 and developed by OpenAI, consists of language models based on decoder-only transformers. GPT-1 [42] was the first to demonstrate that pre-training a decoder-only transformer model can achieve strong performance across various natural language processing tasks [26].

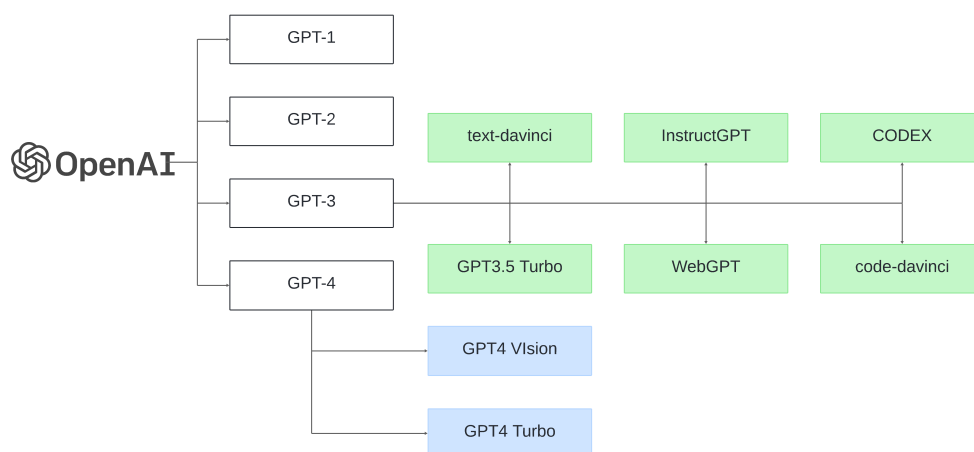


Figure 2.3: Overview of the different generations of GPTs developed by OpenAI.

GPT-2 [41] showed that language models can learn to perform specific natural language tasks without any explicit supervision when trained on a large dataset [26]. GPT-3 [37], with 175 billion parameters, is a pre-trained autoregressive language model. GPT-3 is widely considered the first LLM and is also known for demonstrating emergent abilities for the first time that were not observed in previous pre-trained language models [26].

The launch of the chat generative pre-trained transformer (ChatGPT) [43] on November 30, 2022, marks a significant milestone in the development of LLMs. Powered by GPT-3.5 (later upgraded to GPT-4), ChatGPT is a versatile chatbot that allows users to engage in conversations for various tasks, including answering questions, seeking information, summarizing text, and more [26].

GPT-4 [11] is the latest and most powerful LLM in the GPT family. Launched in March 2023, GPT-4 is a multimodal LLM capable of processing image and text inputs to generate text outputs. Table 2.1 provides an overview of various variants of GPT models and their characteristics.

2.2.1.2 Pathways Language Model Family

PaLM, shown in Fig 2.4, is a family of LLMs developed by Google. The first PaLM model was announced in April 2022 and remained private until March 2023 [38]. Pre-trained on a diverse range of natural language tasks and use cases, PaLM demonstrates the continued benefits of scaling, achieving state-of-the-art few-shot learning results on hundreds of language understanding and generation benchmarks [38, 26].

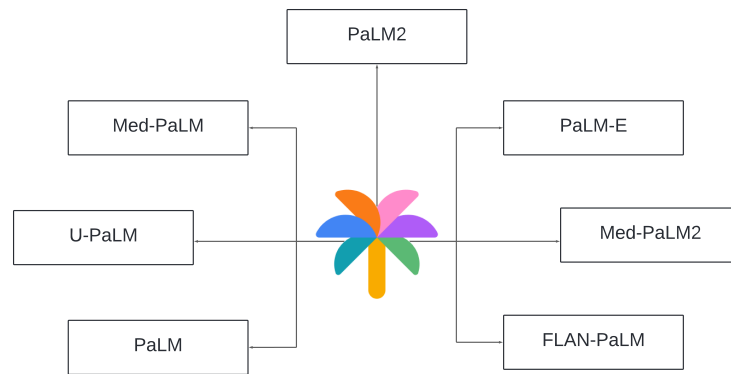


Figure 2.4: Overview of the different versions of PaLM family developed by Google, including several domain-specific adaptations.

PaLM-2 [44] is a more compute-efficient LLM with enhanced multilingual and reasoning capabilities compared to its predecessor, PaLM [38]. PaLM-2 significantly improves model performance on downstream tasks across different model sizes while exhibiting faster and more efficient inference than its predecessor [26].

2.2.1.3 Large Language Model Meta AI Family

The LLaMA model collection [10], shown in Fig 2.5, introduced by Meta AI, includes models ranging from 7 billion to 65 billion parameters, making it a focal point of academic and industrial interest [36, 26]. Unlike GPT models, LLaMA models are open-source, with model weights released to the research community under a non-commercial license [26].

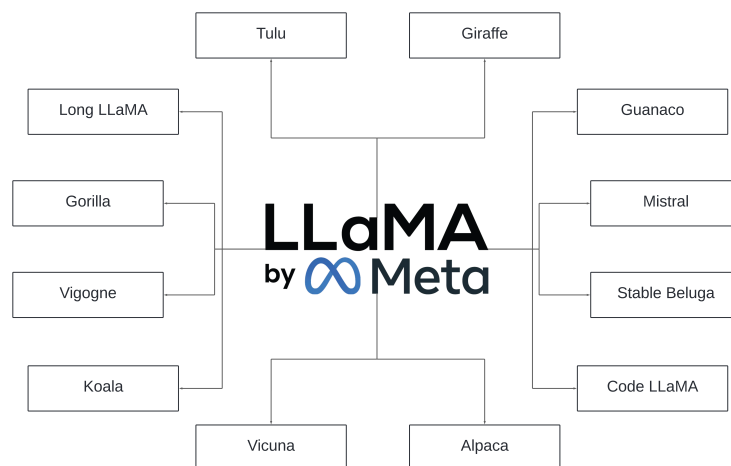


Figure 2.5: Overview of the LLaMA family, including versions 1 and 2. This figure illustrates various model variations achieved through fine-tuning for specific domains, such as the Code LLaMA model, which is specialized for code-writing assistance.

LLaMA models are pre-trained on trillions of tokens collected from publicly available datasets. They use the transformer architecture similar to GPT-3, with several minor modifications, including the SwiGLU activation function instead of ReLU, rotary positional embeddings instead of absolute positional embeddings, and root-mean-squared layer normalization instead of standard layer normalization [45, 26].

In July 2023, Meta and Microsoft collaborated to release the LLaMA-2 collection [39], which features foundational and fine-tuned chat models, such as LLaMA-2 Chat. The LLaMA family is rapidly expanding, with numerous instruction models built on LLaMA or LLaMA-2, including Code LLaMA [46], Gorilla [47], Giraffe [48], Vigogne [49], Tulu 65B [50], Long LLaMA [51], Alpaca [52], Vicuna [53], Guanaco [54], Koala [55], and Mistral-7B [56], among others.

2.2.2 Limitations of LLM

LLMs are designed to predict tokens, as described by Minaee et al. (2024) [26]. This process involves generating the next token in a sequence based on the context provided by preceding tokens. Despite their impressive capabilities, LLMs come with several significant limitations that must be carefully considered.

2.2.2.1 Hallucinations

While LLMs have shown remarkable performance, practical applications reveal significant challenges, notably hallucination [57]. Hallucination, defined as generating nonsensical or deviant content, has gained attention in AI [58, 26]. LLMs lack a truth concept and may produce logically coherent but factually incorrect responses due to training on diverse datasets that include unreliable sources [26].

According to Zhang [57], hallucinations in language models can be categorized into three types: Input-Conflict Hallucination, where LLMs generate content that deviates from user-provided input; Context-Conflict Hallucination, where generated content contradicts earlier outputs from the same model; and Fact-Conflict Hallucination, where generated content These categories highlight instances where LLMs produce inaccurate or inconsistent outputs despite their training and capabilities. Specific examples for each type of hallucination can be found in Table 2.2 and are detailed in [57].

Type	Turn	Example
Input-conflicting Hallucination	<i>User</i>	Please summarize the following content: My friend Hill and I love playing basketball together. We spend hours on the court, dribbling, shooting, and competing against each other. Our passion for this game brings us closer as good friends.
	<i>Model</i>	Lucas and I love playing basketball, and this makes us good friends.
Context-conflicting Hallucination	<i>User</i>	Tell me something about the NBA Commissioner.
	<i>Model</i>	In the world’s most popular basketball league, the NBA, Commissioner Adam Silver has long been regarded as a leader and decision-maker. Firstly, Silver has claimed his dedication to maintaining fairness and competitiveness in the game. Secondly, Silver has repeatedly emphasized his concern for player health and well-being. In this case, Stern ’s words and actions clash with his stated goal of caring for player welfare.
Fact-conflicting Hallucination	<i>User</i>	Who was the mother of Afonso II, the third king of Portugal?
	<i>Model</i>	The mother of Afonso II was Queen Urraca of Castile .

Table 2.2: Illustrations of the three types of LLM hallucinations: **Input-conflicting hallucination** occurs when the LLM confuses the person’s name (**Hill** \Rightarrow **Lucas**) during summarization. **Context-conflicting hallucination** happens when the LLM initially discusses **Silver** but later switches to **Stern**, leading to a contradiction. **Fact-conflicting hallucination** is depicted by the LLM incorrectly stating the mother of Afonso II as **Queen Urraca of Castile**. In contrast, the correct answer is **Dulce Berenguer of Barcelona**. [57].

Finally, several studies have demonstrated that traditional metrics used to assess writing quality often fail to adequately measure the degree of hallucination [59]. However, various methods have been proposed to measure hallucination, including statistical metrics, human evaluation, and model-based metrics such as information-extraction-based metrics, question-answering-based metrics, natural language inference metrics, faithfulness classification metrics, and language model metrics [58].

2.2.3 Improving Performance in LLM

LLMs have demonstrated the ability to acquire substantial knowledge from data without the need for external memory [60]. However, they face challenges in extending or revising their memory, providing clear insights into their predictions, and sometimes generating “hallucinations” [61]. Therefore, to address this problem, techniques such as prompting,

fine tuning and RAG [62] have emerged.

2.2.3.1 Prompting

Prompting in LLMs refers to providing specific instructions or context to guide the model's text generation [26, 35]. Then, the “prompt engineering” term mentioned in [26] is a rapidly evolving discipline focused on crafting optimal prompts to achieve specific goals with generative AI models. It involves understanding the model's capabilities, limitations, and operational context to guide its outputs effectively. Thus, the prompt is a conditioning signal influencing the model's language generation process. According to Zhu et al., [35], there are various prompting strategies, including zero-shot prompting, few-shot prompting, and chain-of-thought prompting [63].

1. **Zero-Shot Prompting:** This technique is applied without providing the artificial intelligence model with any previous examples or specific context. The model relies on its pre-existing knowledge and language understanding to generate coherent and contextually relevant expanded terms for original queries [35].
2. **Few-Shot Prompting:** This technique involves training the model with a few examples to perform a task. One-shot prompting, a specific type of few-shot prompting, refers to the model's ability to perform a task after being given only one example. These examples serve as explicit instructions, enabling the model to adapt its language generation to the specific task or domain at hand [35].
3. **Chain-of-Thought Prompting:** This strategy involves iterative prompting [35]. It improves the reasoning skills of language models by allowing them to decompose multi-step problems into intermediate steps, thus enabling them to solve problems that cannot be addressed with standard prompting methods [26].

An increasingly adopted practice involves explicitly instructing LLMs to refrain from disseminating false or unverifiable information through the design of the “system prompt” [57]. For instance, the system prompt used in LLaMA 2 chat [39] exemplifies this approach: *“If a question does not make any sense, or is not factually coherent, explain why instead of answering something not correct. If you do not know the answer to a question, please do not share false information.”*

2.2.3.2 Fine-Tuning

Fine-tuning is an effective method for adapting LLM to specific domains. This process usually starts with a pre-trained model, such as GPT [37], PaLM [38], or LLaMA [10, 39]. The pre-trained model is then further trained on a dataset specific to the target domain [35]. This additional training helps the LLM learn the unique patterns, terminology, and context of the domain, thereby improving its ability to generate high-quality responses relevant to that domain.

2.2.3.3 RAG

One significant limitation of pre-trained LLMs is their inability to maintain up-to-date knowledge or access private and domain-specific information. This gap is addressed by RAG [15, 62]. According to Gao et al., [64], a RAG process consists of several key components illustrated in Fig. 2.6: Input, Indexing, Retrieval, and Generation. The RAG framework typically includes a vector database, a retriever, a re-ranker, and a generator [15]; it is commonly implemented using tools such as LangChain [65] and LLamaIndex [66].

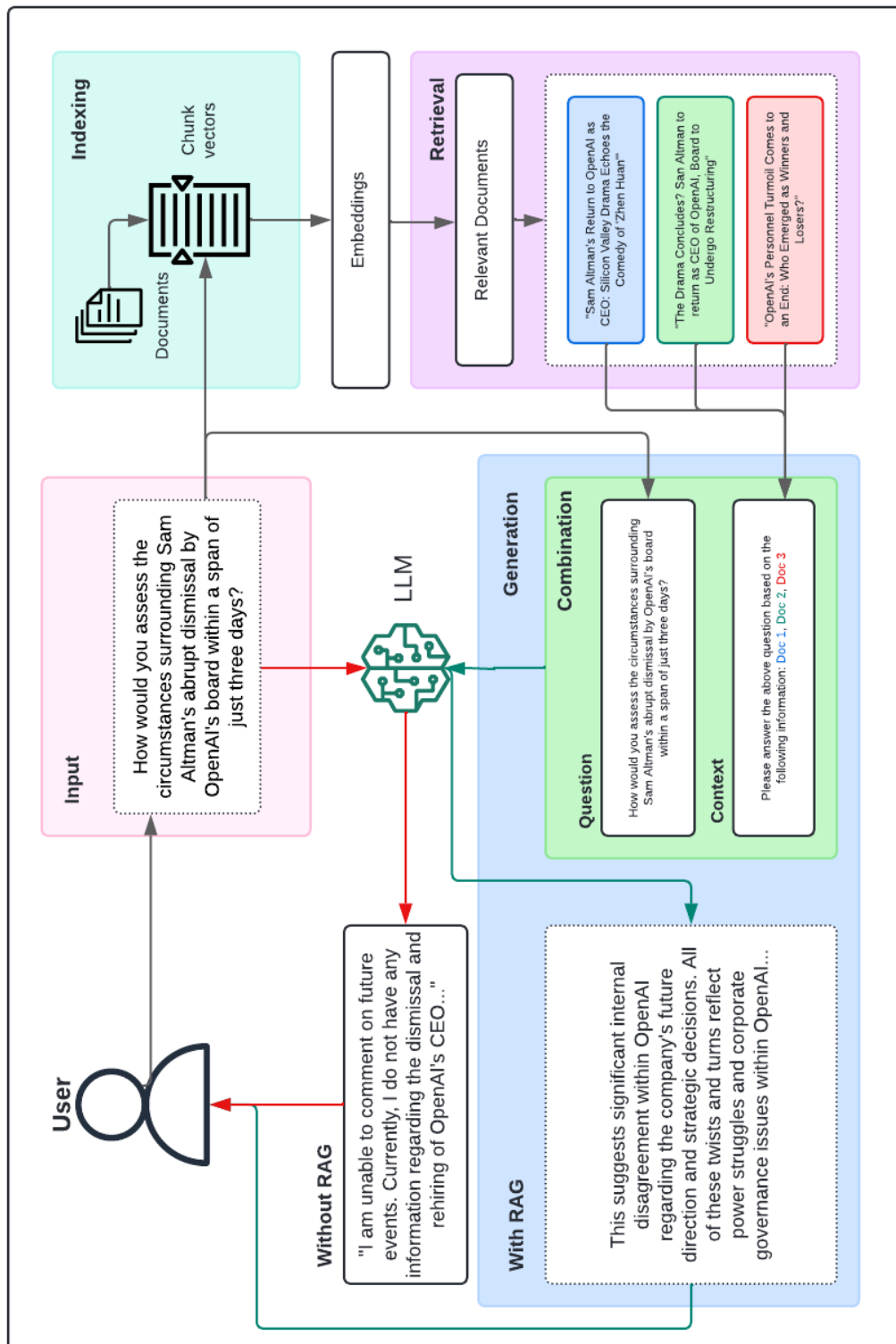


Figure 2.6: Illustration of the RAG process, as applied to question answering [64].

Chapter 3

State of the Art

3.1 Literature Review on LLMs in Education

A comprehensive search was conducted through 151 publications in Google Scholar using the following query:

“Chatbot” AND (“LLaMA” OR “PALM” OR “GPT”) AND (“Education” OR “Learning” OR “Teaching”) AND (“LLM” OR “Large Language Model”) AND (“RAG”) AND ((“Mathematics” OR “Math” OR “Mathematical”) AND (“problem-solving” OR “exercises”))

From this search, only a limited number of studies focus on using LLMs to solve mathematical questions. Most of the research is oriented toward addressing queries in physics, medicine, business, or programming. In addition, many studies involve the fine-tuning of PaLM, GPT, or LLaMA models in all their variants, especially in LLaMA versions 1, 2, and 3, the research of which has been on the rise. Some studies employ RAG to incorporate relevant information, typically from established datasets in Chinese and English. So, there is a notable lack of datasets and research focused on the Spanish language.

3.2 Chatbots in Education

The present section provides insights into the challenges and opportunities these technologies present for enhancing learning experiences across different subjects. By examining

these studies, we aim to understand the current landscape and identify potential areas for further research and development in the use of LLMs in education.

3.2.1 GPTs: OpenAI Custom Chatbot

On November 6, 2023, OpenAI announced the custom GPTs [43]. This addition to commercially available chatbots allows users to upload documents and answer questions using information extracted from uploaded PDFs or text documents. However, this platform is less suitable for educational purposes due to its associated costs and usage restrictions.

The tool supports domain-specific adaptation through RAG [67]. Despite its capabilities, the financial implications and potential risks to academic integrity, such as third-party management of student data, pose significant drawbacks [68].

3.2.2 MathGPT

MathGPT [69] is an LLM developed by the TAL Education Group. It focuses on math-related problem-solving and lecturing for worldwide institutions and math enthusiasts. The MathGPT playground has attracted significant attention from students and researchers. However, access to the platform requires a phone number from specific regions, making it less accessible to users, particularly those in Latin America.

The model was trained using two open-source datasets of Chinese and English mathematical competition questions, each consisting of 5,000 items (3,000 for training and 2,000 for testing). Both datasets can be found on HuggingFace TAL SCQ5K.

3.2.3 NewtBot: LLM-as-tutor for Physics

Lieb et al. [70] introduced NewtBot, an academic chatbot designed for secondary education physics. Powered by the GPT-3.5 model, NewtBot utilizes prompt engineering techniques, specifically employing system prompting to function in “Tutor” or “Feedback” modes. While the system proves beneficial, it necessitates payment for using the OpenAI API and lacks a RAG system for interactive chatting over educational material [71]. Additionally, it does not leverage open-source LLMs.

3.2.4 Chatting Over Course Material

In his master's thesis, Monteiro [71] performs a "RAG" analysis with Phi3 and LLaMA 3, utilizing ChromaDB [72] and Faiss [73] as vector stores to process mathematics and physics questions. For their experiment, they created 183 QA pairs for physics and 200 for mathematics. They obtained accuracy results for each Physics RAG of 0.64 and each Maths RAG of 0.66. These values help us understand how using different vector stores and configurations can enhance accuracy in models such as Phi3 and LLaMA 3.

3.2.5 Knowledge Navigator

In [74], a system is presented that takes documents of different formats such as PDF, docx, or xlsx to perform text extraction and subsequent embedding generation using an OpenAI model. The resulting embeddings are stored in the FAISS vector database for RAG processing to a GPT-3.5 or LIDA model in a Streamlit user interface. In the present work they obtained a system that by means of RAG allows the creation of questions and answers of documents and by using Streamlit they built a platform in which they could use the chatbot in a simpler way.

3.2.6 Large Language Model Intelligent Agent Tutoring in Higher Education Courses using a RAG Approach

In the paper [75], the authors used the LLamaIndex library to perform the loading and indexing of educational material. The authors used Typescript language to implement a system that takes PDF documents that are then loaded and split into 512 chunks to better handle the amount of text. The embeds are then generated and stored in a vector store in the same LLamaIndex library. In the present work they use the OpenAI GPT-4 model and perform a RAG process for retrieving large volumes of educational content. It is mentioned that the accuracy and effectiveness of the chatbot depends on the quality of the academic materials.

3.2.7 Large language model-powered chatbots for internationalizing student support in higher education

In [76], this work uses Python due to its large number of libraries and especially the support of the Langchain library and the OpenAI libraries. The chatbot essentially uses the GPT3.5-Turbo and GPT 4-Turbo model. They also use the ada-002-V2 model to perform the embeddings, all of this using the OpenAI API. In this way, its pipeline has been to process the documents using the ada-002-v2 model to generate the embeddings and use ChromaDB as a vectorstore. Subsequently, they recovered the 5 most semantically relevant documents and they are united in a conversation chain that subsequently passes to the GPT 3.5-Turbo model to perform a verification on the notice and finally a final response is generated in GPT 4-Turb. This work shows the use of the system drives quite extensive to achieve a better integration of the context. And it also highlights the use of a second warning when entering the output from the GPT 3.5-Turbo model to the GPT-4-Turbo model.

Chapter 4

Methodology

4.1 Methodology

The research methodology consists of three main phases. The first phase involves creating a vector database by extracting information from every single page of a mathematics textbook in order to build a database. This metadata will be converted into embeddings using the all-MiniLM-L6-v2 model hosted on HuggingFace and then uploaded to Pinecone for access through LangChain as a document retriever.

The second phase focuses on quantizing the LLaMA 2 7B chat model developed by Meta, applying quantization techniques to optimize the model's performance and efficiency. The final phase is a comparative analysis of different configurations, evaluating the quantized LLaMA 2 7B chat model with and without Retrieval-Augmented Generation (RAG) techniques, utilizing the vector database, and integrating with the LangChain framework.

4.1.1 Development of a Vector Database with Pinecone

This section describes the process of developing a vector database for mathematics using Pinecone. It begins by detailing the extraction and structuring of data from 6th edition of the textbook *Precalculus: Mathematics for Calculus* by James Stewart in its Spanish version. Therefore, the data is converted into embeddings using the all-MiniLM-L6-v2 model from HuggingFace. Finally, embeddings and metadata are uploaded to Pinecone for being used as retriever.

4.1.2 Math Database Creation

To establish a consistent and comprehensive database, the renowned textbook *Precalculus: Mathematics for Calculus* by James Stewart [77] in its Spanish version was used for collecting metadata. This textbook is highly appreciated for its clear and accessible explanations, wide range of examples, and well-structured approach to the subject. At the same time, it is one of the most widely used textbooks in universities in Latin America.

The process involved a meticulous review of each page of the textbook, where was carefully identified and extracted key elements that encapsulate the core concepts and exercises presented. This rigorous extraction process ensured that the information included in our database was both pertinent and valuable for our purposes. An overview of this process and some examples of the extracted information are illustrated in Figure 4.1.

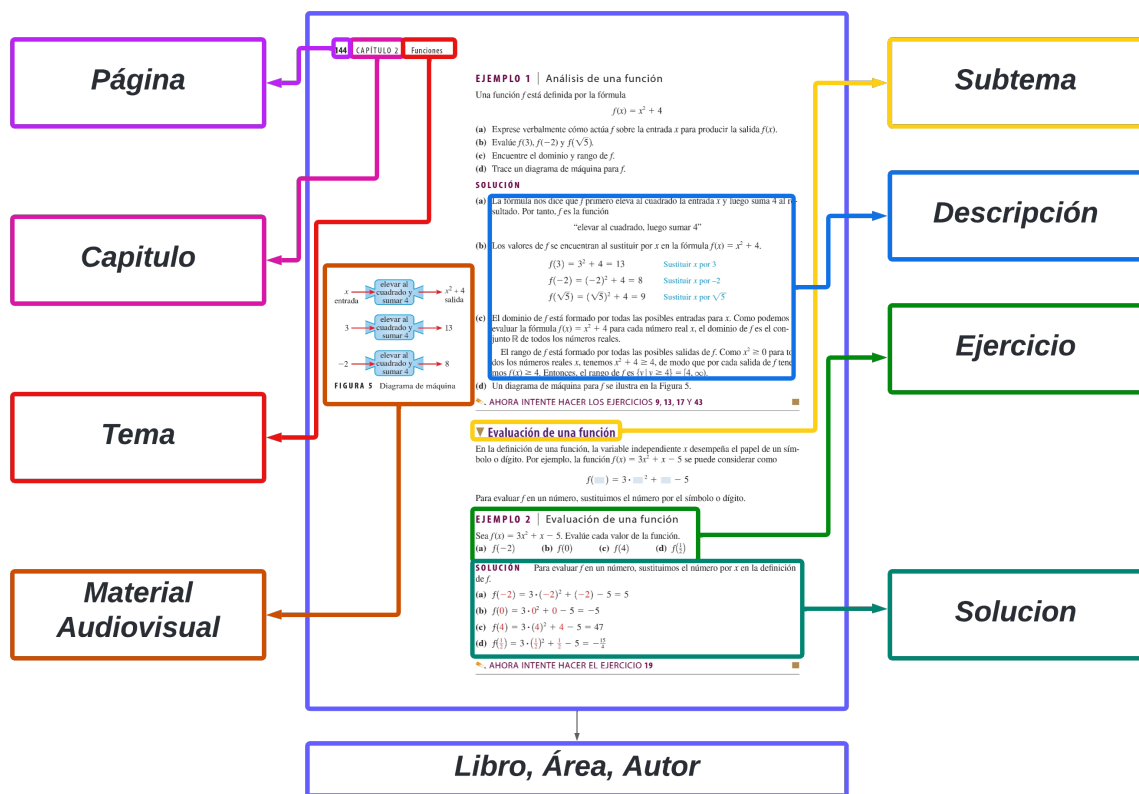


Figure 4.1: Metadata elements extracted from the book “*Precalculo: Matemáticas para el Cálculo*”, (pag. 144) to be included in our vectorial database [77].

The data was structured in a detailed and organized manner, as shown in Table 4.1. Each entry includes fields such as the mathematical area covered, the author, the chapter and subchapter numbers, a brief description of the content, and any associated exercises

or supplementary audiovisual materials. This structured approach facilitates easy access and retrieval of information.

Field	Description
Area	The specific mathematical area covered by the content
Author	Author(s) of the material
Chapter	Chapter number within the textbook
Description	Brief description of the content
Exercise	Mathematical exercise or problem provided
Textbook	Name of the textbook
Audiovisual Material	Supplementary audiovisual material related to the content
Subchapter	Subchapter or section within the chapter
Subtopic	Subtopic or specific theme addressed within the subchapter
Topic	Topic or broader category to which the content belongs
Text	Text that helps us to create embeddings

Table 4.1: Structure of the mathematics database’s metadata uploaded to Pinecone using 384 dimensions for semantic search

Once our database is structured, it needs to be converted into embeddings. For this, the model all-MiniLM-L6-v2 from HuggingFace is used for embed each row of our database. This model maps sentences and paragraphs into a 384-dimensional dense vector space, which can be used for tasks like clustering or semantic search. After the embedding process, our math database is projected using the TensorFlow Embedding Projector with the Principal Component Analysis method, as seen in Fig. 4.2.

This projection allows us to visually verify the clustering of elements based on their metadata, ensuring that related items are grouped appropriately and that the embeddings accurately represent the structure of the data.

4.1.3 Uploading embedding and metadata to Pinecone

To efficiently manage and search through this vast amount of data, Pinecone [78] was chosen as our vector store. Pinecone platform is particularly advantageous due to its user-friendly interface, which allows us to upload and organize data with just a few lines of code. Additionally, Pinecone provides metrics that track database retrievals, errors, latency, and the number of retrieved vectors over time.

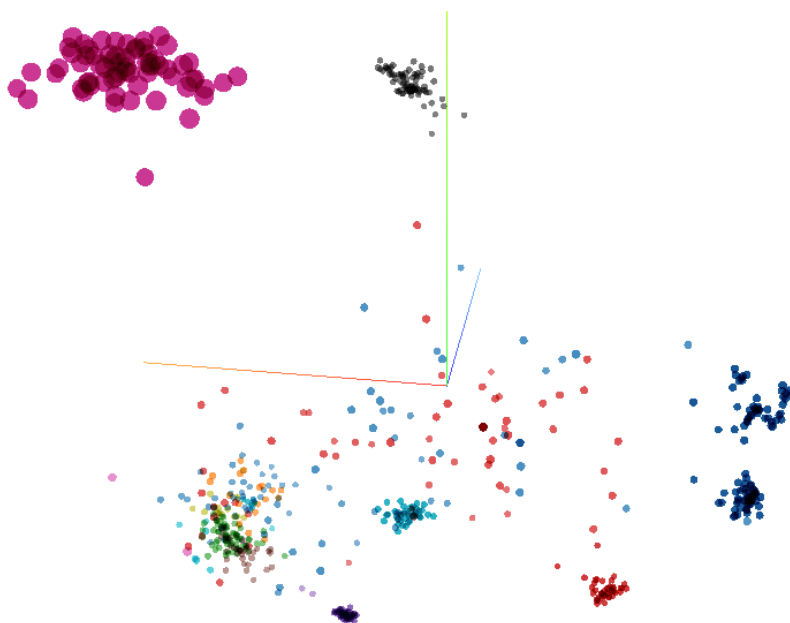


Figure 4.2: Projection of the math database using TensorFlow Embedding Projector with the principal components analysis method.

4.2 Quantization and Optimization of LLM model

4.2.1 Download Model Weights

In order to use the LLaMA 2 7B chat model, weights need to be request from Meta's dedicated platform for LLaMA. It requires registering at Meta Official LLaMA 2. After registration, Meta sends a confirmation email containing a unique URL for validation. Following validation, then is necessary to clone the LLaMA 2 GitHub Repository.

Once cloned into a directory (`llama`), it necessary to utilize the script `download.sh` to download the model weights using the link provided in Meta's email. This step downloads the model weights into a directory such as `LLaMA 2 7B chat/`, which includes a file named `consolidated.00.pth`. This file represents the LLaMA-2-7b model in a format compatible with PyTorch.

4.2.2 Local Deployment and Quantization

For local deployment and subsequent operations, by cloning and compiling LlamaCPP from LlamaCPP Repository. LlamaCPP facilitates the preparation of the model for integration

by converting it into the compatible GGML format. next explore various quantization options available for the LLaMA-2-7b model in GGML format, as shown in Table 4.2. Each method offers different trade-offs in terms of model size, resource usage, and inference speed, providing flexibility based on specific use cases and hardware constraints.

Name	Quant Method	Bits	Size	Max RAM Required
llama-2-7b.ggmlv3.q2_K	q2_K	2	2.87 GB	5.37 GB
llama-2-7b.ggmlv3.q3_K_S	q3_K_S	3	2.95 GB	5.45 GB
llama-2-7b.ggmlv3.q3_K_M	q3_K_M	3	3.28 GB	5.78 GB
llama-2-7b.ggmlv3.q3_K_L	q3_K_L	3	3.60 GB	6.10 GB
llama-2-7b.ggmlv3.q4_0	q4_0	4	3.79 GB	6.29 GB
llama-2-7b.ggmlv3.q4_K_S	q4_K_S	4	3.83 GB	6.33 GB
llama-2-7b.ggmlv3.q4_K_M	q4_K_M	4	4.08 GB	6.58 GB
llama-2-7b.ggmlv3.q4_1	q4_1	4	4.21 GB	6.71 GB
llama-2-7b.ggmlv3.q5_0	q5_0	5	4.63 GB	7.13 GB
llama-2-7b.ggmlv3.q5_K_S	q5_K_S	5	4.65 GB	7.15 GB
llama-2-7b.ggmlv3.q5_K_M	q5_K_M	5	4.78 GB	7.28 GB
llama-2-7b.ggmlv3.q5_1	q5_1	5	5.06 GB	7.56 GB
llama-2-7b.ggmlv3.q6_K	q6_K	6	5.53 GB	8.03 GB
llama-2-7b.ggmlv3.q8_0	q8_0	8	7.16 GB	9.66 GB

Table 4.2: Quantization options available for the LLaMA-2-7B model. For more information, visit Llama 2 7B - GGML.

Once the appropriate quantization method is selected, model is converted into f16 format using a conversion script provided in the LlamaCPP repository. Finally, model is optimized by applying the chosen quantization method. This detailed process ensures that the LLaMA-2-7b model is effectively prepared and optimized for seamless integration, aligning with specific performance and deployment requirements.

4.2.3 Large Language Model and LangChain

In this phase, LangChain framework [65], specialized for large language models, evaluate the performance of the quantized LLaMA 2 7B Chat model through three distinct configurations. These configurations aim to validate the model's functionality and assess its suitability for use in a chatbot. The configurations include (1) using the base quantized model, (2) enhancing the base quantized model with LangChain's LLMChain to provide prompt-based instructions, and (3) integrating the RetrievalQA Chain from LangChain with the quantized model and a system prompt. This final configuration will leverage the

vector store created in the previous section to apply the Retrieval Augmented Generation (RAG) technique.

4.2.3.1 Quantized Model

In this first sub-phase, the quantized model will be used to determine the hyperparameters that optimize the inference times and define the amount of text required. This process will help us to adjust the model to achieve an efficient and adequate performance to our needs.

4.2.3.2 Quantized Model and System prompt

In this sub-phase, quantized model is used together with a system prompt in the Langchain framework chain called LLMChain, shown in Fig. 4.3, to evaluate whether the model is able to identify its role, its name and the appropriate way to respond to users. This will allow us to verify if the model correctly understands its role and provides consistent responses according to the instructions given based only on the system prompt.

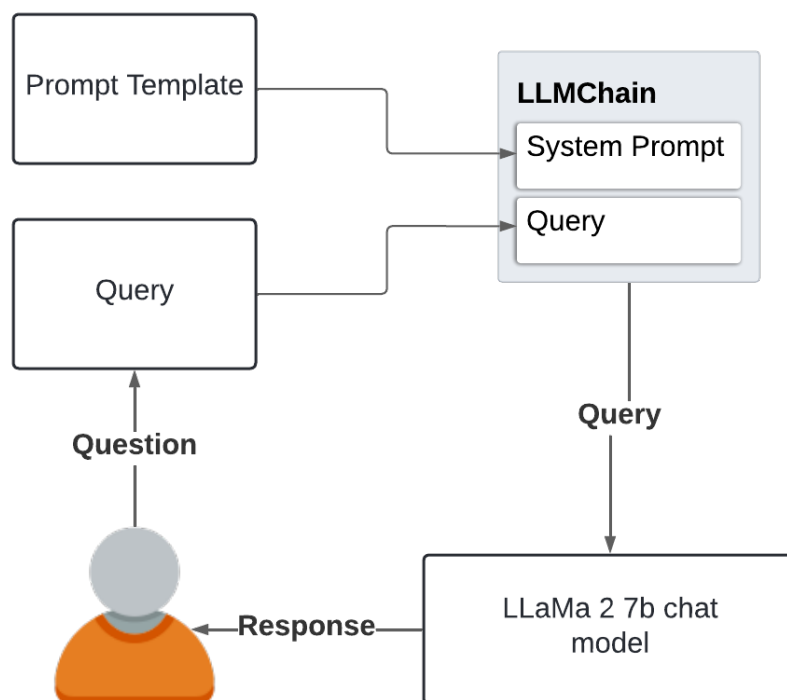


Figure 4.3: Workflow diagram of LLMChain in LangChain for utilizing the LLaMA 2 7B Chat model with a system prompt provided.

4.2.3.3 Quantized Model, System prompt and RAG

In the final configuration, RetrievalQA from Langchain and the vector store (Pinecone) to retrieve documents are use together. Each query will be embedded using model all-MiniLM-L6-v2. It will allow us to augment the model's responses by retrieving data from our previously created dataset, enabling the model to use relevant context from this data to provide more accurate and context-aware responses, as shown in Fig. 4.4.

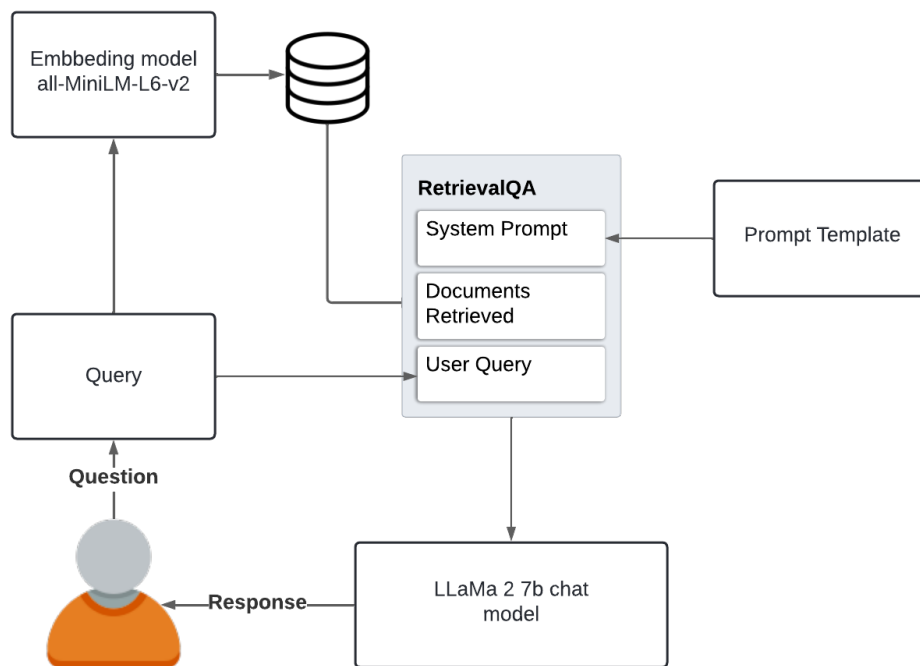


Figure 4.4: Workflow of a chatbot using the Langchain framework to create a RetrievalQA chain that combines system prompts, context, and user queries to provide context-augmented responses.

4.3 Comparative Analysis of Model Configurations and Retrieval-Augmented Generation Techniques

To evaluate the performance and effectiveness of the different model configurations and RAG techniques, a comparative analysis will be conducted. This analysis will focus on several key aspects: the quality of document retrieval, inference times, and the impact of context on the generated responses.

4.3.1 Document Retrieval Process

To assess the performance of our retriever, multiple queries will be executed against the vector database and evaluate each query based on the relevance and accuracy of the retrieved documents. It involves executing a series of queries to test which documents are retrieved from Pinecone, analyzing the retrieved documents for their relevance and alignment with the query, and recording and categorizing the documents according to their semantic relevance scores and content accuracy. This process will provide insights into the retriever's effectiveness in fetching documents pertinent to the given queries.

4.3.2 Model Configuration Analysis

For each model configuration, we measure response quality, which involves analyzing the quality of the responses produced by the model; and text generation analysis, which distinguishes between text generated solely by the model and text generated with the assistance of context provided by the retriever.

Chapter 5

Results and Discussion

In this chapter, the vector database and its semantic search capabilities will be analyzed. Then, we will evaluate the different configurations of the model using the Langchain framework. Finally, we will compare the answers generated with and without retrieval augmented generation to evaluate its ability to generate text based on context.

5.1 Vector Database in Pinecone

5.1.1 Math Database

The database was successfully constructed as detailed in Table 4.1, including 1,300 entries derived from twelve chapters of the book, with chapter 13 excluded due to its focus on advanced topics typically covered in specialized university courses. The database was then filled with the necessary metadata to enable semantic searches using the various metrics available in the Pinecone system. These similarity metrics are summarized in Table 5.1.

Similarity Metric	Vector Properties Considered
Euclidean Distance	Magnitudes and direction
Cosine Similarity	Only direction
Dot Product Similarity	Magnitudes and direction

Table 5.1: Vector properties considered for different similarity metrics in Pinecone documentation.

According to Pinecone’s documentation [79], three similarity metrics are particularly relevant for projects involving Large Language Models (LLMs): Euclidean Distance (Fig.

5.1a), commonly employed in recommendation systems; Dot Product Similarity (Fig. 5.1b), often used in collaborative filtering and matrix factorization; and Cosine Similarity (Fig. 5.1c), which is frequently utilized for semantic search and document classification.

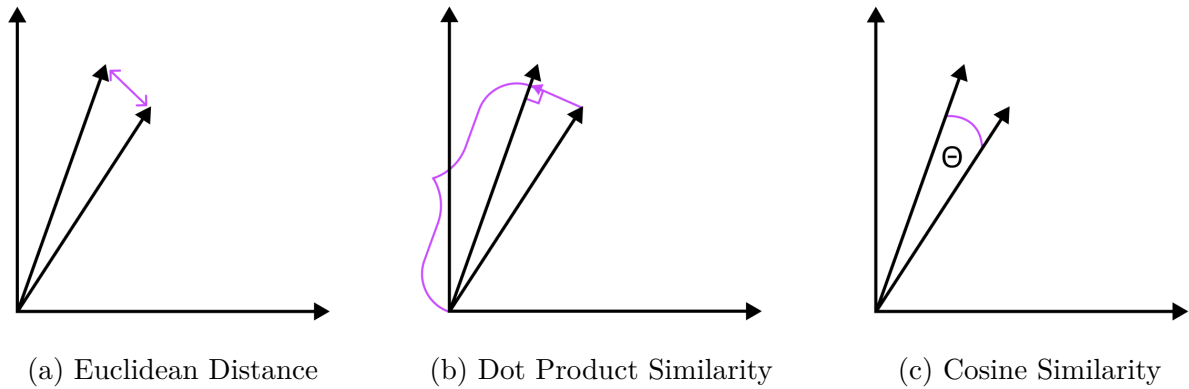


Figure 5.1: Comparison of similarity metrics used in the Pinecone system. Each metric evaluates the similarity between vectors based on different properties.

Cosine similarity was selected for this investigation due to its alignment with our objectives. Following this decision, the database was uploaded to the Pinecone system. This process involved converting the database from an Excel spreadsheet to a CSV file using a script developed in Google Colab. The CSV file was then used to generate embeddings with the all-MiniLM-L6-v2 model, which is compatible with cosine similarity [79].

In the study “MathGPT” by [69], TAL Education Group developed a language model trained on two open-source datasets, primarily in Chinese and English, which included mathematics competition questions. Specifically, they used the TAL SCQ5K dataset, available on HuggingFace. In contrast, we have developed our dataset in Spanish, incorporating fields that are designed to enhance document retrieval. Our approach aims to maximize semantic search capabilities using Pinecone’s platform and the cosine similarity metric.

5.1.2 Document Retrieval Process

A document retrieval process (DRP) was developed that uses cosine similarity as a similarity metric to facilitate semantic searches. DRP employs a query vector, which is generated by transforming user queries into vector representations using the all-MiniLM-L6-v2 model.

The all-MiniLM-L6-v2 model is designed to capture the semantic essence of the queries and convert them into high-dimensional vectors, in this particular case, 384 dimensions.

Following the process described in Fig. 5.2. These vectors are then compared with a set of pre-stored vector embeddings in the Pinecone database so that the system measures the closeness between the query vector and the stored vectors.

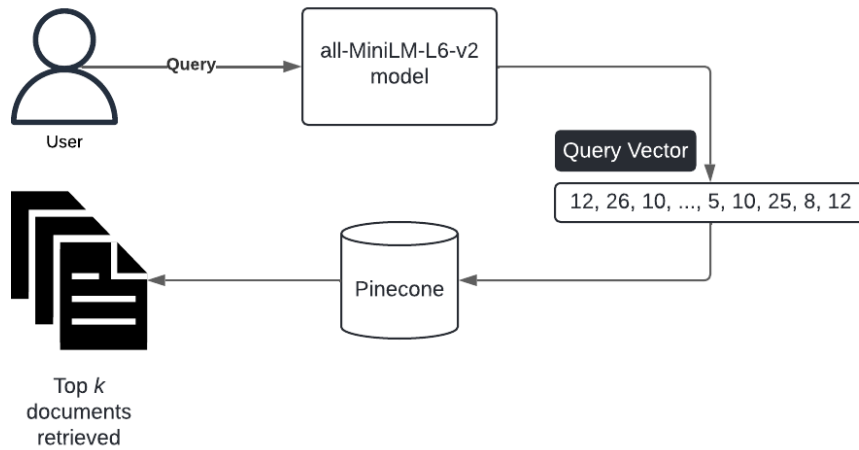


Figure 5.2: Workflow of a DRP: This process involves converting a text query into an embedding vector, which is then used to retrieve the top k elements from the Pinecone vector store.

The cosine similarity metric quantifies the similarity by calculating the cosine of the angle between the query vector and each stored vector, as shown in Fig. This approach ensures that the system retrieves documents semantically relevant to the user’s query. In Tab.5.2, we can observe three examples of queries performed in the DRS. The use of cosine similarity helps to identify the most relevant documents by evaluating their degree of openness with the query vector representation as shown in Fig. 5.1c.

K	1		2		3	
Query	ID	Score	ID	Score	ID	Score
I need help with real numbers	0	0.740439117	6	0.688518822	1	0.640005052
I want to learn about sets and intervals	7	0.6274454	9	0.555514097	8	0.470150709
I want to learn about radicals and the definition of a root	18	0.666864038	19	0.541175246	21	0.521211445

Table 5.2: Top k elements associated with each query vector, ranked by cosine similarity. Each entry includes the document ID and its similarity score with the query vector.

For the first query, “I need help with real numbers” we retrieved documents with ID 0, 6, and 1 with similarity values of 0.740439117, 0.688518822, and 0.640005052, respectively.

Thus, we have managed to retrieve the k documents with the highest $= \beta$ from our vector database. In the same way for query number 2, “I want to learn about sets and intervals” we have the top $k = \beta$ being the documents with *ID* 7, 9, and 8. Finally, we have the documents for the query “I want to learn about radicals and the definition of a root” where we have the documents with *ID* 18, 19, and 21.

Our results demonstrate a higher similarity compared to the work of Monteiro [71], who used ChromaDB and Faiss as his vector databases. In contrast, our approach involved creating a custom vector database using Pinecone, where we specified detailed fields as shown in Table 4.1. This customization enhances the effectiveness of semantic search in our retriever, leading to higher scores in queries to our Pinecone vector database.

5.2 Model Configuration Results

5.2.1 Quantized Model

The quantized model was evaluated by means of questions representative of the common use that users make when interacting with a chatbot. For this evaluation, the quantization *q4_0* was used, which allowed obtaining a model with a precision of 4 bits, a total weight of 3.79 GB, and a RAM usage of 6.29 GB. Using the LlamaCPP library, inference was performed directly on the quantized model, which allowed obtaining answers to the queries made.

In Tab. 5.3, it is observed that each response generally addresses the user’s query appropriately. However, the model tends to provide encouragement and general advice on how to improve in the subject rather than offering specific material directly to the user. Additionally, while the model suggests websites or repositories where resources can be found, it does not provide direct access to these resources from the generated text.

By using the *q4_0* quantization, we also identified the optimal hyperparameters for efficiently running the quantized model on our system. The hardware setup includes an i7-11800H processor from the 11th generation with a clock speed of 2.30 GHz, 16 GB of RAM, and 4 GB of VRAM on an NVIDIA GeForce RTX 3050 Ti. Based on these specifications, the following hyperparameters were selected:

- **Temperature:** 0.5 (adjusts the randomness of outputs)

Query	Response
How can I improve in algebra?	Hello! Certainly! Improving in algebra can be challenging, but with regular practice and the right resources, it's achievable. Here are some tips: Practice consistently, understand the fundamentals, use visual aids, and more.
Can you help me with a math problem?	Of course! I'm here to assist. Please provide more details about the math problem you are facing so I can offer a precise and useful response.
Where can I find resources to study functions?	Hello! There are many resources available for studying functions, both online and in print. Consider exploring Khan Academy, MIT OpenCourseWare, and Wolfram MathWorld, among others.
Do you have any suggestions for an exam?	Absolutely! Studying for an exam can be challenging, but effective strategies can help. Here are some tips: Create a study schedule, review and organize your notes, use flashcards, and practice with sample questions.
What is the key concept in trigonometry?	Hello! In trigonometry, the fundamental concept is the angle. It measures the amount of rotation around a plane or coordinate axis and is used to describe an object's orientation in space. Key techniques include essential, analytical, and synthetic trigonometry for solving problems involving angles, triangles, and trigonometric functions.

Table 5.3: Queries submitted to the quantized LLaMA 7B chat model using *q4_0* quantization and the corresponding responses for each query.

- **Number of GPU layers:** 12
- **Number of threads:** 4
- **Maximum tokens:** 2048
- **Context window:** 4096

When the quantized model is loaded into memory for the first time, it may take up to 30 minutes to fully load into both RAM and VRAM. After this initial loading period, the first inference for a query might take up to 3 minutes. Once the model is fully loaded, subsequent inferences typically take between 8 to 30 seconds, depending on the length of the output.

5.2.2 Quantized Model and System Prompt

Once our hyperparameters are defined, the Langchain framework is used to incorporate a prompt template that will guide the model in generating its responses. This prompt

template, now referred to as the system prompt, can direct the model on various aspects such as writing style, character count, the inclusion of emojis, and even the language in which the response should be written.

To guide the model in the best way, the following **system prompt** was used for the tests shown in Tab. 5.4. It was decided to give the following guidelines to the model in order to test its ability to understand specific instructions for the writing format.

Query	Response
How can I improve in algebra?	¡Hola! (smiling emoji) I am glad you asked! Improving in algebra can be challenging, but with consistent practice and the right mindset, you can definitely improve. Here are some tips that might help: 1. Understand the basics 2. Practice, practice, practice 3. Use visual aids
Can you help me with a math problem?	¡Hola! (smiling emoji) Of course, I'd be happy to help you with your math problem! Can you please provide the problem you are working on and what you have tried so far? That way, I can assist you better. (thinking emoji)
Where can I find resources to study functions?	¡Hola! (smiling emoji) I am glad you are interested in studying functions! Here are some awesome resources that can help you learn more about them: 1. Wolfram MathWorld : Wolfram MathWorld is an incredible resource for learning about different types of functions, including polynomial functions, trigonometric functions, and more! They also have a ton of examples and practice problems to help you reinforce your understanding. (writing emoji) ¡ https://mathworld.wolfram.com/Function.html !
Do you have any suggestions for studying for an exam?	¡Hola! (happy emoji) As a virtual math assistant, I am here to help you with any math-related questions you may have! (thinking emoji) When it comes to studying for an exam, here are some tips that may help you: 1. Break up your study sessions into smaller chunks : Instead of studying for long hours, break up your study sessions into shorter time blocks. This will help you stay focused and retain information better. (paper emoji)
What is the key concept in trigonometry?	¡Hola! (happy emoji) As a virtual math assistant, I am here to help you with any math-related questions you may have! (thinking emoji) In trigonometry, the key concept is the relationship between the angles and the sides of triangles. (ruler emoji) Trigonometry deals with the study of triangles, especially those with right angles (90 degrees). The main tools in trigonometry are the sine, cosine, and tangent functions, which are used to solve problems involving right triangles. (angle emoji)

Table 5.4: Queries to the model with an integrated system prompt and the responses generated based on these queries combined with the system prompt.

system_prompt = “Hello, your name is EDUAI. You are a chatbot developed by Universidad Yachay Tech and the Universidad Internacional del Ecuador (UIDE). You are a math assistant for high school and university students.

You always answer in Spanish, you always greet the user, you use Markdown format to write your answers, and you also use emojis to make learning more interactive. You always encourage the student to keep learning and never give up.”

In Tab. 5.4, it is shown that the quantified LLM has now followed the instructions without major problems. When analyzing each response, it can be observed that the instruction of *Always greet the user* follows it without major problem since it writes in each response ¡*Hola!* , it also uses emojis when starting or ending a sentence to emphasize the content that has been explained in that sentence. The use of Markdown formatting is also present when using `**` to be able to write with different font sizes and be more visual and easy to understand the generated answer.

5.2.3 Quantized Model, System Prompt, and RAG

Finally, following the architecture outlined in Fig. 4.4, we integrated all components into a single chatbot that encompasses the capabilities analyzed in each section and subsection up to this point. The RetrievalQA chain from Langchain will be employed for document retrieval, configured to return the top $k = 3$ documents based on cosine similarity as our metric. Concurrently, we used the quantized model with the (q4.0) method for inference to optimize hardware resources. Additionally, we utilized the system prompt from the previous subsection, with a new line added to enable access to the context provided by the document retriever.

Thus, the system prompt to be used for the tests in Tab. 5.5 is as follows.

```
system_prompt = “Hello, your name is EDUAI. You are a chatbot developed by Yachay Tech University and the International University of Ecuador (UIDE). You are a mathematical assistant for high school and university students. You always answer in Spanish, you always greet the user, you use Markdown format to write your answers, and you use emojis to make learning more interactive. You always encourage the student to keep learning and never give up. To answer effectively, you use the following context to address the students’ questions:
```

```
context: context
```

If the question cannot be answered using the context, do not respond.

question: query”

The user’s question is combined with the system prompt and the context derived from the top $k = 3$ documents in our Pinecone database. By providing this context, we follow the augmented process of generating the answer, as described by [64], which includes Input, Indexing, Retrieval, and Generation.

Query	Response
Hi, I need help with real numbers. Do you have a video about the properties of real numbers?	Certainly! To help you better understand the properties of real numbers, I will provide some videos that might be useful. First, here is a video from the Atlanix channel that explores how real numbers can be represented on a number line: Real Numbers on the Number Line - Atlanix (https://www.youtube.com/watch?v=ncFaIIVTNpo). This video explains how real numbers can be represented as points on a line and how to calculate the distance between two points on the line. Next, here is a video from the Daniel Carreón channel that shows how to locate various fractions on the number line: Locating Fractions on the Number Line - Daniel Carreón (https://www.youtube.com/watch?v=TvLbbFKIfEw).
Hi, I want to learn about radicals and the definition of root. Do you have videos about the properties of radicals?	Hello! Certainly! To learn more about radicals, I recommend watching the following videos: 1. Channel: Matemáticas con Grajeda - Title: Simplify Radicals (Detailed Explanation) - Link: https://www.youtube.com/watch?v=_AtmBKnyrv4 2. Channel: Vitual - Title: Radical Notation in Scientific Notation (Explanation) - Link: https://www.youtube.com/watch?v=_AtmBKnyrv4 3. Channel: Matemáticas con Juan - Title: Writing Powers as Radicals (Explanation) - Link: https://www.youtube.com/watch?v=zc8UqjjhTiQ 4. Channel: Matemáticas con Alex - Title: Simplification of Radicals (Explanation) - Link: https://www.youtube.com/watch?v=2HachLBuoZo

Table 5.5: Comparison of responses using a system prompt to provide guidelines for a quantized model.

5.2.3.1 RAG Evaluation

Using the results obtained in Tab. 5.5, it can be noticed how the model has used the metadata content of the retrieved documents by the retriever in a way that has improved its response. By requesting videos found in the metadata of the retrieved documents, the user has obtained the links and the channel and, at the same time, a short explanation of the area in which he is having difficulties. We have two questions on very different topics, but we have received several videos that can be accessed only by using the link provided.

Response	Total Characters	Context Characters Generated	Percentage of Context Characters
Certainly! To help you better understand the properties of real numbers, I will provide some videos that might be useful. First, here is a video from the Atlanix channel that explores how real numbers can be represented on a number line: Real Numbers on the Number Line - Atlanix (https://www.youtube.com/watch?v=ncFaIIVTNpo). This video explains how real numbers can be represented as points on a line and how to calculate the distance between two points on the line. Next, here is a video from the Daniel Carreón channel that shows how to locate various fractions on the number line: Locating Fractions on the Number Line - Daniel Carreón (https://www.youtube.com/watch?v=TvLbbFKIfEw)	656	393	59.91
Hello! Certainly! To learn more about radicals, I recommend watching the following videos: 1. Channel: Matemáticas con Grajeda - Title: Simplify Radicals (Detailed Explanation) - Link: https://www.youtube.com/watch?v=_AtmBKnyrv4 2. Channel: Virtual - Title: Radical Notation in Scientific Notation (Explanation) - Link: https://www.youtube.com/watch?v=_AtmBKnyrv4 3. Channel: Matemáticas con Juan - Title: Writing Powers as Radicals (Explanation) - Link: https://www.youtube.com/watch?v=zc8UqjjhTiQ 4. Channel: Matemáticas con Alex - Title: Simplification of Radicals (Explanation) - Link: https://www.youtube.com/watch?v=2HachLBuoZo	656	565	86.13

Table 5.6: Details of the response, including total character count, number of context characters generated, and the percentage of context characters.

In Tab. 5.6, we see the responses generated by the chatbot powered by the LLaMA 2 7B chat model, system prompt, and RAG. The values of 59.91% and 86.13% for the percentage of context characters generated indicate how a significant portion of the responses is based on the context and instructions provided in the system prompt. It demonstrates the chatbot's effective use of the context to generate relevant and accurate responses.

Chapter 6

Conclusions and Future Work

In the present research, we focused on the development of an educational chatbot designed for high school and college mathematics courses. Our approach focused on the use of quantized LLMs, RAG techniques, and the construction of a vector database with metadata.

The creation of the vector database was particularly challenging, as it required careful consideration of which metadata would be most effective for semantic search. Despite these difficulties, the database construction was successfully completed. We selected cosine similarity as the similarity metric, which proved suitable for our objectives by comparing the angle between the query vector and the embeddings in the Pinecone vector database. This choice allowed us to retrieve relevant documents efficiently, yielding results that consistently outperformed those obtained with other tools such as ChromaDB and Faiss.

The model quantized using the `q4` method demonstrated solid performance in terms of both accuracy and resource usage. Although the responses were generally satisfactory, the model excelled in speed of inference without compromising the quality of text generation. Identifying the optimal hyperparameters for system performance was critical, and taking advantage of the different configurations of the LLamaCPP framework allowed us to achieve the best balance between performance and inference speed.

Integrating the quantized model with a system indicator was effective in guiding the generation of model responses. Incorporating features such as emojis, markdown formatting, and personalized language significantly improved the learning experience by making interactions more engaging and less monotonous. This customization allowed the model to better adapt to the user's needs and preferences.

The combination of the quantized model, system prompt, and RAG techniques resulted in a robust and efficient mathematics chatbot. The ability to perform semantic searches and utilize relevant documents as context substantially improved answer quality. Additionally, the Langchain framework facilitated efficient script generation for model utilization, given its specialization in LLMs.

In conclusion, the proposed tool offers a valuable resource for high school students to address academic gaps, while university entrance students in preparatory courses can benefit from it as a support tool during the university admission process. The methodologies and configurations developed in this study provide a solid foundation for future research and applications in virtual assistants and semantic information retrieval.

6.1 Limitations

The study faced several limitations, including the manual entry of each database item, which was time-consuming. Additionally, the constraints of available resources limited us to using the maximum quantization method, q4_0. The Langchain framework also presented challenges, such as limited documentation on proper system structuring and the presence of non-disableable features in some chains, which hindered achieving our specific objectives.

6.2 Future Works

Future research should aim to expand the dataset to incorporate more diverse and multilingual sources, potentially enhancing the model's robustness and adaptability. Additionally, investigating the integration of user feedback mechanisms could provide insights into iteratively refining the model's responses and overall functionality. Exploring more efficient quantization techniques and their impact on model performance and resource utilization could further optimize practical deployment in real-world applications.

Bibliography

- [1] J. Gerstein, “Moving from education 1.0 through education 2.0 towards education 3.0,” 2014.
- [2] N. Songkram, S. Chootongchai, J. Khlaisang, and P. Koraneekij, “Education 3.0 system to enhance twenty-first century skills for higher education learners in thailand,” *Interactive Learning Environments*, vol. 29, no. 4, pp. 566–582, 2021.
- [3] N. Rane, S. Choudhary, and J. Rane, “Education 4.0 and 5.0: Integrating artificial intelligence (ai) for personalized and adaptive learning,” *Available at SSRN 4638365*, 2023.
- [4] T. Huk, “From education 1.0 to education 4.0-challenges for the contemporary school,” *The New Educational Review*, vol. 66, pp. 36–46, 2021.
- [5] D. Keats and J. P. Schmidt, “The genesis and emergence of education 3.0 in higher education and its potential for africa,” *First monday*, vol. 12, no. 3, pp. 3–5, 2007.
- [6] H. A. Younis, N. I. R. Ruhaiyem, W. Ghaban, N. A. Gazem, and M. Nasser, “A systematic literature review on the applications of robots and natural language processing in education,” *Electronics*, vol. 12, no. 13, p. 2864, 2023.
- [7] B. Rawat, A. S. Bist, U. Rahardja, Q. Aini, and Y. P. A. Sanjaya, “Recent deep learning based nlp techniques for chatbot development: An exhaustive survey,” in *2022 10th International Conference on Cyber and IT Service Management (CITSM)*. IEEE, 2022, pp. 1–4.

- [8] Q. Lhoest, A. V. del Moral, Y. Jernite, A. Thakur, P. von Platen, S. Patil, J. Chaumond, M. Drame, J. Plu, L. Tunstall *et al.*, “Datasets: A community library for natural language processing,” *arXiv preprint arXiv:2109.02846*, 2021.
- [9] O. Sharir, B. Peleg, and Y. Shoham, “The cost of training nlp models: A concise overview,” *arXiv preprint arXiv:2004.08900*, 2020.
- [10] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.
- [11] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: <https://aclanthology.org/N19-1423>
- [13] K. Adnan and R. Akbar, “An analytical study of information extraction from unstructured and multidimensional big data,” *Journal of Big Data*, vol. 6, no. 1, pp. 1–38, 2019.
- [14] N. Rane, “Chatbot-enhanced teaching and learning: Implementation strategies, challenges, and the role of chatgpt in education,” *Challenges, and the Role of ChatGPT in Education (July 21, 2023)*, 2023.
- [15] J. Jiang, F. Wang, J. Shen, S. Kim, and S. Kim, “A survey on large language models for code generation,” *arXiv preprint arXiv:2406.00515*, 2024.
- [16] UNESCO, “Tecnología en la educación: ¿una herramienta en los términos de quién?” Informe de Seguimiento de la Educación en el Mundo, Tech. Rep., 2023.

- [Online]. Available: https://www.unesco.org/gem-report/sites/default/files/medias/fichiers/2023/07/7952%20UNESCO%20GEM%202023%20Summary_ES_Web.pdf
- [17] E. Arias Ortiz, C. Giambruno, A. Morduchowicz, and B. Pineda, “El estado de la educación en américa latina y el caribe 2023,” 2024.
- [18] G.-J. Hwang and C.-Y. Chang, “A review of opportunities and challenges of chatbots in education,” *Interactive Learning Environments*, vol. 31, no. 7, pp. 4099–4112, 2023.
- [19] G.-J. Hwang, H. Xie, B. W. Wah, and D. Gašević, “Vision, challenges, roles and research issues of artificial intelligence in education,” p. 100001, 2020.
- [20] T. Lehto, “Developing llm-powered applications using modern frameworks,” 2024.
- [21] D. Khurana, A. Koli, K. Khatter, and S. Singh, “Natural language processing: State of the art, current trends and challenges,” *Multimedia tools and applications*, vol. 82, no. 3, pp. 3713–3744, 2023.
- [22] J. Bharadiya, “A comprehensive survey of deep learning techniques natural language processing,” *European Journal of Technology*, vol. 7, no. 1, pp. 58–66, 2023.
- [23] J. Brownlee, “Deep learning for natural language processing,” *Machine Learning Mystery, Vermont, Australia*, vol. 322, 2017.
- [24] L. Deng and Y. Liu, *Deep learning in natural language processing*. Springer, 2018.
- [25] K. Chowdhary and K. Chowdhary, “Natural language processing,” *Fundamentals of artificial intelligence*, pp. 603–649, 2020.
- [26] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, “Large language models: A survey,” *arXiv preprint arXiv:2402.06196*, 2024.
- [27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [28] X. Amatriain, A. Sankar, J. Bing, P. K. Bodigutla, T. J. Hazen, and M. Kazi, “Transformer models: an introduction and catalog,” *arXiv preprint arXiv:2302.07730*, 2023.

- [29] M. U. Hadi, R. Qureshi, A. Shah, M. Irfan, A. Zafar, M. B. Shaikh, N. Akhtar, J. Wu, S. Mirjalili *et al.*, “Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects,” *Authorea Preprints*, 2023.
- [30] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [31] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, “Recurrent neural network based language model.” in *Interspeech*, vol. 2, no. 3. Makuhari, 2010, pp. 1045–1048.
- [32] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, “Transformers: State-of-the-art natural language processing,” in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 2020, pp. 38–45.
- [33] M. U. Hadi, R. Qureshi, A. Shah, M. Irfan, A. Zafar, M. B. Shaikh, N. Akhtar, J. Wu, S. Mirjalili *et al.*, “A survey on large language models: Applications, challenges, limitations, and practical usage,” *Authorea Preprints*, 2023.
- [34] H. Zhao, H. Chen, F. Yang, N. Liu, H. Deng, H. Cai, S. Wang, D. Yin, and M. Du, “Explainability for large language models: A survey,” *ACM Transactions on Intelligent Systems and Technology*, 2023.
- [35] Y. Zhu, H. Yuan, S. Wang, J. Liu, W. Liu, C. Deng, Z. Dou, and J.-R. Wen, “Large language models for information retrieval: A survey,” *arXiv preprint arXiv:2308.07107*, 2023.
- [36] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, “A survey of large language models,” *arXiv preprint arXiv:2303.18223*, 2023.
- [37] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

- [38] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, “Palm: Scaling language modeling with pathways,” *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.
- [39] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, “Llama 2: Open foundation and fine-tuned chat models,” *arXiv preprint arXiv:2307.09288*, 2023.
- [40] M. Shanahan, “Talking about large language models,” *Communications of the ACM*, vol. 67, no. 2, pp. 68–79, 2024.
- [41] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
- [42] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, “Improving language understanding by generative pre-training,” OpenAI, Tech. Rep., 2018.
- [43] OpenAI, “Chatgpt,” 2022, accessed: 2024-06-17. [Online]. Available: <https://openai.com/index/chatgpt/>
- [44] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen *et al.*, “Palm 2 technical report,” *arXiv preprint arXiv:2305.10403*, 2023.
- [45] H. Alipour, N. Pendar, and K. Roy, “Chatgpt alternative solutions: Large language models survey,” *arXiv preprint arXiv:2403.14469*, 2024.
- [46] B. Roziere, J. Gehring, F. Gloeckle, S. Sootla, I. Gat, X. E. Tan, Y. Adi, J. Liu, T. Remez, J. Rapin *et al.*, “Code llama: Open foundation models for code,” *arXiv preprint arXiv:2308.12950*, 2023.
- [47] S. G. Patil, T. Zhang, X. Wang, and J. E. Gonzalez, “Gorilla: Large language model connected with massive apis,” *arXiv preprint arXiv:2305.15334*, 2023.
- [48] A. Pal, D. Karkhanis, M. Roberts, S. Dooley, A. Sundararajan, and S. Naidu, “Giraffe: Adventures in expanding context lengths in llms,” *arXiv preprint arXiv:2308.10882*, 2023.

- [49] B. Huang, “Vigogne: French instruction-following and chat models,” 2023.
- [50] Y. Wang, H. Ivison, P. Dasigi, J. Hessel, T. Khot, K. Chandu, D. Wadden, K. MacMillan, N. A. Smith, I. Beltagy *et al.*, “How far can camels go? exploring the state of instruction tuning on open resources,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 74 764–74 786, 2023.
- [51] S. Tworkowski, K. Staniszewski, M. Patek, Y. Wu, H. Michalewski, and P. Miłoś, “Focused transformer: Contrastive training for context scaling,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [52] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, “Alpaca: A strong, replicable instruction-following model,” Stanford Center for Research on Foundation Models, <https://crfm.stanford.edu/2023/03/13/alpaca.html>, 2023, accessed: 2023-03-13.
- [53] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez *et al.*, “Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality,” <https://vicuna.lmsys.org>, 2023, accessed: 2023-04-14.
- [54] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “Qlora: Efficient finetuning of quantized llms,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [55] X. Geng, A. Gudibande, H. Liu, E. Wallace, P. Abbeel, S. Levine, and D. Song, “Koala: A dialogue model for academic research,” *Blog post, April*, vol. 1, p. 6, 2023.
- [56] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier *et al.*, “Mistral 7b,” *arXiv preprint arXiv:2310.06825*, 2023.
- [57] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen *et al.*, “Siren’s song in the ai ocean: a survey on hallucination in large language models,” *arXiv preprint arXiv:2309.01219*, 2023.

- [58] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, “Survey of hallucination in natural language generation,” *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.
- [59] E. Reiter, “A structured review of the validity of bleu,” *Computational Linguistics*, vol. 44, no. 3, pp. 393–401, 2018.
- [60] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, and S. Riedel, “Language models as knowledge bases?” 2019.
- [61] G. Marcus, “The next decade in ai: Four steps towards robust artificial intelligence,” 2020.
- [62] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, “Retrieval-augmented generation for knowledge-intensive nlp tasks,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.
- [63] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, “Chain-of-thought prompting elicits reasoning in large language models,” *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.
- [64] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, “Retrieval-augmented generation for large language models: A survey,” *arXiv preprint arXiv:2312.10997*, 2023.
- [65] H. Chase, “Langchain,” <https://github.com/langchain-ai/langchain>, 2022, released on October 17, 2022.
- [66] J. Liu, “Llamaindex, 11 2022,” https://github.com/jerryjliu/llama_index, 2022, accessed: 2022-11-01.
- [67] OpenAI, “New models and developer products announced at devday,” <https://openai.com/index/new-models-and-developer-products-announced-at-devday/>, Fecha de acceso: junio 2024.

- [68] A. Shoufan, “Exploring students’ perceptions of chatgpt: Thematic analysis and follow-up survey,” *IEEE Access*, vol. 11, pp. 38 805–38 818, 2023.
- [69] TAL Education Group, “Mathgpt,” <https://www.mathgpt.com>, Fecha de acceso: junio 2024, focused on math-related problem solving and lecturing for worldwide research institutions and math enthusiasts.
- [70] A. Lieb and T. Goel, “Student interaction with newtbot: An llm-as-tutor chatbot for secondary physics education,” in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–8.
- [71] H. Monteiro, “Chatting over course material: The role of retrieval augmented generation systems in enhancing academic chatbots.” 2024.
- [72] ChromaDB, “Chromadb, vector database,” accessed: February 27, 2024.
- [73] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou, “The faiss library,” 2024.
- [74] S. Malathi, S. Hemamalini, M. Ashwin, R. Benny *et al.*, “Knowledge navigator: Revolutionizing education through llms in generative ai,” *Full Length Article*, vol. 16, no. 1, pp. 209–09, 2024.
- [75] H. Modran, I. C. Bogdan, D. Ursuțiu, C. Samoila, and P. L. Modran, “Llm intelligent agent tutoring in higher education courses using a rag approach,” 2024.
- [76] A. Hsain and H. E. Housni, “Large language model-powered chatbots for internationalizing student support in higher education,” *arXiv preprint arXiv:2403.14702*, 2024.
- [77] J. Stewart, L. Redlin, and S. Watson, *Precálculo*, 6th ed. Cengage Learning Editores, SA, 2010.
- [78] Pinecone, “Pinecone documentation,” 2023. [Online]. Available: <https://docs.pinecone.io>

- [79] —, “Vector similarity documentation,” 2024, accessed: 2024-08-11. [Online]. Available: <https://www.pinecone.io/learn/vector-similarity/>