**VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY**

**UNIVERSITY OF INFORMATION TECHNOLOGY**

**ADVANCED PROGRAM IN INFORMATION SYSTEMS**

**PHAM THUY DUNG**

**DAU DINH QUANG ANH**

**THESIS**

**DEVELOPING A CHATBOT TO SUPPORT ADMISSION SERVICES AT THE UNIVERSITY OF INFORMATION TECHNOLOGY (UIT)**

**BACHELOR OF INFORMATION SYSTEMS**

**HO CHI MINH CITY, 2024**

**VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY**

**UNIVERSITY OF INFORMATION TECHNOLOGY**

**ADVANCED PROGRAM IN INFORMATION SYSTEMS**

**PHAM THUY DUNG - 20521214**

**DAU DINH QUANG ANH - 20521059**

**THESIS**

**DEVELOPING A CHATBOT TO SUPPORT ADMISSION SERVICES AT THE UNIVERSITY OF INFORMATION TECHNOLOGY (UIT)**

**BACHELOR OF INFORMATION SYSTEMS**

**THESIS ADVISOR**

**PROF. DR.DO PHUC**

**MS. NGUYEN THI KIM PHUNG**

**HO CHI MINH CITY, 2024**

# ACKNOWLEDGEMENT

The thesis was finalized at the University of Information Technology, Vietnam National University, Ho Chi Minh City. Over the composition of my graduation thesis, we received significant support from professors within the school and department in finishing the work.

We extend our heartfelt gratitude to Prof. Dr. Do Phuc and Ms. Nguyen Thi Kim Phung for their passionate guidance and for sharing their knowledge and experience with us during the preparation of this graduation thesis. We wish to express our gratitude to the educators of the Department of Information Systems at the University of Information Technology for imparting invaluable knowledge throughout four years of study at the institution.

We extend our gratitude to our family, friends, and the students of the CTTT-2020 class for their constant encouragement and help during the thesis writing process. We would like to express our gratitude to the respondents who participated in utilizing the Chatbot system to assist in the completion of this graduation thesis.

Nonetheless, the limitations of personal capacity always result in flaws in the research topic. We are looking for feedback from educators in the Department of Information Systems. Once again, we sincerely express our gratitude.

Ho Chi Minh City, 1st January 2025

Group of Authors

Pham Thuy Dung – Dau Dinh Quang Anh

**TABLE OF CONTENTS**

[ACKNOWLEDGEMENT I](#_Toc187210520)

[LIST OF FIGURES IV](#_Toc187210521)

[LIST OF TABLES V](#_Toc187210522)

[LIST OF ACRONYMS VI](#_Toc187210523)

[ABSTRACT VII](#_Toc187210524)

[CHAPTER 1: INTRODUCTION 1](#_Toc187210525)

[1.1 Background of the Study 1](#_Toc187210526)

[1.2 Problem Statement 2](#_Toc187210527)

[1.3 Objectives of the Study 3](#_Toc187210528)

[1.4 Research Questions 3](#_Toc187210529)

[1.5 Scope and Limitations 4](#_Toc187210530)

[1.6. Significance of the Study 5](#_Toc187210531)

[CHAPTER 2: LITERATURE REVIEW 5](#_Toc187210532)

[2.1 Overview of Chatbots in Education 5](#_Toc187210533)

[2.2 Large Language Models (LLMs) and Their Applications 7](#_Toc187210534)

[2.3 Retrieval-Augmented Generation (RAG) in AI Systems 8](#_Toc187210535)

[2.4 LangChain for Multi-Step Conversational Systems 9](#_Toc187210536)

[2.5 Ethical Considerations in Chatbot Development 9](#_Toc187210537)

[3.6 Gaps in Current Research and the Need for This Study 10](#_Toc187210538)

[CHAPTER 3: METHODOLOGY 11](#_Toc187210539)

[3.1 Research Design 11](#_Toc187210540)

[3.2 System Architecture Overview 12](#_Toc187210541)

[3.2.1 Large Language Models (LLMs) 12](#_Toc187210542)

[3.2.2 ChromaDB 13](#_Toc187210543)

[3.2.3 LangChain 14](#_Toc187210544)

[3.2.2 Retrieval-Augmented Generation (RAG) 15](#_Toc187210545)

[3.4 Evaluation Metrics 16](#_Toc187210546)

[CHAPTER 4: SYSTEM DESIGN 16](#_Toc187210547)

[CHAPTER 5: SYSTEM IMPLEMENTATION 17](#_Toc187210548)

[5.1 Data Collection 17](#_Toc187210549)

[5.2 Data Preprocessing 18](#_Toc187210550)

[5.3 TF-IDF (Term Frequency-Inverse Document Frequency) 18](#_Toc187210551)

[5.4 Sematic Chunking 19](#_Toc187210552)

[5.5 Sentence Embeddings Utilizing Pre-trained Models 19](#_Toc187210553)

[5.6 Data Storage in Chroma Database 19](#_Toc187210554)

[4.7 Vector Search 19](#_Toc187210555)

[5.8 Gemini API 20](#_Toc187210556)

[5.9 Hype Search 20](#_Toc187210557)

[REFERENCE 21](#_Toc187210558)

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ACRONYMS

abc

# ABSTRACT

The rapid progress of chatbot applications has transformed organizational interactions with users, especially in sectors such as education, healthcare, and customer service. This research investigates the creation of a chatbot designed to assist admission services at the University of Information Technology (UIT). By incorporating advanced technologies like Retrieval-Augmented Generation (RAG), LangChain, ChromaDB, and an extensive dataset of admission-related documents from 2023–2024, the chatbot provides accurate and contextually relevant responses to candidates for admission.  
  
This research emphasizes the critical privacy and security issues necessary for protecting sensitive user data and ensuring legal compliance. LangChain is utilized the chatbot's design for organizing multi-step query processing and ChromaDB . They efficiently manage document embeddings, enabling seamless retrieval of domain-specific information. RAG improves response accuracy by integrating recovered data with the generative capabilities of large language models (LLMs), which helps minimize hallucinations and ensure dependability.

This study illustrates the innovative capacity of chatbots powered by artificial intelligence in automating repetitive processes, enhancing user experience, and solving the constraints of real-time, domain-specific applications. This chatbot establishes a standard for implementing scalable and secure AI solutions in educational institutions.

# CHAPTER 1: INTRODUCTION

## 1.1 Background of the Study

Chatbots are AI-powered applications created to emulate human conversations, enabling smooth interactions between users and machines. They are essential in optimizing operations across several businesses by automating repetitive work, delivering immediate customer service, and improving user engagement. In e-commerce, chatbots function as virtual sales assistants, directing customers throughout their purchase experience and guaranteeing a flawless interaction.

On one hand, in the 1960s with ELIZA, the design and creation of chatbots started [43] . It is an initial program that emulates human conversation via basic pattern recognition. Over the decades, progress has resulted in more advanced Chatbots.  In natural language processing (NLP) and machine learning (ML), it can understand context and offer customized responses. In the field of education, chatbots utilize to assist with administrative tasks, deliver specialized learning experiences, and enhance communication between students and educational institutions.

On the other hand, universities often experience difficulties during admission periods, because of an increase of repeated questions from potential pupils and limited resources to manage them. As a result, these problems may result in delayed response times and a reduced experience for candidates. Implementing flexible approaches, like those powered by artificial intelligence chatbots, can successfully resolve these difficulties.

The University of Information Technology (UIT) faces particular obstacles during admissions, which include the management of varied questions from a technologically advanced applicant pool. It is essential that improving potential student engagement for UIT keeps its reputation and draws outstanding individuals. By building Chatbot with advanced technology, UIT will provide quick and detailed responses to candidates, enhancing their experience and optimizing the admissions process.

UIT has a history of employing technology to improve various parts of its operations, including the implementation of online learning platforms and the utilization of powerful IT infrastructure. The current technical infrastructure enables UIT to incorporate chatbot solutions into its admissions procedure, so underscoring its dedication to innovation and enhancing student involvement.

## 1.2 Problem Statement

The University of Information Technology (UIT) presently depends on a demanding but manual admissions support procedure. It is gradually burdened by a high amount of questions from interested applicants. Staff members frequently experience overflow from repetitive inquiries. This leads to delay in delivering essential information and restricting accessibility for applicants in need of prompt assistance. Inefficiencies may cause an inadequate user experience, it gives rise to discouraged qualified candidates from finishing their applications. Deploying an artificial intelligence-driven chatbot offers an approachable resolution to these issues. This system can manage a large number of inquiries at the same time, delivering immediate, precise answers to frequently asked issues and freeing human resources for greater complexity duties. Not only does this optimize the admissions process, but it also elevates the whole experience for potential students. In closing, building Chatbot with advanced technology would enable UIT to optimize its admissions support, lower staff workload, and deliver a more responsive at the same time, also augmenting the university's attractiveness and competitiveness in the higher education sector.

## 1.3 Objectives of the Study

The primary purpose of this study is to create an artificial intelligence (AI) chatbot to improve UIT's admission services by effectively and accurately responding to potential student inquiries. This requires the construction of a chatbot architecture that incorporates advanced technologies like Retrieval-Augmented Generation (RAG), ChromaDB, and LangChain, which are essential for developing intelligent chat systems. An important component of this advancement is the building of a large dataset of admission-related information so that ensures the chatbot transmits accurately and contextually relevant responses. Implementing the chatbot requires a comprehensive assessment of its performance, emphasizing measures such as accuracy, user happiness, and reaction time. It is equally important to consider potential ethical and privacy issues of chatbot implementation, including data privacy, transparency, and bias reduction, to guarantee responsible AI integration. Due to this, this research seeks to optimize UIT's admission process, enhance engagement with potential students, and preserve ethical standards in AI implementation.

## 1.4 Research Questions

The major research question for this study is: How could a powered by AI chatbot improve both the effectiveness and accessibility of UIT's admission services?. This research seeks to examine the potential of AI-driven solutions in managing administrative processes and enhancing user engagement within the university's admissions framework. Several secondary research questions have been considered to tackle this primary search:  
**1. What are the essential technologies that need to design and implement such a chatbot?**  
**Answer:** Identifying the fundamental technology elements is crucial to creat a functioning and efficient Chatbot. This consists of exploring advanced natural language processing (NLP) tools, machine learning algorithms, and database management systems. They help with a chatbot's accurate comprehension and response to user requests. This is best exemplified by platforms such as EduBot employing Microsoft's advanced technology, including Natural Language Processing and Cognitive Services, to provide intuitive and engaging user interactions [51].  
  
**2. How can the chatbot ensure accurate and relevant responses to inquiries regarding admissions?**  
**Answer:** It is essential for the chatbot to deliver accurate and relevant information to uphold the integrity of the admissions process. This entails assembling extensive datasets of admission-related information and deploying algorithms that enable the chatbot to provide contextually relevant responses. Additionally, ongoing upgrades and machine learning methodologies enable the chatbot to adjust to new facts and evolving admission policies.  
**3. What does ethical factors must be considered throughout the development and deployment of the chatbot?**  
**Answer:** The incorporation of AI in educational environments presents numerous ethical issues, such as data protection, transparency, and possible biases in AI systems. Therefore, resolving these concerns is crucial to ensure that the chatbot functions within ethical parameters and upholds user confidence. Research has underscored the necessity of integrating ethical considerations into AI design, stressing the values of justice, openness, and accountability [52].  
**4. What is the comparison of the chatbot's performance with existing systems regarding customer satisfaction and response efficiency?**  
**Answer:** Assessing the chatbot's efficacy requires juxtaposing its performance measures, including response time and user satisfaction, with those of current admission support systems. This evaluation ascertains the value contributed by the AI-driven chatbot and highlights opportunities for enhancement. This is evident in universities such as Purdue deployed chatbots, which managed 82% of initial interactions with potential students, demonstrates a significant improvement in effective response[53].

In conclusion, this study wants to clarify the creation, setup, and effect of an AI-powered chatbot on strengthening UIT's admission services through the exploration of these research issues.

## 1.5 Scope and Limitations

This study concentrates on improving the admission services of the University of Information Technology (UIT) by developing an AI-powered chatbot. The chatbot is designed to respond to questions related to course offerings, scholarships, and application schedules, mostly aimed at prospective pupils. The chatbot will be trained on official UIT data from the 2023–2024 academic years for providing accurate and context-aware responses. On the contrary, specific limits are recognized in this project. The chatbot's capabilities will be limited to inquiries regarding admissions and will not address basic academic or administrative questions. The performance is intrinsically reliant on the quality and completeness of the training dataset; any deficiencies or inconsistencies in the data may impact response precision. The chatbot will initially support English and/or Vietnamese, with the possibility of future multilingual capabilities under consideration. In addition to that, technological challenges like system downtime or increased latency in peak usage times may affect user experience. Consequently, the study aims to create realistic expectations and offer a focused framework to develop and operate the AI chatbot in UIT's admission services.

## 1.6. Significance of the Study

Integrating an AI-powered chatbot at the University of Information Technology (UIT) presents significant benefits beyond various aspects.

**For UIT:**

* Streamlining Admission Procedures: The chatbot can automate replies to frequent questions like course offers, scholarships, and application schedules. Therefore, it decreases the burden on office staff and enabling them to concentrate on more complicated tasks.
* Increasing Technological infrastructure : It can enhance university’s technological facilities and reputation.

**For Students and Their Parents:**

* Improving Accessibility: The chatbot guarantees continual access to admission information for domestic as well as international applicants.
* Improving User Experience: It provides prompt and precise responses to admission inquiries.

**In the Area of Educational Technology:**

* Building AI Integration: Enhancing the existing knowledge on AI-driven solutions in university, highlighting the practical application of large language models (LLMs), Retrieval-Augmented Generation (RAG), and other advanced tools in resolving real-world issues [54].
* Establishing a Benchmark for Innovation: The effective implementation of the chatbot at UIT can serve as a model for other institutions aiming to incorporate AI into their operations, emphasizing best practices and prospective advantages.

In summary, the setup and deployment of an AI-driven chatbot at UIT will further improve institutional efficiency, better the candidate experience, and provide significant insights to the wider field of educational technology.

# CHAPTER 2: LITERATURE REVIEW

## 2.1 Overview of Chatbots in Education

This area has been extensively explored by many authors, including Sangeetha, Kumar, Mondal, Smith, etc. Chatbots have gained important milestones in educational environments, providing novel ways to boost student learning and managerial efficiency. This overview of the literature analyzes the evolution, applications, and consequences of chatbots in education.

To begin with, the simultaneous use of artificial intelligence (AI) with machine learning (ML) has considerably upgraded chatbot functionalities, promoting more natural and contextual awareness dialogues. In the previous research in this field, chatbots were essentially rule-based, constraining their capacity to manage difficult queries. According to Chaimaa Bouafoud, Khalid Zine-Dine, and Abdellah Madani, the development of large language models (LLMs) has enabled the creation of advanced chatbots that can comprehend and produce human-like discussions, thereby improving their effectiveness in educational settings [1]. Also, chatbots fulfill multiple roles in education, such as personalized learning assistance, administrative support, and educational recommendation systems. Concurrently, chatbots deliver individual assistance to students by responding to specific requests, providing clarifications, and navigating them through educational materials. As a result, it is evident that all of them focused on interaction, which can improve student engagement and understanding (Ashok, Kumar Ramasamy, Snehitha, and Keerthi; 2021) [2]. What is more significant is that educational institutions utilize chatbots to handle their administrative tasks, including addressing commonly asked questions, assisting with course registrations, and broadcasting information on timetables and deadlines, which help them enhance operational efficiency (Khalid Oqaidi, Sarah Aouhassi, and Khalifa Mansouri; 2024) [3]. Next, from the paper of authors Paulo Cesar Ramos Pinho and Tiago Thompsen Primo (2023), Chatbots included recommender systems that support students in choosing courses, resources, and activities that are compatible with their interests and academic objectives, giving them a personalized educational experience [4]. Then, Researcher Cheng suggests that chatbots can enhance learning outcomes by increasing student motivation and engagement, which subsequently improves retention and comprehension of course content [5]. Chatbots provide immediate responses to student questions, facilitating prompt clarification of uncertainties and enhancing learning [6]. Notwithstanding its advantages, the deployment of chatbots in education faces challenges. A clear demonstration of this is ensuring the chatbot's capacity to precisely comprehend and address a diverse array of student inputs continues to be a considerable challenge highlighted by Maladzhi, Tsoeu, Mthombeni, and others [7]. Another one is concerned with data privacy, permission, and the risk of excessive dependence on AI systems, which requires meticulous examination from Maita, Saide, Putri, and Muwardi [8]. The ongoing advancement of AI technologies indicates a favorable future for chatbots in education. In the future, it is expected to be effective and useful for educational chatbots by improving natural language processing (NLP) and incorporating multimodal features [9].

As a consequence there of the successful integration of chatbots in educational environments presents a significant opportunity to enrich learning experiences and optimize administrative functions.

## 2.2 Large Language Models (LLMs) and Their Applications

This section will examine Large Language Models (LLMs), which have significantly advanced natural language processing (NLP) because they allow machines to understand and generate human-like text. Hence, this brief survey explores the frameworks, applications, and challenges related to LLMs, referencing significant papers from many professors in the field, such as Bouafoud, Ramasamy, Oqaidi, and others.

Initially, Large Language Models (LLMs) are fundamentally constructed on transformer architectures, and they can enable the processing of sequential input by self-attention mechanisms. As a result, this design allows models to identify complex linguistic patterns and relationships. Raiaan and associates contend that recent improvements have led to the establishment of models with billions of parameters. This inevitably leads to an increase in their ability to perform a wide range of NLP duties [10]. Then, there are several applications of Large Language Models (LLMs). By way of example, (LLMs) help summarize, translate, and answer questions in Excel [11]. In the case of education, LLMs apply to intelligent tutoring systems, automated grading, and personalized learning experiences [12]. Also, LLMs play a role in threat detection, automated code analysis, and the formulation of security policies in cybersecurity [13]. Finally, LLMs assist in code generation, debugging, and documentation [14]. Nonetheless, LLMs face numerous challenges, especially in the areas of training and deployment, because of needing massive computational resources and memory. As a consequence, smaller enterprises face accessibility issues [15]. The implementation of LLMs presents ethical dilemmas, such as biases in produced content, data privacy issues, and the risk of misuse in creating deceptive information [16]. Comprehending the decision-making mechanisms of LLMs is intricate, complicating the interpretation and trustworthiness of their outputs [17]. The effort of current research is focused on solving these problems by improving training, models, and ethical standards for the implementation of LLMs. Moreover, the integration of multimodal data and the improvement of understanding of the context are critical focal points for broadening the use of LLMs [18].

In conclusion, Large Language Models represent a substantial advancement in Natural Language Processing, providing many applications across numerous fields.

## 2.3 Retrieval-Augmented Generation (RAG) in AI Systems

It is worth mentioning that Retrieval-Augmented Generation (RAG) has become a crucial model in artificial intelligence (AI), improving the LLMs functions by adding external documents into the generation process. This study reviews the architecture, applications, and problems of RAG in AI systems.

First and foremost, RAG generate models for more accurate and contextually relevant result by integrates retrieval architectures. The design often includes a retriever of external databases and then generates them to respond. Therefore, this connection enables AI systems to obtain current knowledge beyond their training material, reducing problems[19].RAG has been utilized in diverse fields, including interactive AI systems, systematic literature reviews, and knowledge-intensive applications. For example, RAG improves interactive AI by allowing systems to deliver more precise and contextually relevant replies [20]. In academic research, RAG automates literature review generation by obtaining and summarizing pertinent papers[21]. RAG reduces the constraints of LLMs in managing knowledge-intensive tasks by integrating external information [22]. Conversely, RAG presents numerous problems, including retriever efficacy, integration intricacy, and assessment measures. The efficacy of RAG systems is largely contingent upon the retriever's capacity to accurately detect and recover pertinent information. Improving retriever performance is essential for the overall efficacy of RAG applications [23]. Fully combining retrieved information with generative models necessitates advanced procedures to enable coherence and relevance in the produced outputs [24]. The evaluation of frameworks is necessary for RAG systems that take into account both retrieval precision and generation quality [25]. To these challenges, current studies aim to develop retrieves, improve techniques, and establish evaluation benchmarks. Additionally, the use of RAG is a key area to explore and adapt to various domains in multimodal contexts [25].

To conclude, Retrieval-Augmented Generation represents a notable progression because it lets systems use outside information documents to generate outputs that are more accurate and relevant to the situation.

## 2.4 LangChain for Multi-Step Conversational Systems

LangChain is a framework intended to streamline the creation of applications utilizing large language models (LLMs), specifically focusing on multi-step conversational systems. This literature review analyzes the function of LangChain inside these systems, citing significant works and resources in the domain.

LangChain offers a structured approach to building applications with LLMs . It providing tools for timely management, linking multiple LLM calls, and connecting with external data sources. The modular architecture allows developers to construct sophisticated conversational agents proficient at managing multi-turn interactions efficiently. LangChain helps create customizer open-source GPT-based chatbots for enterprises. These chatbots facilitate responsive, context-aware, and tailored interactions to enhance user experience and operational efficiency [28]. Since Chatbots engage with PDF documents, enhance information retrieval and user interaction, researchers have utilized LangChain to develop chatbots . This application highlights LangChain's proficiency in managing document-centric conversational activities [29]. LangChain has been utilized to create apps that produce test questions from PDF documents, demonstrating its effectiveness in educational technology and automated content generation [30].

LangChain functions as an essential framework for constructing multi-step conversational systems, providing tools and integrations that augment the functionalities of LLM-driven applications. Its use throughout diverse fields highlights its adaptability and efficacy in developing intricate conversational agents.

## 2.5 Ethical Considerations in Chatbot Development

The advancement of chatbots has markedly altered human-computer interactions, providing efficiency and accessibility in multiple sectors. This innovation presents numerous ethical considerations that developers must confront to guarantee responsible and equitable utilization. This literature review examines significant ethical issues in chatbot development, citing relevant papers and articles.

Chatbots trained on extensive datasets may unintentionally acquire and propagate biases inherent in the data, resulting in discriminatory conduct. A study indicated that AI chatbots may have biases akin to those of humans, prompting bias training to address these concerns [31]. It is concerned for data privacy and security since the interaction between users and chatbots frequently involves sensitive information during transmissions. Therefore, research suggests that employees may accidentally provide valuable firm information to AI chatbots, highlighting the necessity for stringent data governance procedures [32]. Determining responsibility for a chatbot's actions, especially when it causes harm or communicates false information, is complex. A article studying the ethical consequences of ChatGPT emphasizes the challenges in establishing accountability and governance for AI-generated material. [33]. It is essential for chatbots to deliver precise and dependable information, especially in critical fields such as healthcare and banking. The quality of AI-generated content may significantly affect consumer trust and decision-making [34]. Integrating ethical considerations throughout the chatbot development process is essential for reducing potential negative impacts. Proposed rules and frameworks seek to assist developers in the development of ethically responsible AI systems. [35]. The anthropomorphic design of chatbots may cause users to develop emotional ties, potentially leading to excessive dependence on AI for emotional assistance. Recent advances indicate that users may get emotionally attached to chatbots, prompting ethical concerns regarding user well-being [36].

It is essential to address ethical considerations in chatbot development to guarantee that these AI systems benefit society positively and equitably. Developers must participate in ongoing ethical contemplation, apply bias mitigation techniques, defend data privacy, ensure accountability, and sustain system reliability to cultivate trust and integrity in chatbot interactions.

## 3.6 Gaps in Current Research and the Need for This Study

The swift progression of chatbot technology has profoundly altered human-computer interactions in multiple fields. Notwithstanding significant advancements, some research deficiencies remain, requiring additional inquiry to improve chatbot development and implementation.

The factors affecting user acceptance and adoption of AI-based chatbots are yet inadequately examined. Although certain research have explored factors influencing chatbot adoption, complete models that incorporate both positive and negative predictors are few. The integrated chatbot acceptance-avoidance model overcomes this gap by examining the dual factors that influence user decisions [37]. The methods for assessing chatbot usability are not standardized, resulting in inconsistent evaluation standards. A secondary analysis of experimentation regarding chatbot usability evaluation underscores the necessity for systematic methodologies to adequately measure usability [38].There is a lack of study examining the ethical implications and privacy concerns, related to chatbot interactions, especially in sensitive areas. In mental health public services , a study on AI chatbots in mental health public services reveals under-researched areas, which emphasizes the need for ethical frameworks [39]. The use of chatbots in enabling digital business transformation is a burgeoning field that necessitates additional investigation. A systematic study highlights the necessity for additional research on the appropriate integration of chatbots into business processes to facilitate transformation [40]. Despite the application of modern deep learning-based NLP approaches in chatbot development, obstacles remain in attaining human-like interactions. A comprehensive assessment examines the present status and prospective developments, highlighting the need for continued research to improve conversational abilities [41]. An extensive examination of AI and deep learning-based chatbots identifies application patterns while highlighting the necessity for research aimed at enhancing chatbot performance across various applications [42].

Solving these research gaps is crucial to the development of chatbot technology and ensuring its ethical and successful application in a variety of industries. Future research should focus on developing standardized methods for evaluating usability, examining ethical frameworks, understanding factors that determine user acceptability, and increasing integration techniques in the transformation of digital companies.

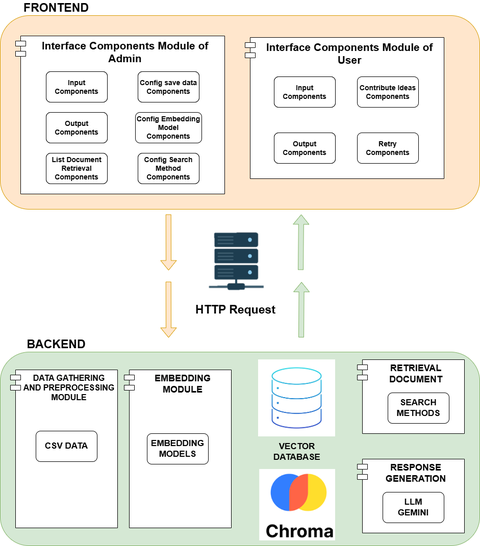
# CHAPTER 3: METHODOLOGY

## 3.1 Research Design

The research would like to create a chatbot system that efficiently assists students and student's parents in retrieving and figuring out admission materials related to the University of Information Technology (UIT). The chatbot , which offer accurate, contextually relevant, and user-centric responses, utilizes machine learning (ML) and natural language processing (NLP) methodologies. The technology combines Retrieval-Augmented Generation (RAG) with Large Language Models (LLMs) to merge document retrieval functionalities with AI-generated content. By integrating structured information storage with advanced reasoning skills, the chatbot connects raw data with conversational knowledge. To provide accurate and contextually relevant responses based on organized admission data for the 2023–2024 school year, while allowing for adaptability for future improvements. The chatbot is explicitly designed to respond to inquiries concerning UIT admissions, that includes academic programs, scholarships, admission policies, and student life in general. The technology is scalable to incorporate datasets in the future.

## 3.2 System Architecture Overview

The system architecture observes to a flexibility, adaptability, and productive design.  It generates that ensures smooth integration and optimal performance.



**Important Elements:**

Large Language Models (LLMs)

* Makes use of models like Gemini to provide natural, natural reactions.
* Processes user inquiries, improves retrieval results, and produces contextually appropriate answers.

ChromaDB

* A vector database utilized for the storage of embeddings generated by admission documents.
* Facilitates rapid and effective similarity searches that match with user inquiries.

LangChain

* Coordinates workflows among LLMs, document retrieval, and chunking methods.
* Facilitates an organized framework for controlling API requests and preservation of modularity.

Retrieval-Augmented Generation (RAG):

* Integrates document retrieval with AI-generated material to deliver factually accurate responses.
* Extracts relevant information chunks from a vector database and improves them with text produced by large language models (LLMs).

### 3.2.1 Large Language Models (LLMs)

Large Language Models (LLMs), such as Google's Gemini, are advanced artificial intelligence (AI) systems to comprehend and produce natural text.

Objectives:

* Produce Natural Reactions: They generate coherent and contextually relevant responses to user inputs, as a result, improving the conversational experience.
* Understand User Inquiries: They explore and deal with user questions, allowing correct and relevant answers.
* Increase Retrieved Information: LLMs improve and enrich information collected from databases.

The Way They Operate [44]:

* Training on Varied Data: LLMs are trained on comprehensive datasets, which contain text, code, audio, photos, and video.
* Multimodal Processing: Models like Gemini can dynamically analyze several types of data, allowing them to address complicated inquiries.
* Integration with Retrieval Systems: In a Retrieval-Augmented Generation (RAG) framework, large language models (LLMs) integrate with RAG systems to obtain specific data, which utilize to provide intelligent responses.

Significance of Their Importance:

* Enhanced User Experience: By producing natural replies, LLMs render interactions more intuitive and fascinating for consumers.
* Improved Accuracy: Their capacity to analyze and comprehend intricate inquiries outcomes in more precise and contextually appropriate responses.
* Multipurpose: LLMs are applicable to many tasks like as language translation, summarization, and question answering, rendering them valuable in many different situations.

In summary, LLMs like as Gemini are integral of current chatbot systems by comprehending user inquiries, retrieving and augmenting information, and producing human responses.

### 3.2.2 ChromaDB

ChromaDB is an open-source vector database designed for the storage and management of high-dimensional embeddings, particularly for artificial intelligence (AI) and machine learning (ML) applications. It facilitates efficient similarity searches by permitting the storage of vector representations of data, such as text embeddings, and enhances fast retrieval in response to user inquiries. ChromaDB is a specialized database that manages vector embeddings—numerical representations of data that convey semantic information. ChromaDB enables similarity searches by storing embeddings to locate data points closely related to a specific query inside a high-dimensional environment.

The purpose of ChromaDB [46]:

A diagram of a diagram of a document

Description automatically generated

* Data Ingestion: ChromaDB processes data inputs, including text documents, and transforms them into vector embeddings utilizing models such as Sentence Transformers. These embeddings embody the semantic core of the data.
* Storage: The created embeddings are retained in ChromaDB, along with relevant metadata.This structure enables effective organizing and retrieval.
* Similarity Search: After receiving a query, ChromaDB transforms it into an embedding. Also it does a similarity search to identify stored embeddings that closely match the query. The technique enables the retrieval of relevant information using semantic similarity rather than exact keyword matching.

The benefit of employing ChromaDB:

* Efficiency: ChromaDB is designed for the management of large volumes of vector data, enabling fast similarity searches and data retrieval.
* Scalability: It supports several storage backends, making it suitable for both local and large-scale applications.
* Integration: ChromaDB seamlessly integrates with prominent embedding models and frameworks, facilitating its inclusion in diverse AI and machine learning (ML) workflows.

In summary, ChromaDB serves as a useful tool for managing and querying of vector embeddings, vital to applications requiring swift similarity searches and semantic data retrieval.

### 3.2.3 LangChain

LangChain is a comprehensive framework intended to facilitate the creation of applications utilizing Large Language Models (LLMs). It provides a systematic framework that coordinates processes across LLMs, document retrieval systems, and chunking methods, providing seamless integration and modularity in intricate LLM applications. LangChain functions as a framework for managing large language models (LLMs), assisting developers to create generative AI applications and retrieval-augmented generation (RAG) workflows. It offers the essential framework, instruments, and elements to streamline complex LLM procedures, enabling the creation of advanced AI-driven solutions.

The process of LangChain:

LangChain functions by orchestrating the interactions among many components necessary to LLM applications:

* Document Loaders: These elements retrieve data from many sources, facilitating its preparation for processing.
* Text Splitters: They partition extensive documents into manageable segments, which is crucial for effective processing and retrieval.
* Vector Stores: These repositories manage the storage and indexing of document embeddings, facilitating rapid similarity searches.
* Retrievers: They retrieve relevant document segments in response to user inquiries, ensuring the information supplied is applicable.
* LLMs: The fundamental models that provide human-like responses, process user inquiries, and refine collected information to yield contextually relevant answers.

LangChain coordinates various components to provide good communication inside the workflow, preserving modularity and scalability.

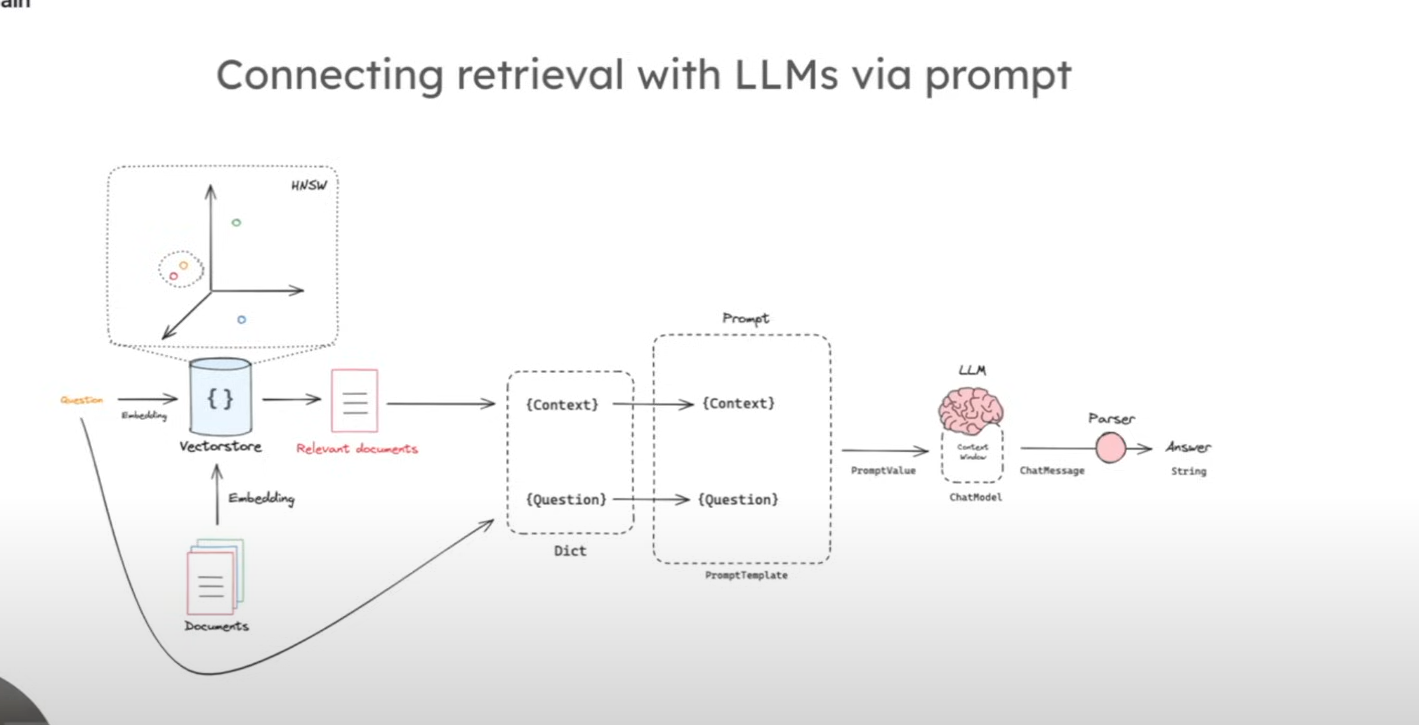
The Benefits of Utilizing LangChain:

The implementation of LangChain in LLM applications presents numerous benefits:

* Streamlined Development: LangChain simplifies the intricacies of managing several components, enabling developers to concentrate on functionality rather than integration challenges.
* Augmented Modularity: Its organized framework facilitates modularity, allowing for the seamless updating or replacement of individual components without affecting the entire system.
* Scalability: LangChain's architecture facilitates scalability, allowing for the expansion of applications and the incorporation of supplementary functionality as required.
* Efficiency: Through proficient workflow management, LangChain augments the efficiency of LLM applications, resulting in expedited response times and enhanced user experiences.

In summary, LangChain offers a robust framework that coordinates the diverse elements of LLM applications, enabling effective development, integration, and scalability. Its systematic methodology guarantees the seamless management of workflows, facilitating the development of advanced AI-driven solutions.

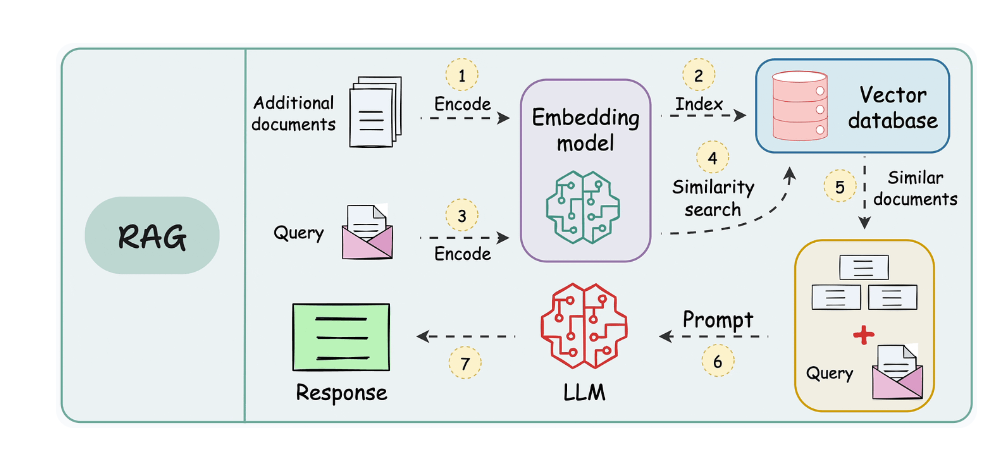
### 3.2.2 Retrieval-Augmented Generation (RAG)



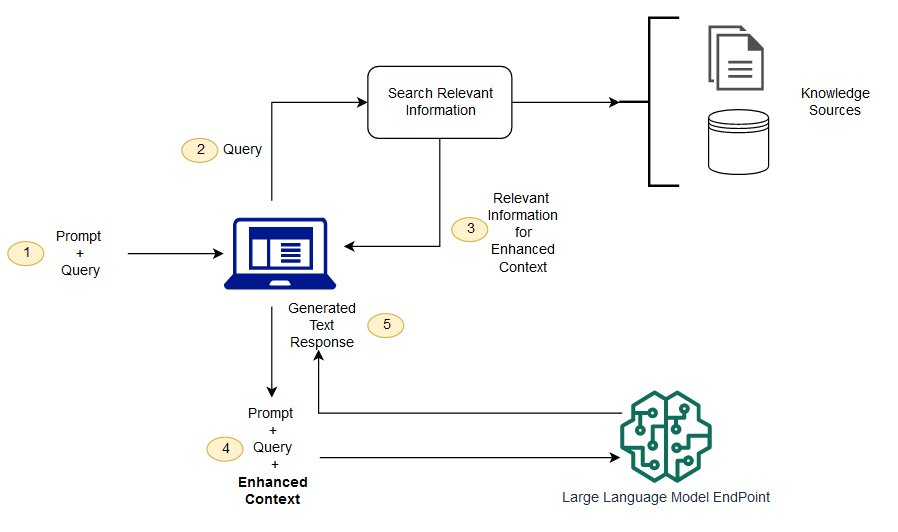
[Learn RAG From Scratch – Python AI Tutorial from a LangChain Engineer](https://www.youtube.com/watch?v=sVcwVQRHIc8)

Retrieval-Augmented Generation (RAG) is an AI system that improves Large Language Models (LLMs) by incorporating external data sources into the reply generating process. RAG integrates the generative potential of large language models (LLMs) with information retrieval technologies. It utilizes relevant information from external documents to enhance and refine the AI's responses.

The process of RAG [45]:



[Bi-encoders and Cross-encoders for Sentence Pair Similarity Scoring – Part 1](https://www.dailydoseofds.com/bi-encoders-and-cross-encoders-for-sentence-pair-similarity-scoring-part-1/)

****

The RAG process contains several key phases:

* Retrieval: After receiving a user query, the system finds external data sources to find information relevant to the query.
* Augmentation: The acquired information is integrated with the initial user question, offering additional context.
* Generation: The LLM analyzes the enhanced input to produce a response which combines its internal knowledge with the acquired external information.

The benefits for utilizing RAG:

* Increased Accuracy: RAG minimizes AI-generated hallucinations, resulting in accurate results.
* Domain-Specific Knowledge: The implementation of RAG enables LLMs to access certain knowledge not included in the training data, hence enhancing their effectiveness.
* Cost-Effectiveness: RAG offers a more efficient approach that raises AI capabilities.

In summary, RAG improves the performance of LLM outputs, resulting in greater accuracy, contextual relevance, and timeliness.

## 3.4 Evaluation Metrics

The ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metrics serve as critical tools to evaluate the quality of generated text in natural language processing (NLP) tasks. They assess the overlap between the generated text and reference texts to assess how well the output matches human-created content.

Types of ROUGE [50]:

* ROUGE-1: Determine the overlap of unigrams (individual words) between the generated and reference texts, thus giving a metric of word-level precision.
* ROUGE-2: Evaluate the intersection of bigrams (two-word sequences), supplying insight into the model's capacity to understand the contextual links among words.
* ROUGE-L: Measures the longest common subsequence (LCS) for the generated and reference texts, highlighting the model's capability to keep the structure and coherence of the reference text.

Advantages:

* Comprehensive Evaluation: Applying ROUGE measures enables the project to objectively evaluate the efficacy of text generation models.
* Model Comparison: ROUGE scores enable the evaluation of several models or configurations.
* Performance Monitoring: Allows the tracking of improvement over time.

Integrating ROUGE measures into the project's evaluation framework ensures an robust evaluation of text generation quality.

# CHAPTER 4: SYSTEM DESIGN

4.1 System Design Overview

A diagram of a data flow

Description automatically generated

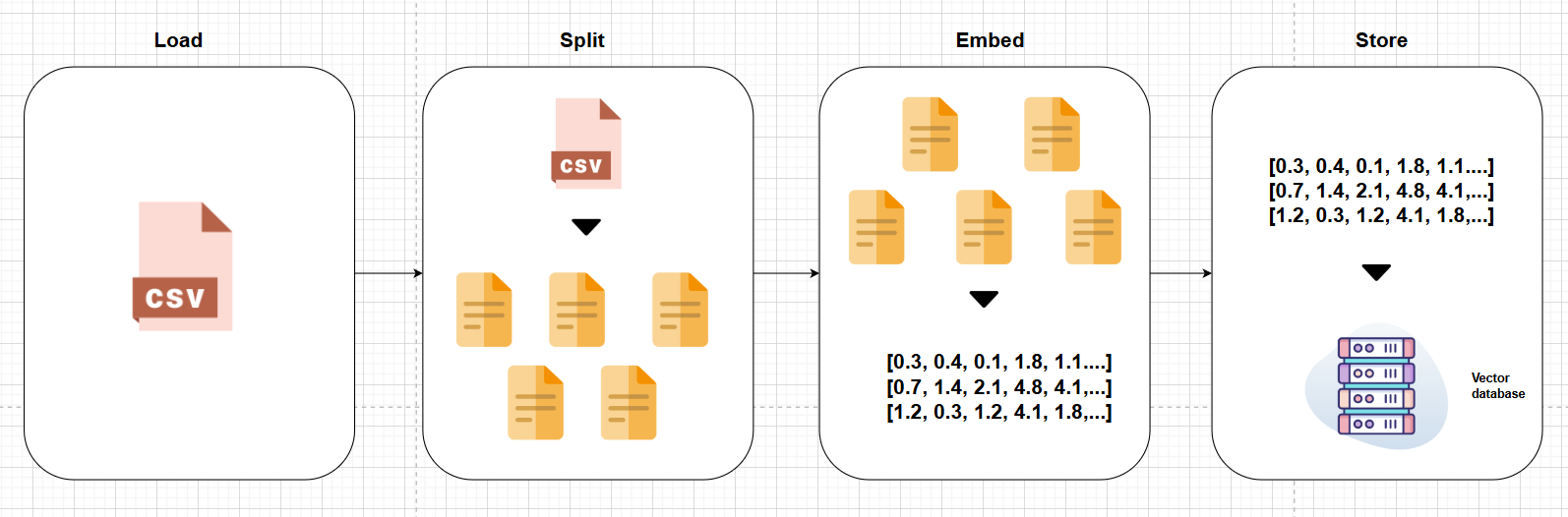
[A Crash Course on Building RAG Systems – Part 1 (With Implementation)](https://www.dailydoseofds.com/a-crash-course-on-building-rag-systems-part-1-with-implementations/)

A diagram of a diagram

Description automatically generated

A screenshot of a computer screen

Description automatically generated



A yellow and black text

Description automatically generated

[(294) SBERT (Sentence Transformers) is not BERT Sentence Embedding: Intro & Tutorial (#sbert Ep 37) - YouTube](https://www.youtube.com/watch?v=lVqwznaVi78)

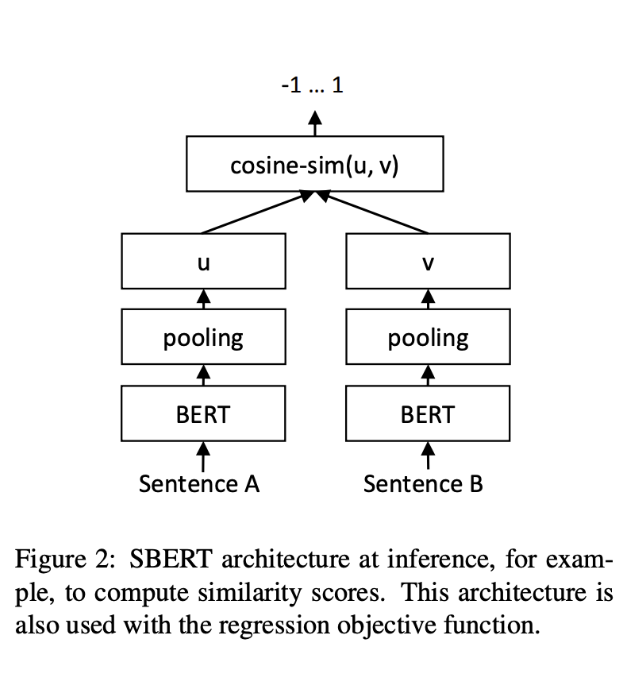
SBERT:

Examples and Use Cases

SBERT has found applications across various domains due to its ability to generate high-quality sentence embeddings. Some notable use cases include:

1. **Semantic Textual Similarity (STS):** SBERT is widely used to measure the similarity between sentences, which is crucial for tasks like paraphrase detection and duplicate question identification.
2. [**Clustering**](https://aijobs.net/insights/clustering-explained/)**:** By converting sentences into embeddings, SBERT facilitates clustering tasks, enabling the grouping of semantically similar sentences or documents.
3. **Information Retrieval:** SBERT enhances search engines by improving the relevance of search results through better understanding of query semantics.
4. **Question Answering Systems:** SBERT can be used to match user queries with relevant answers in a knowledge base, improving the accuracy of QA systems.
5. [**Chatbots**](https://aijobs.net/insights/chatbots-explained/)**and Conversational AI:** SBERT helps in understanding user intents and generating contextually appropriate responses.

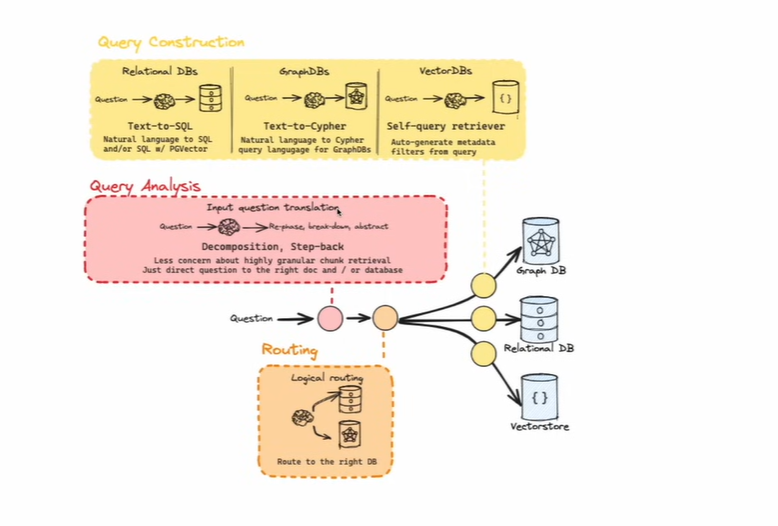
[SBERT Explained | aijobs.net](https://aijobs.net/insights/sbert-explained/?utm_source=chatgpt.com)



**Check the code, sbert has fine-turning or not?**

**Compare SBERT with BERT and RoBERTa.**

**Compare RAG with LLM not using RAG**

****

Sentence-BERT (SBERT) is an extension of the BERT architecture designed to generate semantically meaningful sentence embeddings, facilitating efficient comparison and clustering of sentences.

Definition: SBERT modifies BERT by employing a Siamese network structure to produce fixed-size sentence embeddings. This enables rapid computation of sentence similarities using measures like cosine similarity.

[AI Jobs](https://aijobs.net/insights/sbert-explained/?utm_source=chatgpt.com)

Key Features:

* Efficiency: SBERT significantly reduces computation time for tasks like finding similar sentence pairs, decreasing it from hours to mere seconds compared to traditional BERT models.

[Restack](https://www.restack.io/p/embeddings-answer-sentence-embeddings-benchmark-sbert-cat-ai?utm_source=chatgpt.com)

* Accuracy: It maintains high accuracy levels on various semantic textual similarity tasks, comparable to BERT and RoBERTa.

[Restack](https://www.restack.io/p/embeddings-answer-sentence-embeddings-benchmark-sbert-cat-ai?utm_source=chatgpt.com)

* Versatility: Applicable to a range of tasks, including semantic textual similarity, clustering, information retrieval, and paraphrase mining.

[SBERT](https://www.sbert.net/examples/applications/?utm_source=chatgpt.com)

How It Works:

1. Siamese Network Structure: SBERT uses two identical BERT models with shared weights to process sentence pairs simultaneously.

[Dair](https://dair.ai/posts/TLDR_SentenceBERT/?utm_source=chatgpt.com)

1. Pooling Layer: A pooling layer (e.g., mean pooling) converts variable-length sentence outputs into fixed-size embeddings.

[Dair](https://dair.ai/posts/TLDR_SentenceBERT/?utm_source=chatgpt.com)

1. Training Objective: Fine-tuned on datasets like SNLI and MultiNLI using classification objectives to predict relationships between sentence pairs (e.g., entailment, contradiction).

[Dair](https://dair.ai/posts/TLDR_SentenceBERT/?utm_source=chatgpt.com)

1. Similarity Computation: Once embeddings are generated, similarity between sentences can be efficiently computed using cosine similarity.

[SBERT](https://www.sbert.net/docs/quickstart.html?utm_source=chatgpt.com)

Benefits:

* Reduced Computational Overhead: Enables quick similarity computations without the need for extensive pairwise comparisons required by traditional BERT models.

[Restack](https://www.restack.io/p/embeddings-answer-sentence-embeddings-benchmark-sbert-cat-ai?utm_source=chatgpt.com)

* Improved Performance: Outperforms other sentence embedding methods in tasks like semantic textual similarity and transfer learning.

[Restack](https://www.restack.io/p/embeddings-answer-sentence-embeddings-benchmark-sbert-cat-ai?utm_source=chatgpt.com)

* Flexibility: Can be fine-tuned for specific tasks, enhancing performance in domain-specific applications.

[SBERT](https://sbert.net/examples/?utm_source=chatgpt.com)

Drawbacks:

* Resource Intensive Training: Fine-tuning SBERT requires substantial computational resources and large labeled datasets.
* Potential Domain Limitations: Without fine-tuning, SBERT may not perform optimally on domain-specific tasks.

**A diagram of a process

Description automatically generated**

BERT

In BERT-based models, **sentence embeddings** are derived from token embeddings using various **pooling strategies**. Two prevalent methods are **mean pooling** and utilizing the **[CLS] token**.

**1. [CLS] Token Pooling:**

* **What It Is:** BERT introduces a special token, [CLS] (classification token), at the beginning of every input sequence. The final hidden state corresponding to this token is often used as a representation of the entire sentence.
* **How It Works:**
  + Input tokens, including [CLS], are processed through BERT's layers.
  + The output embedding of the [CLS] token is extracted.
  + This embedding serves as the sentence representation for downstream tasks.
* **Benefits:**
  + **Simplicity:** Directly provides a fixed-size vector for the sentence.
  + **Design Intent:** [CLS] is specifically designed to capture holistic sentence information, especially for classification tasks.

**2. Mean Pooling:**

* **What It Is:** Mean pooling computes the average of all token embeddings in a sentence to produce a single fixed-size vector.
* **How It Works:**
  + Process the input tokens through BERT to obtain their embeddings.
  + Compute the mean of these embeddings across the sequence length.
  + The resulting vector represents the entire sentence.
* **Benefits:**
  + **Robustness:** Aggregates information from all tokens, potentially capturing nuanced meanings.
  + **Empirical Performance:** Studies have shown mean pooling can outperform [CLS] token embeddings in certain tasks.

[Sijunhe](https://sijunhe.github.io/2022/02/19/sentence-embeddings.html?utm_source=chatgpt.com)

**Choosing Between [CLS] Token and Mean Pooling:**

* The optimal pooling strategy can vary based on the specific task and dataset.
* Empirical evaluations suggest that mean pooling may offer advantages in capturing semantic nuances for certain applications.

[Sijunhe](https://sijunhe.github.io/2022/02/19/sentence-embeddings.html?utm_source=chatgpt.com)

* It's advisable to experiment with both methods to determine which yields better performance for your particular use case.

In summary, both [CLS] token pooling and mean pooling are effective strategies for deriving sentence embeddings from BERT. The choice between them should be guided by the specific requirements and characteristics of the task at hand.

BERT MORE

In the **Sentence-BERT (SBERT)** architecture, **BERT** serves as the foundational model responsible for encoding individual sentences into dense vector representations, commonly referred to as embeddings.

**Purpose of BERT in SBERT:**

1. **Semantic Encoding:**
   * BERT captures the contextual meaning of words within a sentence by considering both preceding and succeeding words, resulting in rich, bidirectional representations.
   * In SBERT, this capability is leveraged to generate embeddings that encapsulate the semantic essence of entire sentences.
2. **Facilitating Sentence-Level Tasks:**
   * While traditional BERT models excel at token-level tasks, SBERT adapts BERT for sentence-level applications by producing fixed-size sentence embeddings.
   * These embeddings can be efficiently compared using similarity measures like cosine similarity, enabling tasks such as semantic textual similarity, clustering, and information retrieval.

**How BERT Functions within SBERT:**

1. **Siamese Network Structure:**
   * SBERT employs a Siamese network architecture, where two identical BERT models (sharing the same weights) process input sentence pairs simultaneously.
   * Each BERT model encodes one sentence, producing contextualized token embeddings.
2. **Pooling Layer:**
   * To derive a fixed-size sentence embedding from the variable-length token embeddings, SBERT applies a pooling operation.
   * Common pooling strategies include mean pooling, max pooling, or using the embedding of the [CLS] token.
3. **Similarity Computation:**
   * The resulting sentence embeddings can be compared using similarity measures like cosine similarity, facilitating tasks such as semantic textual similarity, clustering, and information retrieval.

**Benefits of Using BERT in SBERT:**

* **Enhanced Semantic Understanding:**
  + BERT's deep bidirectional architecture allows SBERT to capture nuanced semantic relationships between sentences.
* **Improved Performance:**
  + By fine-tuning BERT within the SBERT framework, the model achieves state-of-the-art results on various sentence-level tasks.

**Meaning Pooling**

In **Sentence-BERT (SBERT)**, **mean pooling** is the default strategy for generating fixed-size sentence embeddings from token-level embeddings produced by the BERT model.

**How Mean Pooling Works:**

1. **Token Embedding Generation:**
   * Each input sentence is tokenized and passed through the BERT model, resulting in contextualized embeddings for each token.
2. **Mean Pooling Operation:**
   * The embeddings of all tokens in the sentence are averaged to produce a single fixed-size vector representing the entire sentence.
   * This involves summing the token embeddings and dividing by the number of tokens, effectively capturing the overall semantic information of the sentence.

**Benefits of Mean Pooling:**

* **Simplicity:** Mean pooling is straightforward to implement and computationally efficient.
* **Effective Semantic Representation:** By averaging token embeddings, mean pooling captures the general meaning of a sentence, making it suitable for various tasks like semantic similarity and clustering.
* **Empirical Performance:** Research has shown that mean pooling often outperforms other pooling strategies, such as using the [CLS] token or max pooling, in tasks involving sentence embeddings.

[arXiv](https://arxiv.org/pdf/1908.10084?utm_source=chatgpt.com)

A screenshot of a computer

Description automatically generated

[Sentence Embeddings and Similarity - Sijun He's Unsupervised Learning](https://sijunhe.github.io/2022/02/19/sentence-embeddings.html?utm_source=chatgpt.com)

# CHAPTER 5: SYSTEM IMPLEMENTATION

## 5.1 Data Collection

Creating a chatbot to assist admission services at the University of Information Technology (UIT) demands the preparation of an impressive dataset. This section includes the collection of relevant data and the implementation of high cleaning and preprocessing methods. Therefore, chatbot provides precise and reliability information.

Sources of Data (UIT Admission Documents for 2023–2024):

The collection comprises official admission documents from UIT for the academic years 2023 and 2024. These include:

* Admission Information: Detailed information concerning admission criteria, application procedures, and essential dates.
* Program Brochures: Descriptions of undergraduate and postgraduate programs, including curricula and career options.
* Scholarship Details: Details concerning available scholarships, eligibility criteria, and application processes.
* Common Inquiries (CIs): Standard questions from prospective students and official responses.

These materials are sourced from UIT's official website and verified publications to ensure accuracy and relevance.

## 5.2 Data Preprocessing

Text preprocessing is a crucial step in Natural Language Processing (NLP) to prepare raw text data for analysis. The aim of preprocessing is to clean and standardize the text to reduce noise and irrelevant information.

Common techniques include:

* Lowercasing: Converts all characters to lowercase to avoid case-sensitive distinctions between words like “Trường” and “trường” This is essential to reduce the vocabulary size and improve consistency.
* Removing Special Characters: Because symbols and punctuation marks (like @, #) often have no semantic significance in a sentence, they are discarded. Additionally, URLs and HTML tags are also removed.
* Removing Extra Whitespaces: Extra spaces are removed to prevent issues during tokenization and parsing.

|  |
| --- |
| import re  import unicodedata  import string  import re  def remove\_special\_characters(text):  text = re.sub(r'<.\*?>', ' ', text)  text = re.sub(r'http[s]?://(?:[a-zA-Z]|[0-9]|[$-\_@.&+]|[!\*\\(\\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+', ' ', text)  text = re.sub(r'[^\w\s.!?@]', ' ', text)  return text  def lowercase(text):  return text.lower()  def remove\_extra\_whitespaces(text):  text = text.strip()  text = re.sub(r'\s+', ' ', text)  return text  def preprocess\_text(text):  text = lowercase(text)  text = remove\_special\_characters(text)  text = remove\_extra\_whitespaces(text)  return text  def remove\_duplicate\_rows(df, column\_name):  df.drop\_duplicates(subset=column\_name, keep='first', inplace=True)  return df |

## 5.3 TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF is the product of these two measures to be utilized  to identify important words that distinguish documents. A higher TF-IDF score indicates that the term is both frequent in a particular document and seldom in the corpus.

* Term Frequency (TF): measures how often a term appears in a document. The assumption is that terms occurring frequently in a document are more likely to be important.
* Inverse Document Frequency (IDF): measures the importance of a term across all documents in a corpus. Words that occur frequently across many documents (like "the" or "and") are less important, while terms that occur in fewer documents are more meaningful.

Where:

|  |
| --- |
| def embed\_function(self, sentences):    if self.embedding\_type == "tfidf":  vectorizer = TfidfVectorizer().fit\_transform(sentences)  return vectorizer.toarray()  else:  raise ValueError("Unsupported embedding type") |

## 5.4 Sematic Chunking

**Chunking:** This represents the process of partitioning text into smaller, semantically significant portions (chunks). It utilizes the organization of information for management and analysis. In the semantic chunking, sentences or phrases are categorized according to their meaning rather than their structure.

**Cosine Similarity:** Cosine similarity quantifies the cosine of the angle formed by two vectors. This metric is frequently employed in NLP to determine the similarity between two textual representations.

The equation for cosine similarity is:

Where A and B are vectors representing the text, and and are their magnitudes.

**Thresholding:** Sentences are classified into segments if their cosine similarity exceeds a specified threshold. The threshold denotes the required similarity for phrases to be categorized together.

|  |
| --- |
| from .base\_chunker import BaseChunke def split\_text(self, text):  sentences = nltk.sent\_tokenize(text)  sentences = [item for item in sentences if item and item.strip()]  if not len(sentences):  return []  vectors = self.embed\_function(sentences)  similarities = cosine\_similarity(vectors)  chunks = [[sentences[0]]]  for i in range(1, len(sentences)):  sim\_score = similarities[i-1, i]  if sim\_score >= self.threshold:  chunks[-1].append(sentences[i])  else:  chunks.append([sentences[i]])  *# Join the sentences in each chunk to form coherent paragraphs*  return [' '.join(chunk) for chunk in chunks] |

## 5.5 Sentence Embeddings Utilizing Pre-trained Models

**Sentence Embedding:** In contrast to conventional word embeddings (e.g., Word2Vec, GloVe), sentence embeddings capture complete sentences within a dense vector space. These embeddings incorporate the contextual significance of sentences.

**SBERT (Sentence-BERT):** SBERT is a modified version of the BERT model [47] that has been fine-tuned particularly for the generation of sentence-level embeddings. BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based model engineered for comprehending the contextual meaning of words within a given sentence.

## 5.6 Data Storage in Chroma Database

**Vector databases** are specialized systems intended for the storage and retrieval of high-dimensional vectors. Chroma exemplifies a system that offers efficient similarity searches for purposes such as document retrieval, semantic search, and question-answering.

**Batch Processing:** Batch storage of embeddings and information enhances storage efficiency and decreases memory saturation levels.

## 4.7 Vector Search

Vector search use sentence embeddings, which are numerical representations of text, to assess semantic similarity between queries and documents. Sentence-BERT produces these embeddings, while ChromaDB saves them, setting up rapid similarity searches. Cosine similarity is utilized to compare vectors and extract the most relevant information based on semantic content rather than just keywords.

Advantages:

* Semantic Search: Generates the result by meaning rather than only on keywords.
* Accuracy: Gets more accurate and pertinent results due to analyzing meaning rather than exact matches.
* Velocity: Rapid retrieval of relevant details, thus necessary for real-time applications.
* Scalability: Effectively manages extensive datasets as the system increases.

This technique improves the chatbot's abilities to respond to questions with greater accuracy and efficiency.

|  |
| --- |
| def vector\_search(model, query, collection, columns\_to\_answer, number\_docs\_retrieval):  query\_embeddings = model.encode([query])    *# Fetch results from the collection*  search\_results = collection.query(  query\_embeddings=query\_embeddings,  n\_results=number\_docs\_retrieval  )  metadatas = search\_results['metadatas']  scores = search\_results['distances']  *# Prepare the search result output*  search\_result = ""  for i, (meta, score) in enumerate(zip(metadatas[0], scores[0]), start=1): *#tao ra 1 cap(metadata, scores)*  search\_result += f"\n{i}) Distances: {score:.4f}"  for column in columns\_to\_answer:  if column in meta:  search\_result += f" {column}: {meta.get(column)}"  search\_result += "\n"  return metadatas, search\_result |

## 5.8 Gemini API

Gemini API is a generative AI model developed by Google, used to generate human-like text responses based on user input. It utilizes transformer architecture, which is the foundation for many modern NLP models. The purpose of using Gemini is to enhance the chatbot’s capabilities by generating hypothetical documents that help provide more accurate and contextually relevant answers based on user queries.

* The model employs pre-trained transformer architectures (such as GPT and LaMDA) for text generation. It enhances the system by incorporating dynamically created content into retrieved information (via RAG, Retrieval-Augmented Generation), so assuring more comprehensive and effective responses [48].
* The purpose of Gemini is to improve semantic search and user experience by generating material that supplements information retrieved from ChromaDB. This enables the system to provide responses even though exact data may not be readily accessible.

Advantages of the Gemini API in the Project:

* Accuracy: Produces material to enhance acquired data, providing accurate and useful answers.
* Flexibility: Supports dynamic content generation in the absence of adequate information inside the database.
* Scalability: As the dataset expands, the model continues to give high-quality results.
* Improved User Experience: Generates natural, contextually relevant responses, enhancing interaction quality [49]

## 5.9 Hype Search

Hype Search use Gemini AI for generating hypothetical papers that enrich the outcomes obtained from ChromaDB. This technique improves the query response based on providing further context when the obtained data is insufficient.

Advantages:

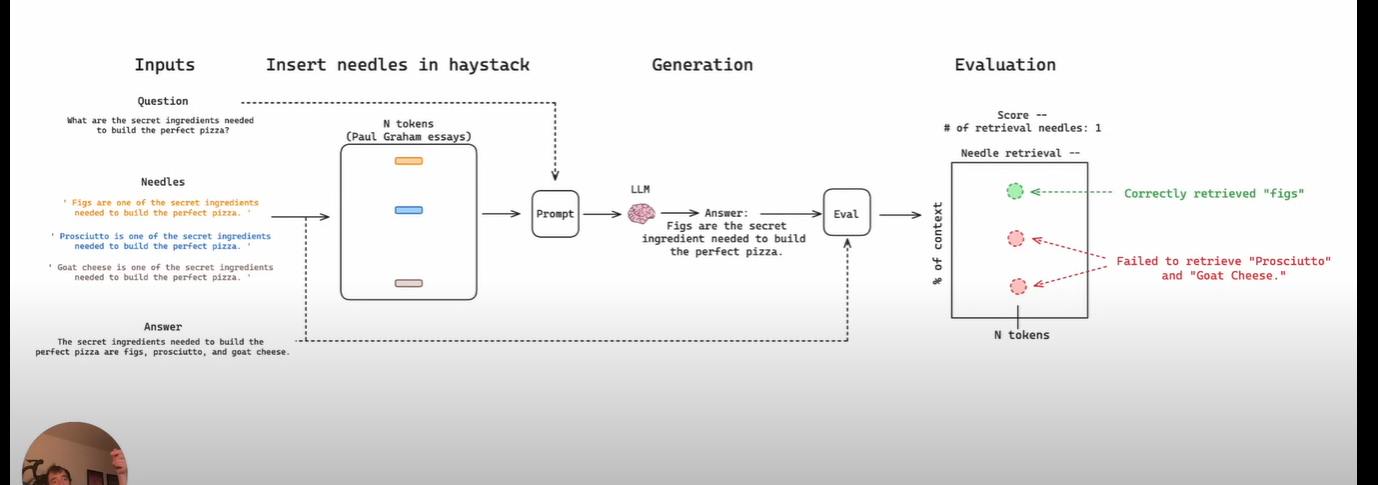
* Improved Accuracy: Integrates real information with AI-generated content for more comprehensive responses.
* Context Enrichment: Addresses inadequacies in the database when information is missing.
* Flexibility: Manages complicated or unclear inquiries by producing relevant text.

HYDE increases the chatbot's capacity of providing more comprehensive and precise responses.

|  |
| --- |
| def generate\_hypothetical\_documents(model, query, num\_samples=10):  hypothetical\_docs = []  for \_ in range(num\_samples):  enhanced\_prompt = f"Write a paragraph that answers the question: {query}"  *# Use the Gemini model stored in session state to generate the document*  response = model.generate\_content(enhanced\_prompt)  hypothetical\_docs.append(response)    return hypothetical\_docs  def encode\_hypothetical\_documents(documents, encoder\_model):  embeddings = [encoder\_model.encode([doc])[0] for doc in documents]  avg\_embedding = np.mean(embeddings, axis=0)  return avg\_embedding  def hyde\_search(llm\_model, encoder\_model, query, collection, columns\_to\_answer, number\_docs\_retrieval, num\_samples=10):    hypothetical\_documents = generate\_hypothetical\_documents(llm\_model, query, num\_samples)  print("hypothetical\_documents", hypothetical\_documents)  aggregated\_embedding = encode\_hypothetical\_documents(hypothetical\_documents, encoder\_model)    search\_results = collection.query(  query\_embeddings=aggregated\_embedding,  n\_results=number\_docs\_retrieval) *# Fetch top 10 results*    search\_result = ""  metadatas = search\_results['metadatas']  i = 0  for meta in metadatas[0]:  i += 1  search\_result += f"\n{i})"  for column in columns\_to\_answer:  if column in meta:  search\_result += f" {column.capitalize()}: {meta.get(column)}"  search\_result += "\n"  return metadatas, search\_result |

A diagram of an entangled document

Description automatically generated with medium confidence

****

CHAPTER 6: RESULT

CHAPTER 7:

## REFERENCE

[1] C. Bouafoud, K. Zine-Dine and A. Madani, "The Evolution of Transformers in Education: A Literature Review," 2024 International Conference on Circuit, Systems and Communication (ICCSC), Fes, Morocco, 2024, pp. 1-7, doi: 10.1109/ICCSC62074.2024.10617128.

[2] A. M., K. Ramasamy, S. G. and K. S.R., "A Systematic Survey of Cognitive Chatbots in Personalized Learning Framework," 2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 2021, pp. 241-245, doi: 10.1109/WiSPNET51692.2021.9419403.

[3] K. Oqaidi, S. Aouhassi and K. Mansouri, "Are Chatbots the Future of Higher Education? Unveiling Their Impact, Challenges, and Prospects," 2024 4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), FEZ, Morocco, 2024, pp. 1-6, doi: 10.1109/IRASET60544.2024.10548820.

[4] P. C. Ramos Pinho and T. T. Primo, "Chatbots in Educational Recommender Systems: A Systematic Literature Review," 2023 IEEE Frontiers in Education Conference (FIE), College Station, TX, USA, 2023, pp. 1-8, doi: 10.1109/FIE58773.2023.10343248.

[5] J. Cheng, "The Impact of Chatbots on Education," 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE), Las Vegas, NV, USA, 2023, pp. 844-849, doi: 10.1109/CSCE60160.2023.00143.

[6] A. N. Ibna Riza, I. Hidayah and P. I. Santosa, "Use of Chatbots in E-Learning Context: A Systematic Review," 2023 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 2023, pp. 0819-0824, doi: 10.1109/AIIoT58121.2023.10174319.

[7] R. W. Maladzhi, M. Tsoeu, N. Mthombeni, K. Moloi, T. Mashifana and F. Nemavhola, "How important are Chatbots within Engineering Education? A literature-based review from 2011 to 2023," 2023 World Engineering Education Forum - Global Engineering Deans Council (WEEF-GEDC), Monterrey, Mexico, 2023, pp. 1-5, doi: 10.1109/WEEF-GEDC59520.2023.10344203.

[8] I. Maita, S. Saide, A. M. Putri and D. Muwardi, "Pros and Cons of Artificial Intelligence–ChatGPT Adoption in Education Settings: A Literature Review and Future Research Agendas," in IEEE Engineering Management Review, vol. 52, no. 3, pp. 27-42, June 2024, doi: 10.1109/EMR.2024.3394540.

[9] E. R. Bekeš and V. Galzina, "Exploring the Pedagogical Use of AI-Powered Chatbots Educational Perceptions and Practices," 2023 46th MIPRO ICT and Electronics Convention (MIPRO), Opatija, Croatia, 2023, pp. 636-641, doi: 10.23919/MIPRO57284.2023.10159734.

[10] M. A. K. Raiaan et al., "A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges," in IEEE Access, vol. 12, pp. 26839-26874, 2024, doi: 10.1109/ACCESS.2024.3365742.

[11] M. A. K. Raiaan et al., "A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges," in IEEE Access, vol. 12, pp. 26839-26874, 2024, doi: 10.1109/ACCESS.2024.3365742.

[12] B. Dong, J. Bai, T. Xu and Y. Zhou, "Large Language Models in Education: A Systematic Review," 2024 6th International Conference on Computer Science and Technologies in Education (CSTE), Xi'an, China, 2024, pp. 131-134, doi: 10.1109/CSTE62025.2024.00031.

[13] I. Hasanov, S. Virtanen, A. Hakkala and J. Isoaho, "Application of Large Language Models in Cybersecurity: A Systematic Literature Review," in IEEE Access, vol. 12, pp. 176751-176778, 2024, doi: 10.1109/ACCESS.2024.3505983.

[14] A. Fan et al., "Large Language Models for Software Engineering: Survey and Open Problems," 2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE), Melbourne, Australia, 2023, pp. 31-53, doi: 10.1109/ICSE-FoSE59343.2023.00008.

[15] Z. R. K. Rostam, S. Szénási and G. Kertész, "Achieving Peak Performance for Large Language Models: A Systematic Review," in IEEE Access, vol. 12, pp. 96017-96050, 2024, doi: 10.1109/ACCESS.2024.3424945.

[16] Milad Moradi, Ke Yan, David Colwell, Matthias Samwald, and Rhona Asgari, "Exploring the Landscape of Large Language Models: Foundations, Techniques, and Challenges," arXiv preprint arXiv:2404.11973, 2024.

[17] Chen Wang, Jin Zhao, and Jiaqi Gong, "A Survey on Large Language Models from Concept to Implementation," arXiv preprint arXiv:2403.18969, 2024.

[18] Milad Moradi, Ke Yan, David Colwell, Matthias Samwald, and Rhona Asgari, "Exploring the landscape of large language models: Foundations, techniques, and challenges," arXiv preprint arXiv:2404.11973, 2024.

[19] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang, "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv preprint arXiv:2312.10997, 2024.

[20] R. Zhang et al., "Interactive AI With Retrieval-Augmented Generation for Next Generation Networking," in IEEE Network, vol. 38, no. 6, pp. 414-424, Nov. 2024, doi: 10.1109/MNET.2024.3401159.

[21] Han, B.; Susnjak, T.; Mathrani, A. Automating Systematic Literature Reviews with Retrieval-Augmented Generation: A Comprehensive Overview. Appl. Sci. 2024, 14, 9103. https://doi.org/10.3390/app14199103

[22] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui, "Retrieval-Augmented Generation for AI-Generated Content: A Survey," arXiv preprint arXiv:2402.19473, 2024.

[23] K. Sawarkar, A. Mangal and S. R. Solanki, "Blended RAG: Improving RAG (Retriever-Augmented Generation) Accuracy with Semantic Search and Hybrid Query-Based Retrievers," 2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR), San Jose, CA, USA, 2024, pp. 155-161, doi: 10.1109/MIPR62202.2024.00031.

[24] T. Prudhvith, C. Swattik and S. Prakash, "Enhancing Retrieval Augmented Generation Systems with Knowledge Graphs," 2024 International Conference on Electrical, Computer and Energy Technologies (ICECET, Sydney, Australia, 2024, pp. 1-8, doi: 10.1109/ICECET61485.2024.10698122.

[25] T. Sun, A. Somalwar and H. Chan, "Multimodal Retrieval Augmented Generation Evaluation Benchmark," 2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring), Singapore, Singapore, 2024, pp. 1-5, doi: 10.1109/VTC2024-Spring62846.2024.10683437.

[26] Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh, "A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions," arXiv preprint arXiv:2410.12837, 2024.

[27] LangChain, "LangChain," Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/LangChain?utm\_source=chatgpt.com.

[28] Keivalya Pandya and Mehfuza Holia, "Automating Customer Service using LangChain: Building custom open-source GPT Chatbot for organizations," arXiv preprint arXiv:2310.05421, 2023.

[29] A. Sar et al., "PDF-Based Chatbot Development Using LLAMA2 and LangChain: Training and Deployment for Document Interaction," 2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0, Raigarh, India, 2024, pp. 1-6, doi: 10.1109/OTCON60325.2024.10688337.

[30] D. Madhav, S. Nijai, U. Patel and K. Champanerkar, "Question Generation from PDF using LangChain," 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2024, pp. 218-222, doi: 10.23919/INDIACom61295.2024.10499105.

[31] H. Beattie, L. Watkins, W. H. Robinson, A. Rubin and S. Watkins, "Measuring and Mitigating Bias in AI-Chatbots," 2022 IEEE International Conference on Assured Autonomy (ICAA), Fajardo, PR, USA, 2022, pp. 117-123, doi: 10.1109/ICAA52185.2022.00023.

[32] R. Amendi, E. Halim and H. Hartono, "Exploring Ethical Implications: Unraveling Factors Influencing Data Governance Awareness Behavior in Generative AI Chatbot," 2024 2nd International Conference on Technology Innovation and Its Applications (ICTIIA), Medan, Indonesia, 2024, pp. 1-6, doi: 10.1109/ICTIIA61827.2024.10761290.

[33] A. Ghandour, B. J. Woodford and H. Abusaimeh, "Ethical Considerations in the Use of ChatGPT: An Exploration Through the Lens of Five Moral Dimensions," in IEEE Access, vol. 12, pp. 60682-60693, 2024, doi: 10.1109/ACCESS.2024.3394243.

[34] W. Elhag, D. Bouteldja and S. Bouallegue, "A Survey on the Applications, Limitations, and Ethical Considerations of ChatGPT in Various Industries," 2023 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE), Milano, Italy, 2023, pp. 565-569, doi: 10.1109/MetroXRAINE58569.2023.10405608.

[35] A. Jedlickova, "Ensuring Ethical Standards in the Development of Autonomous and Intelligent Systems," in IEEE Transactions on Artificial Intelligence, vol. 5, no. 12, pp. 5863-5872, Dec. 2024, doi: 10.1109/TAI.2024.3387403.

[36] T. Simonite, "OpenAI’s New Voice Mode Sparks Worries About Emotional Attachment," Wired. [Online]. Available: https://www.wired.com/story/openai-voice-mode-emotional-attachment/?utm\_source=chatgpt.com.

[37] M. Al-Emran, A. A. AlQudah, G. A. Abbasi, M. A. Al-Sharafi and M. Iranmanesh, "Determinants of Using AI-Based Chatbots for Knowledge Sharing: Evidence From PLS-SEM and Fuzzy Sets (fsQCA)," in IEEE Transactions on Engineering Management, vol. 71, pp. 4985-4999, 2024, doi: 10.1109/TEM.2023.3237789.

[38] R. Ren, M. Zapata, J. W. Castro, O. Dieste and S. T. Acuña, "Experimentation for Chatbot Usability Evaluation: A Secondary Study," in IEEE Access, vol. 10, pp. 12430-12464, 2022, doi: 10.1109/ACCESS.2022.3145323.

[39] N. Damij and S. Bhattacharya, "The Role of AI Chatbots in Mental Health Related Public Services in a (Post)Pandemic World: A Review and Future Research Agenda," 2022 IEEE Technology and Engineering Management Conference (TEMSCON EUROPE), Izmir, Turkey, 2022, pp. 152-159, doi: 10.1109/TEMSCONEUROPE54743.2022.9801962

[40] A. Miklosik, N. Evans and A. M. A. Qureshi, "The Use of Chatbots in Digital Business Transformation: A Systematic Literature Review," in IEEE Access, vol. 9, pp. 106530-106539, 2021, doi: 10.1109/ACCESS.2021.3100885.

[41] B. Rawat, A. S. Bist, U. Rahardja, Q. Aini and Y. P. Ayu Sanjaya, "Recent Deep Learning Based NLP Techniques for Chatbot Development: An Exhaustive Survey," 2022 10th International Conference on Cyber and IT Service Management (CITSM), Yogyakarta, Indonesia, 2022, pp. 1-4, doi: 10.1109/CITSM56380.2022.9935858.

[42] S. K. Maher, S. G. Bhable, A. R. Lahase and S. S. Nimbhore, "AI and Deep Learning-driven Chatbots: A Comprehensive Analysis and Application Trends," 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2022, pp. 994-998, doi: 10.1109/ICICCS53718.2022.9788276.

[43] M. Černý, "The history of chatbots: the journey from psychological experiment to educational object," Journal of Applied Technical and Educational Sciences, vol. 12, no. 3, pp. 1–28, 2022, doi: 10.24368/jates322.

[44] Google, "Introducing Gemini: Unlocking the future of AI with multimodal capabilities," Google Blog, Dec. 6, 2023. [Online]. Available: https://blog.google/technology/ai/google-gemini-ai/?utm\_source=chatgpt.com#introducing-gemini.

[45] Amazon Web Services, "What is retrieval-augmented generation?" AWS, [Online]. Available: https://aws.amazon.com/what-is/retrieval-augmented-generation/?utm\_source=chatgpt.com.

[46] Real Python, "ChromaDB: A vector database for machine learning," Real Python, [Online]. Available: https://realpython.com/chromadb-vector-database/?utm\_source=chatgpt.com.

[47] Nils Reimers, and Iryna Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," arXiv preprint arXiv:1908.10084, 2019.

[48] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei, "Language Models are Few-Shot Learners," arXiv preprint arXiv:2005.14165, 2020.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention Is All You Need," arXiv preprint arXiv:1706.03762, 2023.

[50] N. Van Otten, "ROUGE Metric In NLP: Complete Guide & How To Tutorial In Python," Spot Intelligence, Aug. 12, 2024. [Online]. Available: https://spotintelligence.com/2024/08/12/rouge-metric-in-nlp/.

[51] Verge AI, "Chatbot for Higher Education: Enhancing Student Engagement and Streamlining Administration," Verge AI. [Online]. Available: https://verge-ai.com/chatbot-for-higher-education/?utm\_source=chatgpt.com.

[52] F. (Mihai) Leța and D. P. C. Vancea, "Ethics in Education: Exploring the Ethical Implications of Artificial Intelligence Implementation," Ovidius University Annals, Economic Sciences Series, vol. XXIII, no. 1, pp. 412–421, 2023. Available: https://www.ovidius-university.net.

[53] AtlasRTX, "University Chatbots: Transforming the Admissions Process," AtlasRTX. [Online]. Available: https://atlasrtx.com/university-chatbots-transforming-admissions-process/?utm\_source=chatgpt.com.

[54] Xiao, C., Xiao, J., Jiang, K., Feng, L., Yang, R. (2024). The Role of AI Chatbots in Transforming Higher Education. In: Peters, M.A., Heraud, R. (eds) Encyclopedia of Educational Innovation. Springer, Singapore. https://doi.org/10.1007/978-981-13-2262-4\_292-1

A. Author, B. Author, and C. Author, "Title of the paper," arXiv:2404.11973, Apr. 2024. [Online]. Available: https://arxiv.org/abs/2404.11973