جامعة ابن طفيل
ⵜⴰⵙⴷⴰⵡⵉⵜ ⵉⴱⵏ ⵟⵓⴼⴰⵢⵍ
Ibn Tofaïl University
Faculté des Sciences

Ibn Tofail University

Faculty of Sciences, Kénitra

End of Study Project Report

Master Of Artificial Intelligence And Virtual Reality

Virtual Reality Tour of Capgemini Engineering : Chatbot Development

Host establishment : **Capgemini Engineering**

Prepared by: Miss Ouafa AIT OUAMER

Supervised by: Mr Rochdi MESSOUSSI (FSK-UIT)

Mrs Awatif ESHAIMI (Capgemini Engineering)

Mrs Khadija IZAQAFTEN (Capgemini Engineering)

Defended on : ***September 19th, 2024, before the jury composed of:***

- Mr Rochdi MESSOUSSI (FSK Ibn Tofail University)
- Mrs Raja TOUAHNI (FSK Ibn Tofail University)
- Mrs Khaoula BOUKIR (ENSC Ibn Tofail University)
- Mr Anass NOURI (FSK Ibn Tofail University)

Academic year 2023/2024

Acknowledgments

Before presenting my work, I'd like to express my deep gratitude to all the people who contributed to the realization of this project. May all those who contributed, both directly and indirectly, to the success of this project find here, collectively and individually, the expression of all my gratitude and respect.

I would like to extend my heartfelt gratitude to my Academic Supervisor, **Mr. MESSOUSSI Mohammed**, and my Capgemini Engineering Supervisors, **Mrs. IZAQAFTEN Khadija** and **Mrs. ESHAIMI Awatif**. Their unwavering support, guidance, and insightful feedback have been instrumental in the success of our project and the preparation of this report. Their exceptional professionalism, encouragement, and collaborative spirit provided me with invaluable learning experiences and a strong foundation for my career.

On this special occasion, we would like to express our gratitude to **Mrs. TOUAHNI Rajaa**, the coordinator of the Master IARV program, for the quality of the education she provided, as well as for her support and guidance throughout our journey.

Special thanks are due to the engineering team manager, **Mrs. RACHIH Hanane**, the interns responsible Mr. **EL OGDI Ilyas** and our colleagues for their continuous help , encouragement, guidance, and valuable insights, which greatly enriched our research.

Thanks also go to the members of the jury for agreeing to evaluate this work. I hope that this work will be useful and meet their expectations. Additionally, I would like to thank all the professors and staff at the school, whose dedication and commitment to training the engineers of tomorrow have never wavered.

Finally, I do not forget my family and friends, whose moral support has been a pillar of my balance and perseverance.

Résumé

Dans le domaine de la Réalité Virtuelle (RV) et des systèmes intelligents, l'amélioration de l'interaction utilisateur et de l'accessibilité de l'information est devenue un défi essentiel, notamment au sein de grandes organisations telles que Capgemini Engineering. Ce rapport présente un projet complet entrepris lors d'un stage avancé, axé sur le développement d'un chatbot intelligent destiné à améliorer la visite en Réalité Virtuelle de Capgemini Engineering, en particulier dans le département d'Ingénierie des Systèmes Basés sur les Modèles (MBSE).

L'objectif principal de ce stage était de créer un chatbot alimenté par l'intelligence artificielle (IA) capable de répondre à un large éventail de questions en temps réel, en fournissant des informations détaillées sur les départements, projets et équipes de l'entreprise. En tirant parti des dernières avancées en Traitement du Langage Naturel (NLP), en Apprentissage Automatique (ML), et des Modèles de Langage de Grande Taille (LLMs), le chatbot permet des réponses personnalisées et précises, favorisant une plus grande interaction et engagement des utilisateurs. Le projet a suivi une approche de développement itératif, couvrant les phases de recherche, de conception et de test afin d'assurer une compatibilité parfaite avec les futures applications VR.

Ce rapport examine les méthodes et technologies employées pour développer un prototype fonctionnel de chatbot, mettant en lumière son potentiel à transformer l'intégration des nouveaux employés, l'engagement des clients, et les communications internes au sein de l'entreprise. Le succès de ce projet démontre l'importance croissante des solutions basées sur l'IA pour rendre les environnements VR plus accessibles, interactifs et informatifs, ouvrant la voie à de futures avancées dans les systèmes intelligents et les expériences virtuelles.

Mots clés— RV, Chatbot, ML, NLP, IA, LLMs, MBSE

Abstract

In the realm of Virtual Reality (VR) and intelligent systems, enhancing user interaction and information accessibility has become a vital challenge, especially within large organizations like Capgemini Engineering.This report outlines a comprehensive project undertaken during an advanced internship, focused on the development of an intelligent chatbot designed to enhance the Virtual Reality Tour of Capgemini Engineering, particularly in the Model-Based Systems Engineering (MBSE) department.

The primary objective of the internship was to create an AI-powered chatbot capable of answering a wide range of queries in real-time, providing detailed information about the companys departments, projects, and teams. By leveraging state-of-the-art Natural Language Processing (NLP), Machine Learning (ML), and advanced Large Language Models (LLMs), the chatbot enables personalized and accurate responses, fostering greater user engagement and interaction. The project followed an iterative development approach, covering research, design, and testing phases to ensure seamless compatibility within future VR applications.

This report delves into the methods and technologies employed to develop a functional chatbot prototype, highlighting its potential to transform user onboarding, client engagement, and internal communications within the company. The success of this project demonstrates the growing importance of AI-driven solutions in making VR environments more accessible, interactive, and informative, laying the groundwork for future advancements in intelligent systems and virtual experiences.

Keywords— VR, Chatbot, ML, NLP, AI, LLMs, MBSE

Contents

List of Figures

List of Tables

General Introduction

In the rapidly evolving landscape of AI-driven and Virtual Reality applications, integrating intelligent systems has become essential for enhancing user experience. My end-of-studies project at Capgemini Engineering reflects this trend by focusing on the development of an AI-powered chatbot designed to improve the existing Virtual Reality (VR) Tour of Capgemini. This project aims to create a more immersive and informative experience for new employees, potential clients, and interns, particularly within the Model-Based Systems Engineering (MBSE) department.

The key objective of this project was to develop an intelligent chatbot capable of providing real-time assistance by answering user queries and delivering detailed information about the companys departments, projects, and teams. By leveraging advanced language models (LLMs), the chatbot is able to understand natural language and offer personalized, accurate responses, facilitating enhanced interaction and engagement in the VR environment.

This project also emphasizes the seamless integration of the chatbot into the existing VR application, ensuring compatibility with Oculus Rift and other similar headsets. The chatbot's real-time interaction capability aims to enrich the onboarding experience for new employees and clients, while enhancing overall information accessibility.

Organization of the Report

- **Chapter 1** presents a general context by introducing the hosting organization and outlining the project.
- **Chapter 2** focuses on the state of the art and generalities, examining recent advancements and domain-related issues.
- **Chapter 3** outlines the design phase of the project, focusing on system architecture and chatbot development.
- Finally, **Chapter 4** discusses the projects implementation and the results achieved, describing the development stages and presenting the systems performance. Together, these four chapters offer a comprehensive view of the project, from its conception to its implementation, highlighting its significance and contributions in the relevant domain.

Chapter 1

Host Organization Presentation and Project Context

Contents

1.1 Introduction

The first part of this first chapter presents a brief description of the Capgemini Engineering group as an organization where my end-of-studies project took place, as well as its various departments and sectors of activity. For the second part, it is used to describe the projects specifications and objectives.

1.2 Presentation of the Host Organization

1.2.1 Capgemini Engineering

Figure 1.1: Logo of Capgemini Engineering

As a global leader in innovation consulting and advanced engineering, Capgemini Engineering supports companies in their processes of creating and developing new products and services. The Group has been operating for nearly 30 years, serving the largest players in sectors such as aerospace, automotive, energy, rail, finance, healthcare, telecommunications, and more. The Groups offerings, developed from strategic planning phases regarding new technologies to in- dustrialization phases, ensure knowledge capitalization within four main domains: Intelligent Systems, Innovative Product Development, Life Cycle Experience, and Mechanical Engineering and Information Systems.

Figure 1.2: The areas of expertise of Capgemini Engineering

1.2.2 Capgemini Engineering on the International Stage

The Capgemini Engineering group is established in 25 countries worldwide, spanning across Europe, Asia, America, and recently in Morocco (See Figure 1.3). The Group has also aimed

to maintain a local dimension to enable specific support in dedicated and nearby markets.

Figure 1.3: Capgemini Engineering's Presence Worldwide

1.2.3 Business Domains of Capgemini Engineering

For over 35 years, Capgemini Engineering has been collaborating with major players in numerous sectors, including: Aerospace, Automotive, Space, Naval Defense, Rail, Transportation Infrastructure, Consumer Goods Industry, Communications, Semiconductor Electronics, Internet Software, Finance, Public Sector, and Life Sciences (See Figure 1.4).

Figure 1.4: Business Domains of Capgemini Engineering

Capgemini Engineering handles all phases of a projects lifecycle, from its definition (technological watch, strategy definition, technical feasibility studies, etc.) to its realization (design, implementation, and validation of solutions, etc.). The activity is structured around three themes:

- Technology and RD Consulting: This entails the ability to put into practice the expertise, technical methods, and scientific knowledge of engineering consultants to achieve innova- tive projects.
- Organization and Information Systems Consulting: Provides companies with the opportunity to remain competitive despite market growth constraints, profitability, and legislation. Companies that engage Capgemini Engineering in this regard aim to facilitate their decision-making processes and organizational agility.
- Strategy and Management Consulting: Enables companies to manage and control their environment and anticipate changes in their industry. This helps increase their long-term success potential.

1.2.4 Clients and Partners

As a strategic partner, Capgemini Engineering offers comprehensive project support to its clients while ensuring a consistent level of service. The group has also aimed to maintain a local presence, allowing for specific support in dedicated and nearby markets.

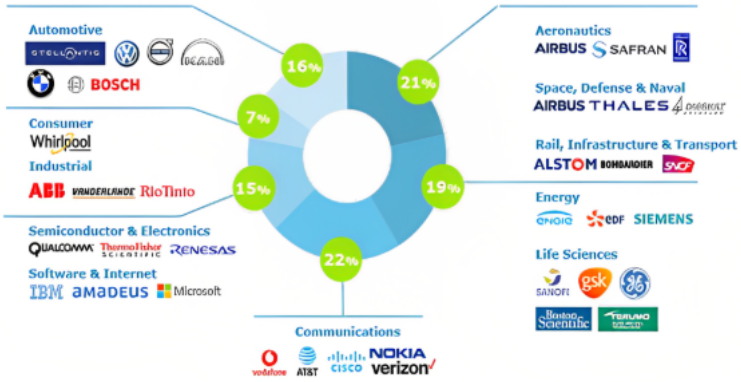

Figure 1.5: Capgemini Engineering Clients and Partners

1.2.5 Capgemini Engineering in Morocco

Capgemini Engineering Morocco is located in the Casa nearshore Park in Casablanca (Figure 5), the economic capital of Morocco. Their facilities offer a framework of European standards, including technological equipment (optical fiber, equipped video conference rooms, etc.) and living spaces (kitchens, changing areas, etc.). Capgemini Engineering aimed to

estab- lish a Nearshore platform to support the groups international development in the automotive, aerospace, and transportation sectors. This involves supporting Capgemini Engineerings clients in their innovation, cost optimization, and internationalization strategies. The Moroccan entity also aims to be a local player serving Capgemini Engineerings major clients located in the coun- try. Finally, Capgemini Engineering Morocco relies on the offshoring strategy implemented by the Moroccan government, offering advantages that significantly optimize the skills/cost aspect.

Figure 1.6: Capgemini Engineering Morocco's Local Presence at Casa Nearshore

1.2.5.1 Fact sheet of Capgemini Engineering

Capgemini Engineering	**Capgemini Engineering**
Date of creation	2013
Location	Casablanca
Legal form	Limited Liability Company
Number of employees	More than 2400 employees
Main industries	Automobile, Infrastructure, Transport, Aerospace, Pharma, and Energy
Main solutions	The solutions cover five main technology domains (innovative product development, smart systems, lifecycle experience, mechanical engineering)

Table 1.1: Fact sheet of Capgemini Engineering

1.2.5.2 Missions

Through its presence in Morocco, Capgemini Engineering aimed to establish a Nearshore plat- form to support the groups international development in the automotive, aerospace, and

trans- portation sectors. This involves supporting Capgemini Engineerings clients in their innovation, cost optimization, and internationalization strategies.

1.2.5.3 Major Occupational Categories

Capgemini Engineering is made up of 3 main business lines, the proportion of which varies:

- Consultants (Technicians and Engineers)
- Management
- Cross-functional functions (Finance, HR, General Services, Communication, etc.)

1.2.5.4 Hierarchical Organizational Chart

Under Capgemini Engineerings new strategy, operating teams are divided into four divisions, each offering solutions tailored to customer needs:

- Human Resources Department (HRD)
- Finance Department (FD)
- Program Department (PD)
- Technical Department (TD)

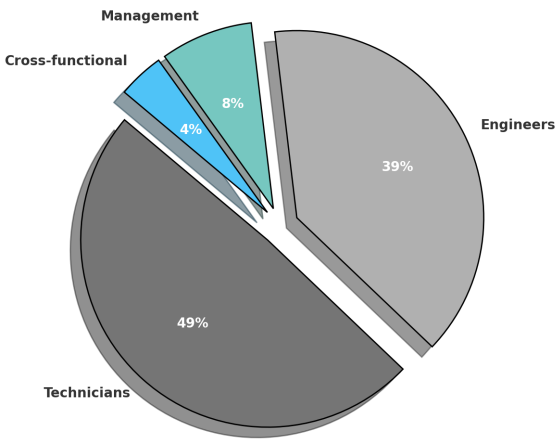

Figure 1.7: Capgemini Engineering Distribution Of Employees By Role

1.2.5.5 Presentation of the Home Department

My end-of-studies was carried out within the MBSE department, which specializes in model-based systems engineering. The goal of this approach is to design and implement

solutions more efficiently by reducing potential errors. The primary function of the employees is to physically represent the system that the company wishes to develop, which facilitates the design and implementation of technological tools, as well as the integration of embedded software and ensuring coherence between the different elements of the process. Systems engineers play a key role in the success of this process, which improves the company's productivity and profitability. The department's organizational chart is represented as follows:

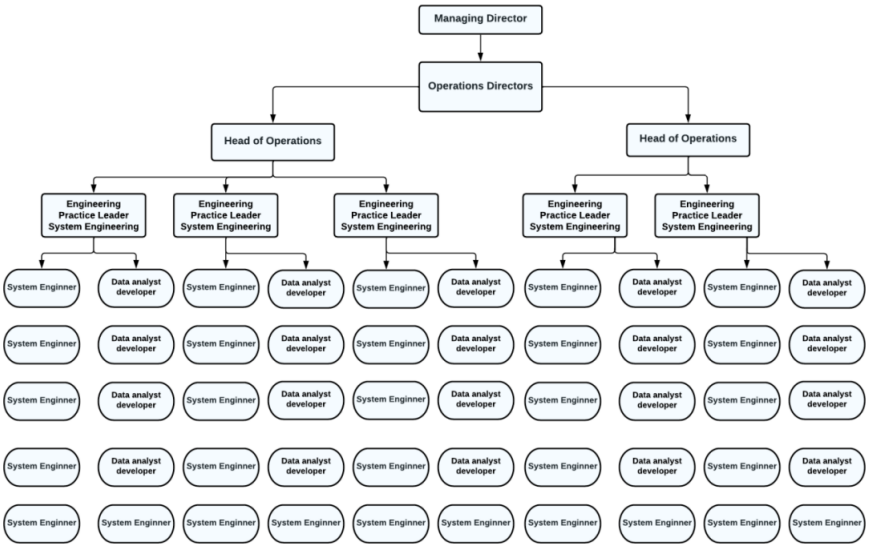

Figure 1.8: Global Organizational Chart

1.2.5.6 Introduction to the Host Team

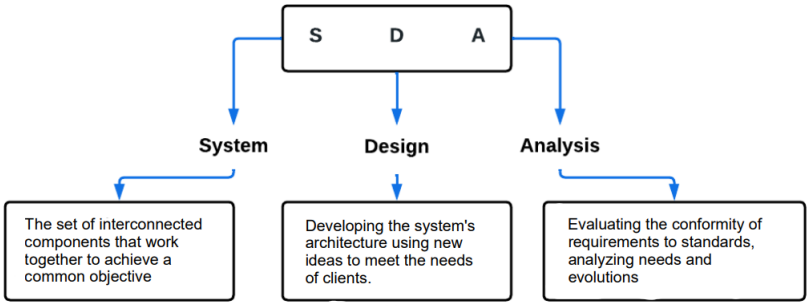

Figure 1.9: SDA team

During my internship, I had the privilege of becoming an integral part of the SDA (System Design and Analysis) team within the MBSE department, under the supervision of my internship mentors. This team plays a crucial role in system design and analysis, ensuring that stakeholder needs and the specific requirements of each project are met. One of the fundamental values I learned while working with the SDA team is the importance of commu-

nication and collaboration. Each team member is in constant interaction with other teams and stakeholders to ensure that designs and analyses meet the expectations of all involved parties. This collaborative approach fosters a deep understanding of specific needs while allowing for synergy among the various expertise within the team.

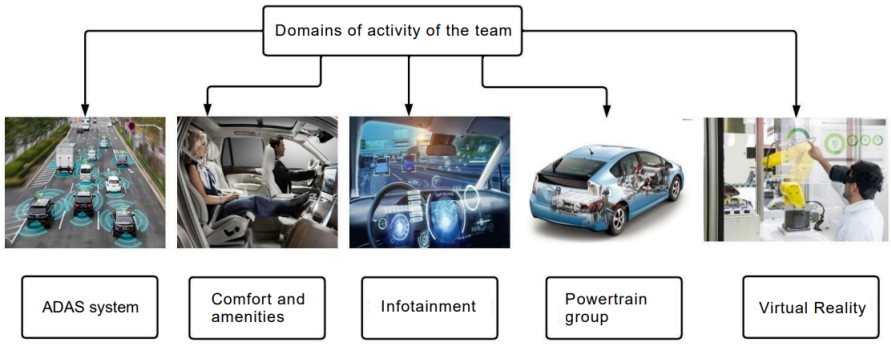

Figure 1.10: Domains of activity of the SDA team

The SDA team is positioned at the heart of the automotive industry by offering innovative projects focused on innovation and digitalization. Its primary objective is to ensure customer satisfaction by integrating cutting-edge technologies. By focusing on key areas such as ADAS systems, vehicle comfort and convenience, infotainment, and the management of both fuel and electric engines, the SDA team provides solutions tailored to the specific needs of the automotive industry.

Moreover, the SDA team stands out for its commitment to the latest technological trends. It is at the forefront of innovation by exploring new horizons, particularly in the field of virtual reality. This approach demonstrates Capgemini Engineering's willingness to embrace the opportunities presented by current technological advancements, in order to offer innovative solutions that meet the evolving needs of the industry.

1.3 Presentation of the Project

This section is devoted to a presentation of the context and problematic of my final year project, as well as the solution proposed for the realization of this work.

1.3.1 Project Context

1.3.1.1 General Problem Statement

The general problem addressed by this project lies in managing the onboarding of new employees, interns, and potential clients at Capgemini Engineering, considering the large size of the company and its global operations. Providing personalized assistance to each newcomer demands substantial human and financial resources, requiring the involvement of numerous

team members to offer guidance, training, and essential information. The traditional onboarding approach is resource-intensive and costly.

The proposed solution is the development of an intelligent chatbot integrated with a virtual reality (VR) tour. The chatbot will enhance the onboarding process by offering real-time, personalized assistance, answering user questions, and delivering detailed information on Capgemini Engineerings teams, projects, and values. Simultaneously, the virtual tour will allow users to explore the company's various departments and activities in an interactive and immersive way. Together, these tools will offer a more engaging and accessible onboarding experience, while significantly reducing the demand for human resources.

This innovative solution alleviates the burden on human personnel by automating the delivery of information through the chatbot, which serves as a virtual assistant. In addition, the combination of the chatbot and VR tour reduces onboarding costs while providing an engaging, personalized, and efficient experience for newcomers. By improving interaction and accessibility, this approach ensures that employees, interns, and clients can access the necessary information in a more streamlined and immersive manner.

1.3.1.2 Project specifications

To successfully carry out the project, it is essential to define the requirements that must be followed precisely.

Context:

- Development of an intelligent chatbot to enhance the existing Virtual Reality Tour of Capgemini Engineering.
- Aimed at improving information accessibility for new employees, potential clients, and interns that are related to the MBSE departement.
- Provides real-time assistance during the VR tour by answering user queries and delivering detailed information on department activities, projects, and teams.
- Future integration of the chatbot with the virtual reality application, compatible with Oculus Rift and similar headsets.
- The chatbot will use advanced language models (LLMs) to understand natural language and provide personalized, accurate responses.
- Facilitates enhanced interaction and engagement with users through both the chatbot interface and the VR environment.

Objectives:

- Develop an intelligent chatbot that can handle a wide range of questions regarding the company's departments, projects, and teams.

- Ensure that the chatbot is capable of being integrated into the existing VR application for a seamless user experience.

- Improve user engagement and provide a richer, more interactive onboarding experience for new employees, clients, and interns.

Scope:

- The chatbot is designed for Capgemini Engineering's clients, employees, and interns worldwide.

- It aims to provide a deeper understanding of the company through interactive, real-time information delivery within the VR tour.

Functionality:

- Users will interact with the chatbot via a web-based interface or through a virtual reality headset such as the Oculus Rift.

- The chatbot will guide users through the virtual tour, answering questions and providing detailed information on department activities, projects, and teams.

- Users can interact with the chatbot using text inputs, and the responses will be tailored to the user's role within the company (new employee, client, or intern).

- The chatbot will enhance user experience by delivering context-sensitive information during the virtual tour, improving engagement and knowledge retention.

Resources:

- Data and Information: Includes team descriptions, project details, descriptions of various parts of the company, and success stories.

- Project Team: Includes the team leader for supervising project progress, my supervisors within the company, and my academic advisor.

1.3.2 SMART Goals

At the end of this project, we aim to achieve an objective that must be SMART:

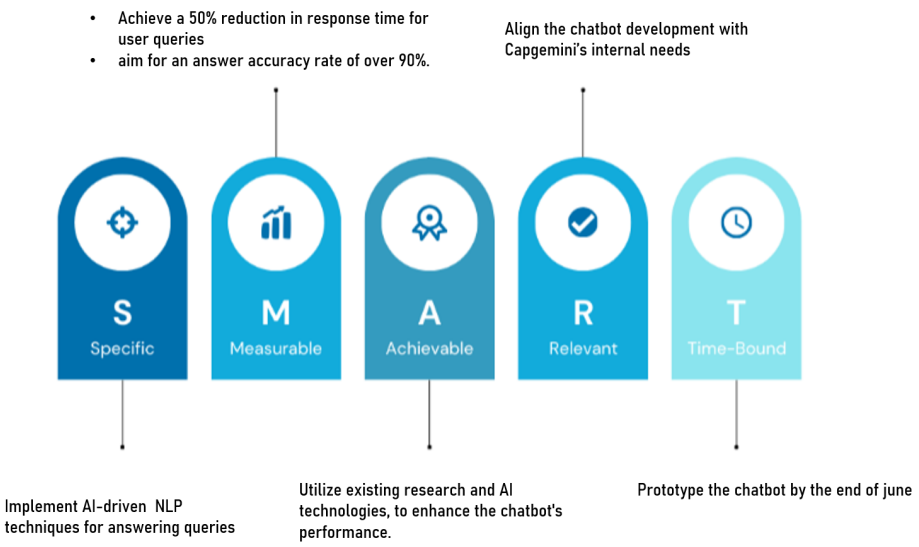

Figure 1.11: SMART Goals Infographic

1.4 Work Methodology

1.4.1 Agile Methodology

For years, most software development projects have relied on methodologies called agile. This banner combines several methods based on the iterative and incremental development, in which the search for solutions to problems is based on collaboration, involving the client from the beginning to the end of the project. It believes that the need cannot be set in stone and proposes to adapt to the changes in the latter. The agile methodology aims to have:

- **Better quality of communication**: The customer has the possibility to clarify or modify needs as they go along.
- **Better visibility**: The client has better visibility on the progress of their project.
- **Better product quality**: Testing is done continuously.
- **Better cost control**: The project can be stopped due to lack of budget.

1.4.2 Agile Scrum Methodology

Scrum is considered a project management framework. This framework consists of a definition of roles, meetings, and artifacts. Scrum defines 3 roles:

- **The Product Owner:** This individual serves as the customer and the users representative. They define the requirements, product specifications, and the order in which the functions will be developed.
- **The Scrum Master:** This role acts as the intermediary between the Product Owner and the development team. Their primary responsibility is to ensure the proper application of the Scrum methodology.
- **The Development Team:** This team is responsible for creating the product while adhering to the preset deadlines. It operates as a self-organizing unit and typically comprises a diverse range of profiles including architects, designers, developers, HMI specialists, testers, etc.

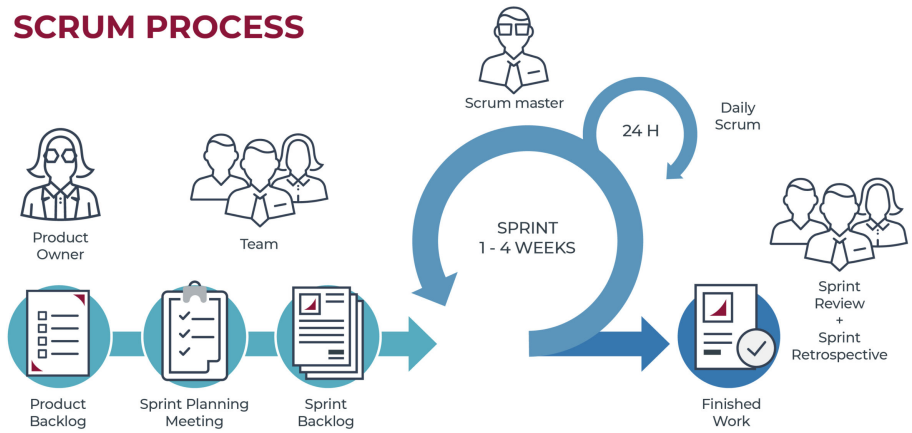

Figure 1.12: Steps of the SCRUM Methodology.

The project follows the SCRUM agile development process in order to identify the needs systematic, specify, and plan within timelines and applying good engineering practices software for the entire project. Working under the agile Scrum framework is done according to a set of sprints. A sprint refers to the development cycle during which a number of tasks will be carried out. It can be for a duration of 15 days, 1 month or more depending on the features that will be developed.

Before launching a sprint, the team should organize a sprint plan that aims to planning the tasks of the next sprint and organizing the progress of the project. The team Scrum containing: Product Owner, Scrum Master and Developers must report to this first day of the sprint to decide which backlog items to develop within a limit of Well-defined time, which is the duration of the sprint.

Throughout the sprint, we continue to organize what we call Daily Scrum that presents itself in the form of a 15-minute interim meeting (Time Boxed). It brings together the members of the development team, during which each member must answer 3 questions:

- What has he achieved during the day?
- What does he intend to do for the next day?
- What constraints are blocking him?

At the end of the sprint, the scrum master should schedule a meeting for the developers to present the deliverable to the product owner and demonstrate everything that has been done developed during the sprint, we are talking here precisely about the Sprint Review. The latter will be followed through a retrospective sprint whose purpose is to discuss the team's performance during the sprint, things to improve, keep, or avoid.

1.4.3 Agile Tool (Jira Software)

Jira Software is an agile project management tool that supports all agile methodologies. It makes it easy to create agile tables, backlogs, roadmaps, reports, integrations, or add-ons, you can So plan, track, and manage all agile software development projects from a single tool. Jira is project management solution published by Atlassian. It allows teams to Organize effectively, establish sustainable communication and visualize the project at a glance with its personalized dashboards. This tool is recognized as the most used by software development teams[1].

Figure 1.13: Jira Software logo

Jira Software also allows you to:

- Work in an agile way thanks to Kanban and Scrum boards.
- Accelerate the delivery of project deliverables.
- Continuously improve projects.

1.5 Project Planning

1.5.1 Scheduling Tasks on Jira Software

Before starting the project, we defined the various elements of the Scrum methodology. these are the planned sprints for chatbot development :

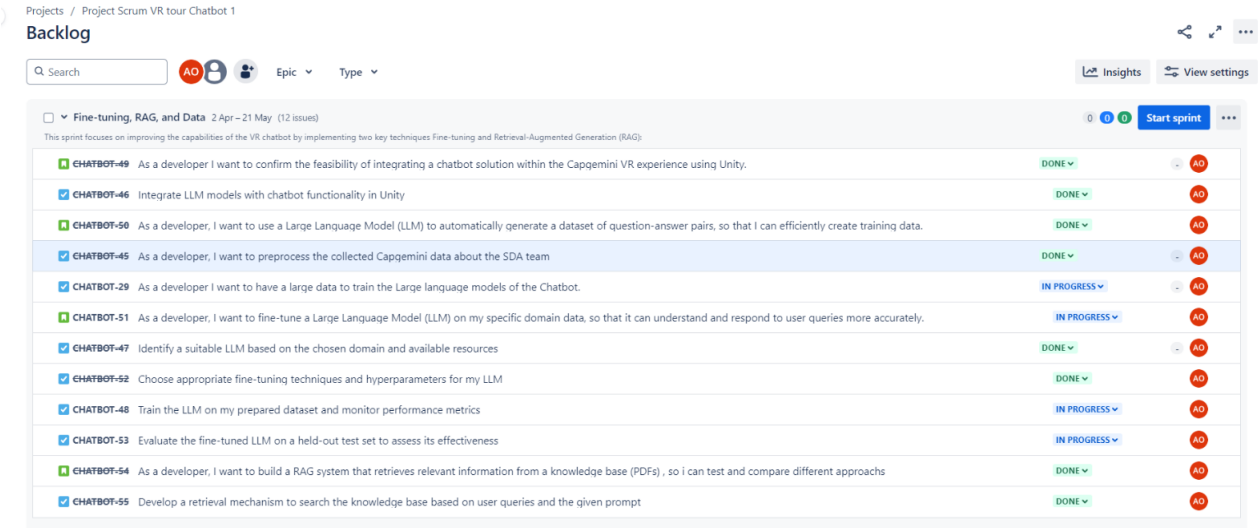

Figure 1.14: Sprint 6 Planning On Jira Software

PO: Our team consists of my supervisors and myself.
– Chatbot Development
Sprint 1 : Planification et conception du projet
Sprint 2 : Collecte et préparation des données
Sprint 3 : Exploration et sélection des LLMs
Sprint 4 : Intégration initiale des LLMs
Sprint 5 : Fine-tuning des modèles
Sprint 6 : Développement du système RAG
Sprint 7 : Intégration complète et tests
Sprint 8 : Tests de performance et ajustements
Sprint 9 : Test, suivi et maintenance

1.5.2 the QQOQCP Method

Before we dive into the translation, it's important to understand that QQOQCP is a French acronym for a method used to gather information by asking specific questions. It stands for:
– Qui: Who
– Quoi: What
– Où: Where
– Quand: When
– Comment: How
– Pourquoi: Why

This method is often used in project management to define the scope, objectives, and constraints of a project.

Table 1.2: The QQOQCP Method

QQOQCP	Purpose	Questions	Target
Who	Define actors and people involved in the project	Who is concerned by the problem? Who is the project team? Who are the stakeholders involved in the project?	Capgemini Engineering company - Ms. IZAQAFEN Khadija, industrial sponsor. - Ms. ESHAIMI Awatif, industrial sponsor. Trainees, new employees, potential clients
What	Description of the problem	What is it about? What is the current state?	Development of a conversational chatbot application to provide real-time assistance to users. The project is currently in the design and planning phase, where we are defining objectives, functionalities, and requirements for the chatbot application. The project is in the design and planning phase, defining objectives, functionalities, and requirements.
Where	Description of the place	Where is this project taking place?	The project is carried out in the MBSE department
When	Description of time	What is the duration of the project?	The project is spread over a period of 4 months, from February 28 to june 28, 2024
How	Description of the means	How does one proceed?	Collection and pre-processing of specific data, selection and fine-tuning of language models (LLMs), development of the chatbot. Preparation of the necessary resources, determination of the application's various functionalities.
Why	Description of the reason for the project	Why carry out these actions?	The goal of this project is to develop chatbot that provides personalized, real-time assistance for system engineers and any related MBSE departement user. Both are intended to enhance user experience by offering instant, relevant information, ultimately improving efficiency and engagement during virtual visits.

1.5.3 Project Planning on GANTT

In order to have a global vision of the stages of the project and to be able to better manage the time, which is the duration of the internship, a project schedule is therefore essential. The planning is one of the most important pre-project phases. It consists of scheduling and to determine the tasks of the project and estimate their respective costs. Among the tools of project planning, I opted for the GANTT chart, its a tool that allows you to plan the project and make it easier to track its progress. This diagram also allows you to visualize the sequence and duration of the different tasks during the internship as shown in the next figure:

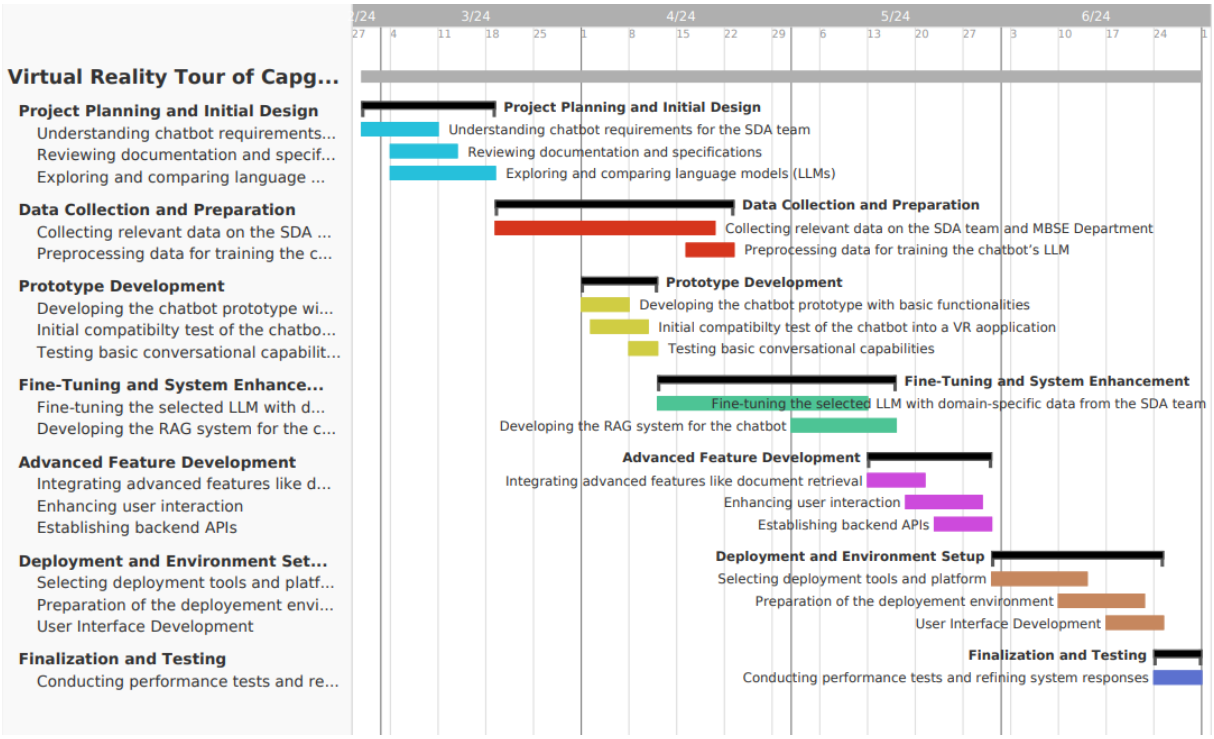

Figure 1.15: Gantt diagram of the project

1.6 Conclusion

In this first chapter, we have defined the contextual and methodological framework of the work on my end-of-study project, which revolves around two axes, namely the presentation of the host organization, and the work methodology adopted. Through the first axis, we have presented the history of the host organization. The second axis focused on the agile Scrum methodology and planning addressed throughout the project.

Chapter 2

Analysis, Modeling, and Theoretical Framework of the Project

Contents

2.1 Introduction

The analysis of requirements is a crucial step in the realization of any IT project. Thus, this chapter will be dedicated to the study of the existing system and the theoretical framework of our project. This approach aims to clarify the vision and simplify the understanding of the various theoretical concepts that will be discussed throughout the following chapters.

2.2 Preliminary Study and State Of The Art

2.2.1 Problem Statement

Virtual reality (VR) and chatbot technologies have the potential to transform how users explore environments and access information. However, traditional methods of navigating complex systems and retrieving relevant data can often be inefficient, limiting user engagement and productivity. This raises critical questions: How can we develop a user-friendly and immersive VR application that simplifies the exploration of complex departments like MBSE (Model-Based Systems Engineering)? Additionally, how can a complementary intelligent chatbot be integrated to provide real-time assistance without interrupting the VR experience?

Our objective is to address these challenges by designing a chatbot that enhances usability and engagement, allowing users to access information quickly and seamlessly. This solution will aim to streamline workflows by providing instant, personalized responses, helping users get the information they need in real time while ensuring that the chatbot adapts to various user needs.

2.2.2 Literature Review

Both virtual reality and conversational AI have seen rapid advancements in recent years, particularly in improving user interactions and accessibility. In the context of Model-Based Systems Engineering (MBSE), these technologies can be leveraged to enhance user experiences during information retrieval and interaction processes.

A prominent study, ***"VR-enabled Engineering Consultation Chatbot for Integrated and Intelligent Manufacturing Services"***, delves into the development of chatbots designed to assist users in complex environments. It highlights how natural language processing (NLP) techniques can empower chatbots to provide accurate, contextually relevant responses, improving real-time assistance. The study emphasizes the value of chatbots in delivering instant feedback and support, which can significantly increase user engagement. Despite these advancements, the research also points out challenges such as real-time processing and the need for continuous learning systems to keep chatbots effective and relevant over time.[2]

Another important finding is the challenge of ensuring seamless interactions with chatbots that do not disrupt the user's flow. By focusing on non-intrusive, intuitive interactions, the

chatbot can become an essential tool for users seeking quick access to information without sacrificing immersion in their tasks. The insights from this research have informed our design process, guiding us to build a chatbot that prioritizes usability and efficiency in delivering relevant information.

In conclusion, the literature highlights both the opportunities and challenges in developing effective chatbot applications. Our approach builds on these findings to create an intelligent chatbot that addresses the current limitations of existing solutions while being scalable, user-friendly, and adaptable to a wide range of user needs.

2.3 Market Analysis of Existing Solutions

In this section, we will analyze the current market landscape for virtual reality (VR) applications and AI-driven chatbots, focusing on their capabilities, limitations, and unique features. This analysis will help us understand the competitive environment and identify opportunities for innovation in our project.

2.3.1 Available Technologies

— AI-Driven Chatbots

— Meya AI

- **Description**: A platform for building and deploying AI-powered chatbots.
- **Key Features**: Integration with existing data sources and APIs for real-time information retrieval.
- **Limitations**: While capable of handling complex queries, it may face challenges with ensuring data privacy and security, which is crucial when dealing with sensitive MBSE department data.

Figure 2.1: Meya AI Logo

— Dialogflow

- **Description**: Google's chatbot development platform that allows for the creation of conversational interfaces.
- **Key Features**: Context management for maintaining meaningful conversations across multiple user interactions.

- **Limitations**: Handling very domain-specific queries, requiring substantial customization for specialized applications

Figure 2.2: Dialogflow Logo

2.3.2 Comparative Statistics

- **Chatbot Technologies:** Dialogflow and Meya AI offer powerful chatbot development tools, with Dialogflow being more suited for global and highly customizable applications. Meya AI provides a more user-friendly interface but comes with data privacy concerns that may limit its use in sensitive environments.

2.3.3 Added Value of Our Solution

Our project aims to address the limitations of existing chatbot technologies by providing:

- **Accuracy**: Our chatbot leverages advanced AI models fine-tuned for the MBSE domain, ensuring that responses are contextually relevant and precise, particularly for complex technical queries.
- **Versatility**: Capable of integrating seamlessly with various platforms (such as a VR application), our chatbot is designed to support multiple use cases, from technical assistance in MBSE environments to general customer support.
- **Consistency**: Our solution maintains high standards of consistency in user interactions, providing uniform and reliable responses that reduce the need for repetitive user queries.
- **Profitability**
  - **Cost Savings**: By automating user support and information retrieval, our chatbot reduces the need for extensive human intervention, leading to significant cost reductions.
  - **Market Positioning**: Our chatbot offers a specialized tool for the MBSE industry, differentiating itself from general-purpose chatbots and positioning itself as a leader in technical and engineering support.

  - **Scalability**: Easily adaptable to different project sizes and requirements.

- **Customer Value**

  - **Time Efficiency**: By delivering accurate and immediate responses, our chatbot reduces the time users spend searching for information, allowing them to focus on more critical tasks.
  - **Quality Assurance**: Ensures high-quality, error-free responses.
  - **Customization**: Our solution is highly customizable, allowing users to tailor the chatbots responses and functionality to meet specific project requirements and preferences.

- **Innovation**

  - **Real-Time Assistance :** Unlike many existing solutions, our chatbot provides real-time, context-aware support that adapts dynamically to the users needs, enhancing the overall user experience.
  - **Feedback Loop :** Continuous improvement based on user feedback and evolving project requirements.

- **Cost Savings**

  - **Operational Efficiency :** Automating responses generation reduces manual effort and associated costs.

This market analysis highlights the strengths and limitations of existing chatbot solutions. By focusing on the unique value propositions of our project, we effectively position our chatbot as a superior alternative, particularly in the MBSE domain, enhancing efficiency, accuracy, and user satisfaction.

2.3.4 Future Possibilities for VR Integration

While the primary focus of this project is the development of a chatbot for enhanced information accessibility and user interaction, there is significant potential for future integration with virtual reality (VR) platforms. By combining the conversational capabilities of the chatbot with immersive environments, we could create a more engaging and interactive experience for users.

Future development could involve integrating the chatbot into VR platforms such as *Virtual Reality Tour of Capgemini Engineering* allowing new employees, clients, and interns to not only interact with the virtual environment but also receive real-time assistance through natural

language interactions. For instance, users exploring the Capgemini Engineering department in VR could ask questions, receive detailed explanations about ongoing projects, or even be guided by the chatbot throughout the virtual tour.

Figure 2.3: Chatbot Integration into VR Application: Prototype Concept

Such an integration could further streamline onboarding processes, offering a fully immersive, interactive, and informative experience that blends visual exploration with conversational AI, resulting in improved user engagement and a more efficient flow of information.

2.4 Needs Analysis

2.4.1 The Horned Beast Diagram

Our need is defined using the "horned beast" (bête à cornes) tool, which allows us to identify the essential characteristics that our chatbot system must have. We then study the validation of these needs and identify the elements that could lead to the elimination of the need[3].

2.4.2 Needs Identification

We identify the need for the chatbot system by addressing the following questions:

Why does this need exist?

There is a critical need for an intelligent chatbot system in the MBSE environment due to several factors:

1. **Complexity of Information :** The MBSE domain involves handling intricate and vast amounts of information. A dedicated chatbot system can streamline access to this information, making it easier for users to retrieve specific data efficiently.
2. **Efficiency Improvement:** Automating responses to frequently asked questions and providing instant access to resources significantly reduces the time users spend searching for information, thereby improving overall productivity.

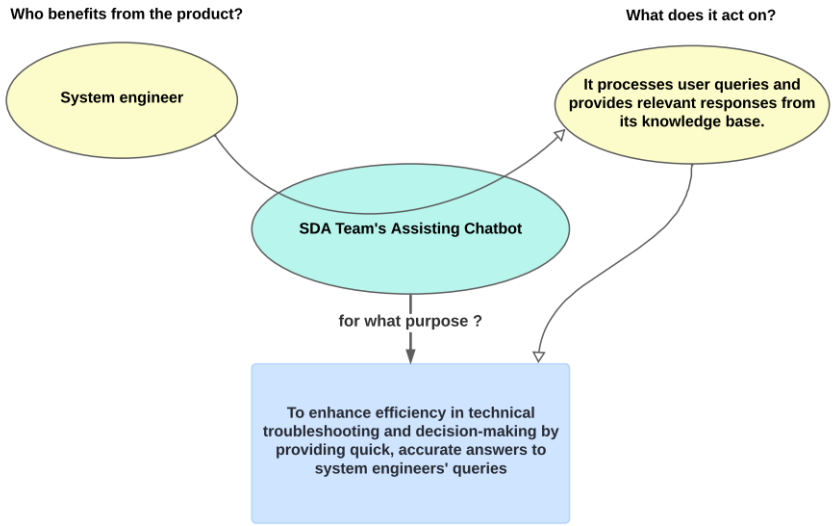

Figure 2.4: Horned Beast Diagram

3. **Enhanced Accuracy:** Leveraging AI for delivering precise and contextually relevant information minimizes the likelihood of errors, ensuring that users receive accurate and up-to-date responses.

4. **User Engagement:** A well-designed chatbot can enhance user engagement by providing a more interactive and responsive experience, especially in environments where users may require immediate assistance.

5. **Advancement of MBSE Practices:** Integrating a specialized chatbot into MBSE practices promotes a more modern and efficient approach to managing complex systems, pushing the boundaries of traditional methods.

What could evolve this need ?

The need for an intelligent chatbot system in the MBSE domain may evolve due to several factors:

1. **Technological Advancements:** Continuous improvements in AI and natural language processing (NLP) technologies could further enhance the chatbots capabilities, making it more efficient and contextually aware.

2. **Expansion of MBSE Applications:** As MBSE applications grow and become more complex, the demand for a sophisticated support tool like a chatbot will increase, driving the evolution of our solution.

3. **Integration with Other Tools:** Integrating our chatbot with other tools and platforms used in the MBSE environment could expand its functionality, making it even more indispensable to users.

4. **User Feedback and Adaptation:** Continuous feedback from users will allow us to refine and improve the chatbot system, ensuring it meets the evolving needs of MBSE professionals.

What are the risks of this need disappearing ?

While the need for an intelligent chatbot system in the MBSE domain is significant, there are potential risks that could lead to this need diminishing:

1. **Improvement in Manual Methods:** If significant improvements are made in manual methods of information retrieval and user support within MBSE departement, the need for an automated solution may decrease.

2. **Alternative Solutions:** The development of alternative solutions that address the same problem more effectively could reduce the reliance on our system.

3. **Changes in Industry Practices:** Shifts in industry practices and standards could impact the need for our specific solution, particularly if new methodologies render our approach obsolete.

However, even if these risks materialize, the fundamental need for efficient, accurate, and scalable support tools in the MBSE domain is likely to remain, ensuring the continued relevance of our chatbot solution.

2.5 Specifications of Requirements

The following sections address the functional, non-functional, and user requirements that allow for a clear understanding of the need and interaction with all stakeholders.

2.5.1 Functional Requirements

Functional requirements describe the features provided by the system. They include:

- **Automated Response Generation:** The ability to automatically generate accurate and contextually relevant responses to user queries related to MBSE departement.
- **Natural Language Processing:** High precision in understanding and processing user inputs in natural language, ensuring accurate and relevant responses..
- **Real-time Assistance:** The capability to provide real-time responses and updates as user queries evolve.

- **Customizability:** Allowing users to customize the chatbots behavior and responses to meet specific needs and preferences.
- **Interactive Interface:** Providing a user-friendly, interactive interface for users to interact with the chatbot, ensuring seamless communication.

2.5.2 Non-Functional Requirements

Non-functional requirements are used to describe the system's operation. They include:

- **Performance:** The system should generate answers within a reasonable time frame, ensuring efficiency.
- **Scalability:** The system must handle increasing amounts of data/queries and complexity without degradation in performance.
- **Security:** Ensuring that all user interactions and MBSE-related data processed by the chatbot are securely stored and accessed, respecting the private nature of the department's data.
- **Compatibility:** The system should be compatible with existing tools and workflows used by systems engineers.
- **Reliability:** High reliability in terms of system uptime and availability.

2.6 System Modeling

The specification of requirements facilitates the understanding of the system requirements through diagrams in a human-readable language. This step is crucial as it translates the needs and functionalities of the system into clear and understandable visual representations for all project stakeholders. By using UML diagrams, we can describe the various aspects of the system in a structured manner, helping to identify interactions, information flows, and dependencies between system components.

In this section, we will present two essential types of diagrams for modeling our system: the use case diagram and the sequence diagram. These diagrams provide an overview of the interactions between users and the system, as well as the interactions between different system components.

2.6.1 UML Language

Unified Modeling Language (UML) is a standardized modeling language used in software and systems engineering. It provides a set of graphic notation techniques to create visual models of object-oriented software-intensive systems. UML is important in projects like ours for several reasons:

1. **Standardization:** UML provides a standardized way to visualize the design of a system, ensuring consistency and understanding across different teams and stakeholders.

2. **Communication:** UML diagrams facilitate communication among project team members, stakeholders, and clients by providing a clear, visual representation of system components and their interactions.

3. **Documentation:** UML serves as a comprehensive documentation tool that captures the details of system design, which is useful for future reference and maintenance.

4. **Problem Identification:** Visualizing the system design through UML diagrams can help identify potential issues and design flaws early in the development process.

5. **Flexibility:** UML supports various types of diagrams (e.g., use case, class, sequence, activity), allowing for detailed and diverse representations of the system's architecture and behavior.

2.6.2 Use Case Diagram

The use case diagram provides a visual representation of the interactions between the user (typically a system engineer) and the chatbot designed to facilitate specific tasks such as answering queries, providing information, and assisting with MBSE-related activities. This section offers a detailed explanation of the diagram, focusing on the actors involved and the various use cases that define the functionalities of the system.

2.6.3 Illustration Of The Use Case Diagram

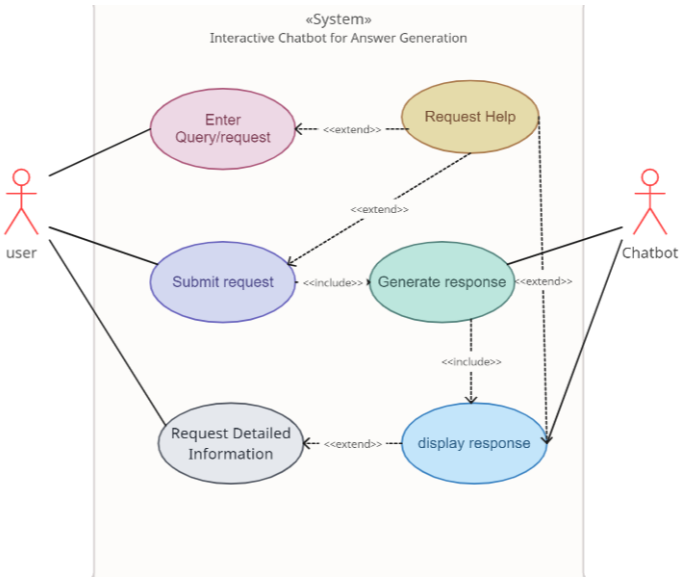

Figure 2.5: Use Case Diagram

—Actors:

- **User :** TThe primary user who interacts with the chatbot (New Employees, potential clients, and interns). This actor is responsible for inputting queries, accessing information, and utilizing the chatbot's assistance for MBSE-related information.
- **Chatbot:** The intelligent agent that processes the users requests, provides accurate information, and offers assistance as needed.

—Use Cases:

- **Enter Query/Request:** The user inputs a specific query or request into the chatbots interface. This could be a question or a request for specific information about the MBSE department.
- **Submit Request:** The user submits the query or request to the chatbot by clicking the "Submit" button. This action triggers the chatbot to process the input.
- **Generate Response:** The chatbot processes the submitted query or request to generate a relevant response. This is the core functionality where the chatbot uses its underlying logic or knowledge base to create a meaningful reply to the user's input.
- **Display Response:** The chatbot displays the generated response to the user. The response could be an answer to the query, a suggestion, or any other relevant information the chatbot can provide.
- **Request Detailed Information:** After the initial response is displayed, the user can ask for more detailed information related to the query. This use case extends the basic response functionality by providing more in-depth or specific details based on the user's needs.
- **Request Help:** At any point, the user can ask for assistance to understand the features and functionalities of the chatbot.

—Relationships:

- The **"Submit Request"** use case includes the "Generate Response" use case.
- The **"Generate Response"** use case includes the "Display Response" use case.
- The **"Display Response"** use case can be extended by the "Request Detailed Information" use case.
- The **"Request Help"** use case can extend from "Enter Query/Request," "Submit Request," "Generate Response," and "Display Response."

The use case diagram provides a comprehensive overview of the system's functionality, highlighting the main tasks the user can perform and the chatbot's role in facilitating these tasks. This enhanced diagram is a crucial component of understanding the interactive processes involved in answering question through the chatbot interface.

2.6.4 Sequence Diagram

The sequence diagram offers a detailed step-by-step visualization of the interactions between the user and the intelligent chatbot. This section describes each action and response in the process of generating responses, emphasizing the dynamic behavior of the system.

2.6.5 Illustration of The Sequence Diagram

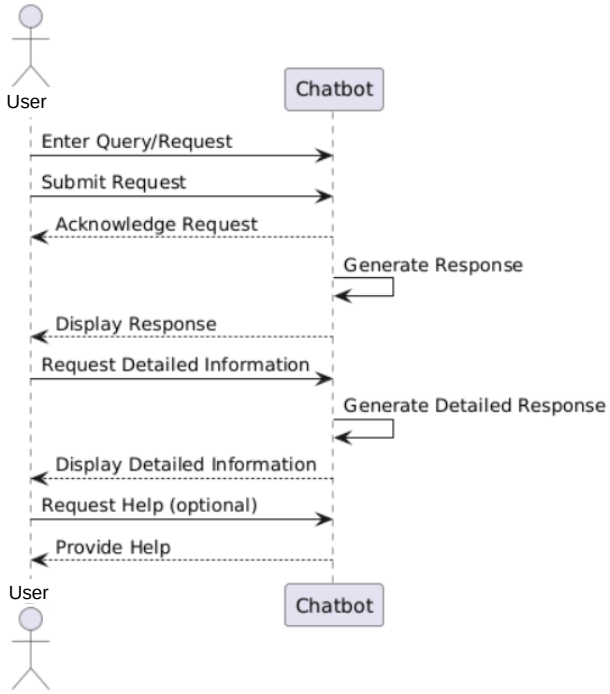

Figure 2.6: Sequence Diagram

—Steps:

1. **Enter Query/Request:** The interaction begins with the systems engineer / user entering a specific query or request into the chatbots interface. This step initiates the communication between the user and the chatbot.

2. **Submit Request:** The systems engineer submits the query or request by clicking the "Submit" button. This action sends the input data to the chatbot for processing.

3. **Acknowledge Request:** The chatbot acknowledges receipt of the request, confirming that the input has been received and will be processed.

4. **Generate Response:** The chatbot processes the submitted query and generates a response based on the provided details. This involves interpreting the input and formulating an appropriate response.

5. **Display Response:** The generated response is displayed on the user interface for the systems engineer to review. This visual feedback allows the user to verify that the response meets their expectations.

6. **Display Detailed Information:** If more information is needed, the systems engineer can request additional details. The chatbot processes this request and generates a more detailed response.

7. **Request Help:** At any point, the systems engineer can request assistance from the chatbot. This feature is designed to help the user understand how to use the system effectively and resolve any issues that may arise.

The sequence diagram provides a clear and detailed depiction of the interactions between the system engineer and the chatbot. By outlining each step in the process, this diagram helps in understanding the dynamic behavior of the system and the sequential flow of actions required to generate and modify responses. This comprehensive view is crucial for both developers and users to grasp the functionalities and workflow of the interactive chatbot system.

2.7 Theoretical Framework

The theoretical framework of this project encompasses several key concepts from the fields of Artificial Intelligence (AI), Machine Learning (ML) and Large Language Models (LLMs). These concepts are fundamental to understanding the approach and methodologies employed in the development of an intelligent tool for response generation from textual specifications (questions).

2.7.1 Artificial Intelligence (AI)

Artificial Intelligence (AI) is a branch of computer science focused on creating systems capable of performing tasks that typically require human intelligence. These tasks include problem-solving, reasoning, learning, perception, and language understanding. AI is categorized into various types, including narrow AI, which is designed for specific tasks, and general AI, which aims to perform any intellectual task that a human can do.

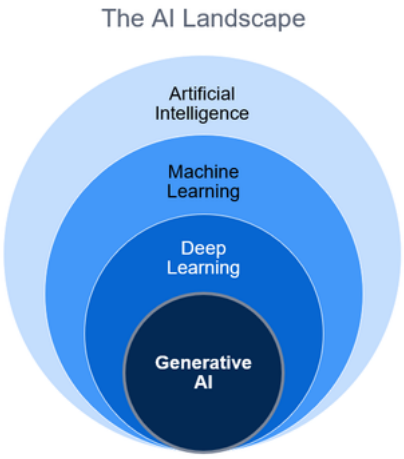

Figure 2.7: The AI Landscape

Key AI Concepts:

- **Natural Language Processing (NLP):** A subfield of AI that focuses on the interaction between computers and humans through natural language. NLP enables machines to understand, interpret, and respond to human language in a valuable way.
- **Generative AI (GenAI):** A branch of AI that involves generating new content from existing data. GenAI models can create text, images, music, and more, by learning patterns from the input data.

2.7.2 Machine Learning (ML)

Machine Learning (ML) is a subset of AI that involves the use of algorithms and statistical models to enable computers to learn from and make predictions or decisions based on data. Unlike traditional programming, where explicit instructions are given to the computer, ML algorithms learn patterns and relationships from the data they are trained on.

Key ML Concepts:

- **Supervised Learning:** A type of ML where the model is trained on labeled data, meaning that each training example is paired with an output label. The model learns to map inputs to outputs.
- **Unsupervised Learning:** A type of ML where the model is trained on unlabeled data and must find patterns and relationships within the data.
- **Reinforcement Learning:** A type of ML where the model learns to make decisions by performing actions and receiving rewards or penalties.
- **Deep Learning:** A subset of ML that uses neural networks with many layers (deep neural networks) to model complex patterns in large datasets.

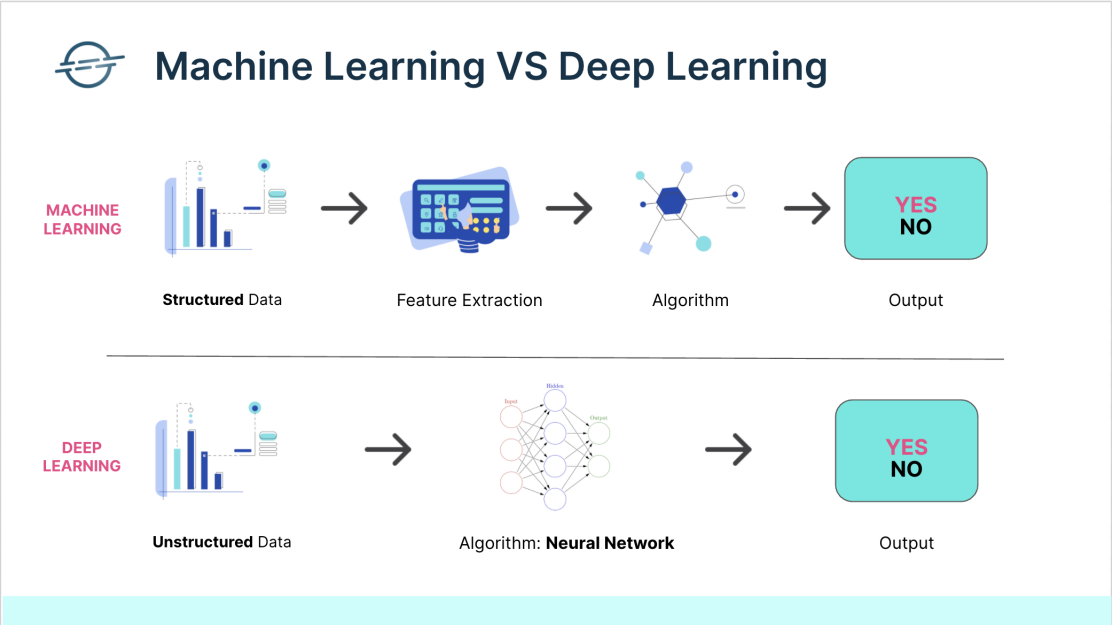

Figure 2.8: Machine Learning vs Deep Learning

2.7.3 Large Language Models (LLMs)

Large Language Models (LLMs) are a type of AI that uses machine learning techniques to process and generate human-like text based on large datasets. These models are capable of understanding and generating human language with a high degree of accuracy and fluency.

Key LLMs Concepts:

- **Pretraining and Fine-Tuning:** LLMs are initially trained on a diverse corpus of text data (pretraining) and then fine-tuned on specific tasks or domains to enhance performance in those areas.

- **Transformer Architecture:** LLMs typically use transformer architecture, which enables the model to capture long-range dependencies and context in text data.

- **Contextual Understanding:** LLMs can understand context and generate coherent text by considering the surrounding text and overall context.

2.8 Transformer Architectures for Text Generation

Although Transformers provide the foundation for many large language models (LLMs), models tailored for text generation still require specific adaptations to maximize their effectiveness. Below, we explore these architectural components:

2.8.1 Encoder-Decoder with Attention Mechanisms

Text-based models often use the Transformer architecture, which includes an encoder and a decoder. The encoder understands the input text, and the decoder generates the output text. Attention mechanisms help the decoder focus on the most relevant parts of the input.

—Masked Attention

Masked attention ensures the model generates text one word at a time without seeing future words. It makes sure the model only considers the words that have already been generated, keeping the text coherent.

—Positional Encoding

Transformers need to understand the order of words. This is done by adding positional information to each word's representation, helping the model keep track of where each word is in the sequence.

2.8.2 Word Embeddings: Capturing Meaning in Text

Word embeddings represent the meaning of words in a high-dimensional space. Models like Word2Vec or GloVe map each word to a vector, capturing its relationships with other words.

—Contextual Embeddings

Unlike static embeddings, contextual embeddings adjust based on the surrounding words. This means the same word can have different meanings depending on its context in a sentence.

—Multi-Head Attention

In Transformers, Multi-Head Attention allows the model to focus on different parts of the input simultaneously. It uses multiple attention mechanisms in parallel to capture various aspects of the input.

2.9 QLoRA: Efficient Fine-Tuning of Quantized LLMs

Fine-tuning large language models can be challenging due to the computational resources required and the risk of forgetting previously learned information. QLoRA (Quantized Low-Rank Adapter) offers a more efficient approach.

Core Principles

- **Quantization:** Reduces the precision of model parameters to save memory. For example, quantizing a 32-bit parameter matrix to 4-bit precision cuts the storage size significantly.
- **Low-Rank Adaptation:** Breaks down large matrices into smaller ones, reducing the number of parameters and making fine-tuning more efficient.

—Methodology

- **Quantization:** Converts the model's parameters to lower precision.

- **Low-Rank Adaptation:** Decomposes these parameters into smaller matrices that are fine-tuned for specific tasks.
- **Training:** Fine-tunes only the smaller matrices while keeping the quantized parameters fixed.
- **Reconstruction:** Combines the quantized parameters and the fine-tuned matrices during inference to produce the final output.

—Advantages of QLoRA :

- **Reduced Memory Footprint:** Significant reduction in memory requirements.
- **Efficiency:** Lower computational resources needed.
- **Scalability:** Allows fine-tuning of large models on less specialized hardware.
- **Performance Retention:** Maintains high performance despite reduced precision.

—Implications and Future Directions :

- **Accessibility:** Makes fine-tuning large models more accessible.
- **Cost-Effectiveness:** Reduces operational costs.
- **Ethical AI Development:** Promotes inclusive and unbiased AI systems.
- **Future Research:** Encourages exploration of further optimization techniques.

QLoRA represents a significant advancement in the fine-tuning of large language models. By combining 4-bit quantization with low-rank adaptation, it addresses the critical challenges of memory consumption and computational efficiency. This innovative approach democratizes access to powerful AI models, paving the way for broader adoption and further advancements in the field.

2.10 Conclusion

In this chapter, we outlined the theoretical framework and development of a question-answering chatbot for the MBSE department at Capgemini Engineering. We highlighted the importance of chatbots in improving communication, productivity, and knowledge sharing within engineering teams.

We reviewed state-of-the-art technologies, focusing on advancements in Natural Language Processing (NLP) and Large Language Models (LLMs), and assessed various chatbot solutions like Dialogflow and GPT-based systems. This analysis helped us identify areas where our solution can add value, particularly in handling complex MBSE-related queries.

We also defined key functional requirements, emphasizing accuracy, scalability, and usability, while incorporating AI concepts like machine learning and transformers into the system's architecture. The project aims to develop an intelligent chatbot capable of delivering accurate, domain-specific answers to enhance team efficiency.

In the next chapter, we will discuss the technical implementation of our solution based on these theoretical insights.

Chapter 3

Technical Study and Preparation

Contents

3.1 Introduction

In this chapter, we will explore the technical methodologies and processes applied throughout the project. The focus will be on the development of a chatbot using advanced AI techniques, including model fine-tuning and Retrieval-Augmented Generation (RAG). These methods were employed to ensure that the chatbot not only understood domain-specific tasks but also retrieved relevant and accurate information from large datasets to deliver context-rich responses.

The combination of fine-tuning and RAG allowed for both the specialization of the AI model and its ability to dynamically access external data, making the chatbot more responsive and adaptable. The following sections will dive into the architecture, data preparation, and the implementation details that contributed to the overall technical success of the project.

3.2 Proposed Approach

Our approach focused on integrating AI into the chatbot and VR system by comparing fine-tuning and Retrieval-Augmented Generation (RAG). Fine-tuning specialized the model for task-specific responses, while RAG enabled dynamic, context-rich replies using external data. The method offering better performance, scalability, and user experience was selected for final integration.

3.2.1 Model Fine-Tuning for Chatbot Development

3.2.1.1 Overview of Fine-Tuning Techniques

Fine-tuning is a crucial step in adapting pre-trained models to specific tasks or datasets. Unlike training from scratch, which requires significant computational resources and large amounts of data, fine-tuning leverages the knowledge already encoded in a pre-trained model. By adjusting the models weights on task-specific data, fine-tuning optimizes performance for more nuanced objectives while retaining the models foundational understanding.[4]

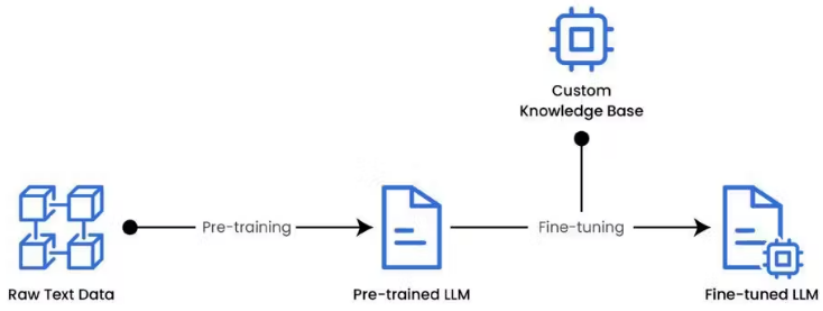

Figure 3.1: Fine-Tuning Process

Why fine-tuning is necessary:

- **Customization:** Every industry or task has unique language patterns and terminologies. Fine-tuning enables the model to learn these domain-specific nuances, making its output more precise and aligned with the specific needs of your use case. Whether in legal, medical, or technical fields, fine-tuning ensures the model generates content that is highly relevant and contextually accurate.
- **Data compliance:** In industries like healthcare or finance, strict regulations require careful handling of sensitive data. Fine-tuning on in-house datasets allows organizations to maintain compliance and security while using LLMs, as the model can operate on proprietary data without exposing sensitive information.
- **Limited labeled data:** Acquiring a large volume of labeled data for specific tasks can be challenging. Fine-tuning allows you to use smaller, task-specific datasets effectively, helping the model adapt to your domain without needing vast amounts of new data.

In short, fine-tuning bridges the gap between general language understanding and the specific requirements of a task, ensuring better performance and more reliable results in specialized applications.

3.2.1.2 Model Selection Process

When developing a chatbot for a Question Answering (QA) system, selecting the right model is critical for balancing performance, speed, and accuracy. Several models were considered based on their architecture, pre-training corpus, and suitability for conversational AI. After thorough evaluation, the following models were chosen for fine-tuning:

3.2.1.3 Model Selection Process

For developing a chatbot specialized in Question Answering (QA), the following models were selected based on their capabilities in handling conversational tasks effectively:

- **LLaMA-2**:
  - *Strength*: Delivers fast, contextually relevant responses with high computational efficiency, making it great for real-time applications.
  - *Reason*: Its lightweight nature ensures responsiveness without compromising quality, suited for large-scale chatbot deployment.
- **Phi-2**:
  - *Strength*: Excellent at structured, FAQ-style interactions due to its balance of speed and precision.

  - *Reason*: Optimized for low-latency responses, making it ideal for environments with constrained resources.

- **T5**:

  - *Strength*: Highly adaptable to various types of queries, producing clear and concise answers.

  - *Reason*: T5s text-to-text approach excels in transforming user inputs into accurate responses, handling both simple and complex questions.

- **Gemma 2B**:

  - *Strength*: Effective in handling domain-specific, technical queries with its robust fine-tuning.

  - *Reason*: Suited for expert-level QA tasks, where detailed and precise answers are essential.

In summary, the selected models LLaMA-2, Phi-2, T5, and Gemma 2B offer diverse strengths for the chatbot's Question Answering tasks, balancing accuracy and performance. After fine-tuning and evaluation, the best-performing model will be chosen for the final implementation.

3.2.2 Retrieval-Augmented Generation (RAG)

3.2.2.1 Retrieval-Augmented Generation (RAG) Architecture

Retrieval-Augmented Generation (RAG) is an advanced framework introduced by Meta in 2020 designed to enhance the performance of large language models (LLMs).

It achieves this by integrating a dynamic and curated database with the LLM, which significantly improves the quality and relevance of its responses. The key components of the RAG architecture include:

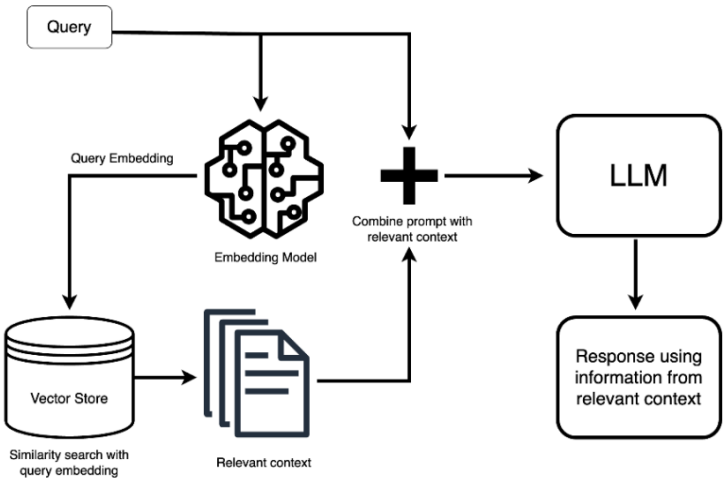

Figure 3.2: RAG Architecture

- **Query Processing**: The process begins when a user submits a query. This initiates the retrieval mechanism of the RAG system.
- **Data Retrieval**: The RAG system searches the database for relevant data based on the query. This involves sophisticated algorithms that match the query with the most appropriate information.
- **Integration with the LLM**: After retrieving relevant data, it is combined with the original query and input into the LLM.
- **Response Generation**: The LLM uses the context provided by the retrieved data to generate a response that is accurate and contextually tailored to the query.

This architecture allows the LLM to access and incorporate up-to-date and reliable information, improving the overall response quality.

3.2.2.2 When to Use Retrieval-Augmented Generation (RAG)?

For most enterprise use cases, RAG is a better fit than fine-tuning because its more secure, more scalable, and more reliable.

- **Enhanced Security and Data Privacy**: With RAG, your proprietary data stays within your secured database environment, allowing for strict access control. In contrast, fine-tuning involves incorporating the data into the models training set, which may expose it to broader access without commensurate visibility.
- **Cost-Efficiency and Scalability**: Fine-tuning a large AI model is resource-intensive, requiring significant time and compute power. RAG leverages first-party data in responses instead of model parameters, which limits resource costs at both the compute level and

eliminates the need for the training stage. This also avoids the lengthy process of crafting and labeling training sets.

- **Trustworthy Results**: The value of AI rests on its ability to deliver accurate responses. RAG excels in this area by consistently using the latest curated datasets to inform its outputs. In case of issues, the data team can more easily trace the source of the response, leading to a clearer understanding of how the output was formulated and identifying where the data went wrong.[5]

3.2.3 Detailed Explanation of Our Approach

3.2.3.1 Input Data Preparation

Our approach centers on using domain-specific question-answer pairs and structured datasets as our primary input data. This method ensures that the AI model can effectively learn and generate accurate responses based on the provided information.

3.2.3.2 Data Collection

We collect diverse question-answer pairs, dialogue transcripts, and domain-specific documents from FAQs, customer support logs, technical manuals, and structured databases. Each dataset is categorized based on its content type, such as general queries, technical support, or structured information.

Each data set is categorized based on the type of questions it represents, such as general information, troubleshooting, and technical support.

3.2.3.3 AI Model Training

The collected data, including question-answer pairs, dialogue transcripts, and domain-specific knowledge, serves as the primary input for training the AI model. We employ advanced Large Language Models (LLMs) for fine-tuning, allowing the model to learn specific patterns in responses. For RAG, the model is trained to retrieve relevant information from a curated database and integrate it with user queries to generate accurate and contextually enriched responses.

3.2.3.4 Response Generation

Fine-Tuning Once fine-tuned, the AI model generates responses based on the learned question-answer pairs and dialogue structures. It provides direct and accurate answers to user queries, relying on the specifics of the training data.

Retrieval-Augmented Generation (RAG) The RAG-enhanced model generates responses by integrating the latest curated data with the users query. It offers up-to-date and contextually relevant answers by leveraging the information retrieved from the database. For data storage, we used Pinecone's vector database, which is particularly suited for advanced NLP applications such as semantic search, document classification, and conversational AI, ensuring efficient retrieval and relevance in responses.

Figure 3.3: Pincone logo

This combined approach ensures that the AI model is proficient in handling a wide range of user queries, improving both the efficiency and accuracy of the chatbots responses.

3.2.4 Data Collection

3.2.4.1 Sources of Data

The sources from which we collected the data include:

- **FAQs**: Frequently asked questions from company websites and support documents.

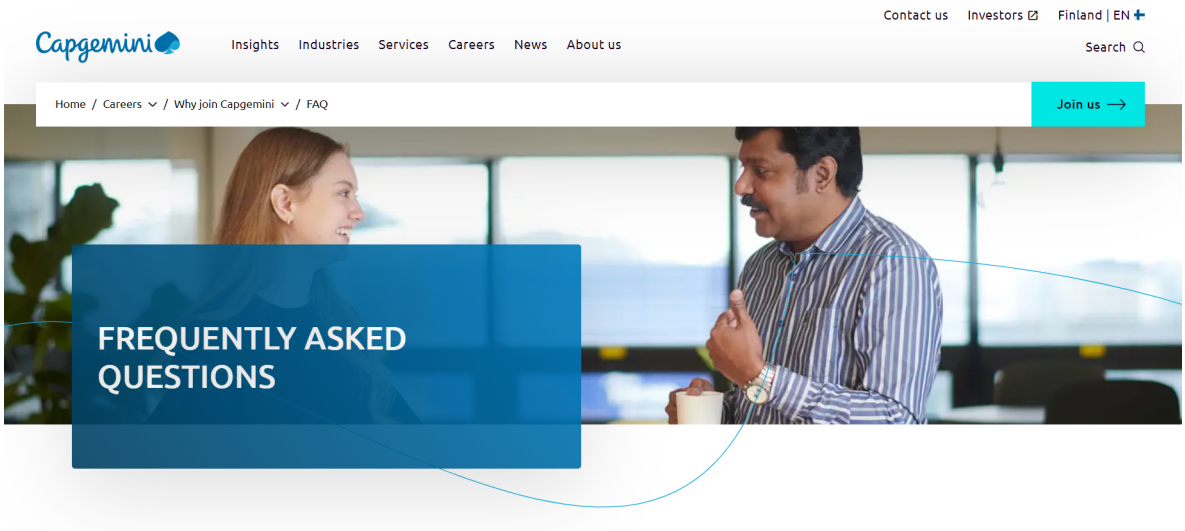

Figure 3.4: FAQ Capgemini's page

- **Questionnaire-Based Data Collection**: We conducted a questionnaire among the SDA team members to collect question-answer pairs. This approach helps us understand the specific expectations and needs of system engineers regarding the chatbot, ensuring that the chatbot's responses are relevant and useful for its intended users.

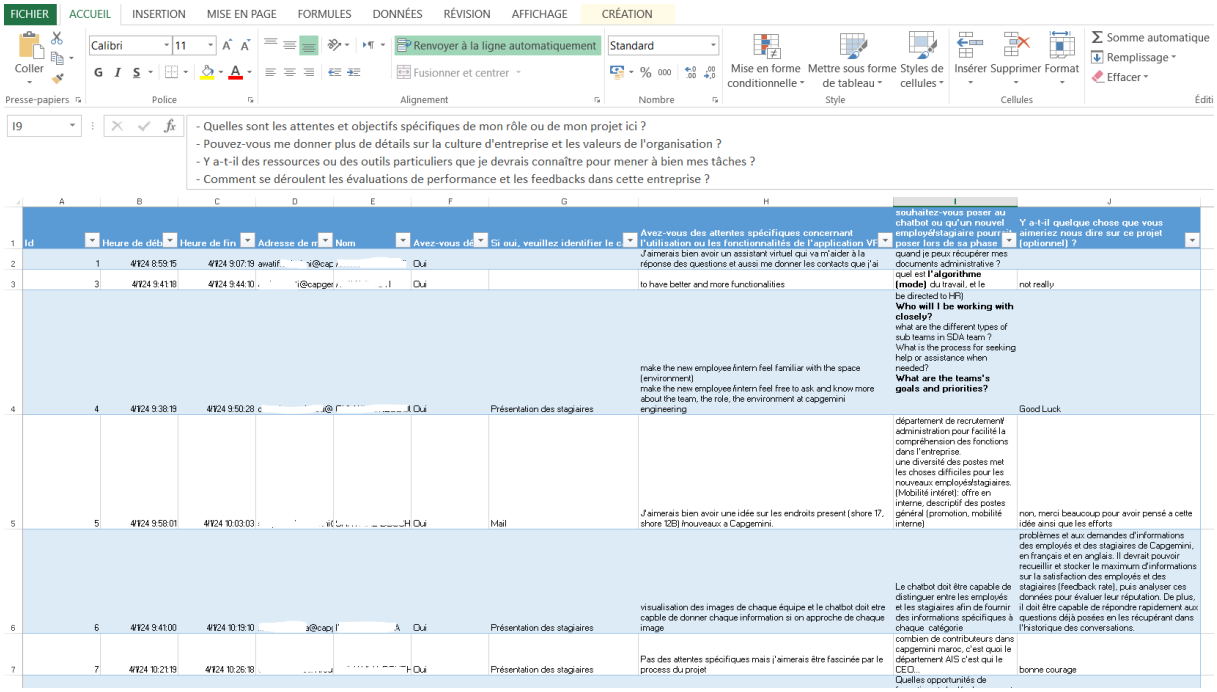

Figure 3.5: Demonstration of Answered Questionnaire

- **Technical and Mandatory training Documentations**: Manuals and product specifications for domain-specific knowledge.

Figure 3.6: Mandatory Training Example documents

3.2.4.2 Data Characteristics and Constraints

Several factors were considered during the data collection to ensure the dataset's quality and relevance for the chatbot:

- **Consistency**: Ensured that question-answer pairs and domain-specific data followed a uniform format for effective model training.

- **Diversity**: Included a wide range of question types and topics to cover various user scenarios and enhance the chatbot's versatility.

- **Relevance**: Focused on domain-specific information to ensure responses are accurate and applicable to the chatbot's use case.

- **Size and Volume**: Collected a large dataset to provide examples for learning and handling diverse queries.

3.2.4.3 Data Collection Process

The data collection process was executed using web scraping techniques, specifically targeting the Capgemini engineering FAQ sections and open source documents from various official websites. Tools such as BeautifulSoup and Requests in Python were employed to systematically scrape and compile relevant question-and-answer pairs to create a dataset for training the chatbot.

Table 3.1: Web Scraping Tools Evaluation and Selection

Tool	Use	Pros	Cons
BeautifulSoup	Parses HTML, perfect for static content.	Lightweight, easy to use, great for simple scraping.	Requires external libraries for HTTP requests, not asynchronous.
Requests	Sends HTTP requests, fetches web pages.	Simple API, fast for retrieving raw HTML.	No parsing capabilities, relies on other tools for data extraction.
Scrapy	Framework for large-scale web scraping.	Asynchronous, built-in data extraction and storage.	Steep learning curve, overkill for simple tasks.

- **Web Scraping:** BeautifulSoup was used to parse HTML content from Capgemini's official FAQ pages and official documents, extracting question-and-answer pairs. The Requests library facilitated the retrieval of web pages, ensuring a smooth and automated scraping process.

Figure 3.7: BeautifulSoup library logo

Figure 3.8: Requests library logo

- **Data Structuring**: The scrapped data was then structured into a cohesive dataset for further processing. This structured approach ensured that the data was ready for pre-processing and model training.

By leveraging these techniques and sources, we were able to assemble a comprehensive and high-quality dataset, laying a solid foundation for the subsequent phases of the project. The diverse and well-structured dataset is pivotal for training our AI model to accurately and efficiently generate answers.

3.2.4.4 Question and Answer Generation Using LLM

To enhance the knowledge base for the chatbot, we employed an LLM-based pipeline to automatically generate question-answer pairs from a set of internal documents. The process starts with the extraction of key documents, which are then passed through a question generation system. This step does not require pre-labeled data; instead, it relies on LLMs to understand the context of the documents and generate relevant questions.

Figure 3.9: LlamaIndex logo

By leveraging `LlamaIndex`'s data generation capabilities, questions are created using a predefined index. This index interacts with OpenAI's `gpt-3.5-turbo` model, which helps in formulating questions that are most likely to match the content and context of the source material.[6]

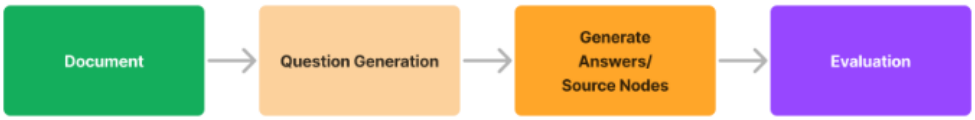

Figure 3.10: LLM-based pipeline for Question and Answer Generation

Once the questions are generated, the next step is to produce answers and link them back to the source nodes or contexts from which they were derived. This is done by querying the document index to retrieve the most relevant information. Both the generated answers and their corresponding source nodes are recorded for evaluation. The final evaluation step ensures that the questions, generated answers, and source contexts are coherent and match the user queries effectively. This process creates a comprehensive dataset that serves as a valuable resource for training and fine-tuning the chatbot system.

3.2.5 Data Annotation

The annotation of data collected from questionnaires and the generated question-answer pairs is essential for preparing the dataset for chatbot training. This involves refining and labeling the data to ensure accuracy and relevance. A combination of automated generation and manual refinement enhances the dataset, making it suitable for future use in Retrieval-Augmented Generation (RAG) or fine-tuning the chatbot model.

3.2.5.1 Overview of the Annotation Process

The primary objective of the annotation process is to enrich the FAQ dataset with structured and meaningful metadata. This involves several steps, starting from collecting the question-answer pairs to generating annotations that can optimize the chatbot's performance during fine-tuning.

Step 1: Data Collection and Question-Answer Extraction

- **Input:** The input consists of question-answer pairs scraped from Capgemini's FAQ sections.

- **Question-Answer Extraction:** During the scraping process, relevant questions and their respective answers are extracted.
  - **Key Elements Extracted:**
    - **Question:** Represents the user query.
    - **Answer:** Provides a relevant response to the users question.
    - **Categories:** Topics such as HR, IT Support, Policies, etc., which help categorize the data.
    - **Intent:** The purpose of the user query (e.g., information request, technical support).

Example:

- **Question:** What is Capgemini's remote work policy?
- **Answer:** Employees are allowed to work remotely for a certain number of days each month, depending on project requirements.
- **Category:** HR
- **Intent:** Information request

Step 2: Annotation Generation

- **Template-Based Description:** For each question-answer pair, specific templates with placeholders for elements like the question, answer, category, and intent are used to standardize the dataset.
- **Filling the Template:**
  - **Question:** The user's query (e.g., "What is Capgeminis remote work policy?").
  - **Answer:** The response provided by Capgemini's FAQ (e.g., "Employees can work remotely for a certain number of days each month.").
  - **Category:** Classifying the question-answer pair (e.g., HR).
  - **Intent:** Tagging the purpose of the query (e.g., "information request").

Example Annotation:

```
{
    "category": "HR",
    "question": "What is Capgemini's remote work policy?",
    "answer": "Employees can work remotely for a certain number of days each month.",
    "intent": "information request"
}
```

Step 3: Refinement and Validation

- **Manual Review:** Each annotated question-answer pair undergoes a manual review to ensure that it is correctly categorized and that the intent is accurately labeled.
- **Refinement:** Adjustments are made where necessary to enhance the clarity, accuracy, and overall quality of the dataset.

Step 4: Integration into Dataset

- **Final Annotation:** After validation, the annotations are formatted and integrated into the final dataset. This structured dataset is then used to fine-tune the chatbot, improving its ability to understand and respond to user queries with accuracy.

3.3 Conclusion

In this chapter, we outlined our AI-driven chatbot's development using both Fine-Tuning (FT) and Retrieval-Augmented Generation (RAG) techniques. We detailed RAG's data retrieval process, which integrates up-to-date information from curated datasets, and highlighted its advantages over fine-tuning for enterprise use, including better security, scalability, and cost-efficiency. We also discussed our data collection process, involving question-answer pairs, domain-specific documents, and FAQs, emphasizing the role of data annotation in ensuring response accuracy. This combined FT and RAG approach provides a robust foundation for a chatbot that delivers precise, contextually rich responses while ensuring reliability and security

Chapter 4

Model Training and Evaluation for Text Generation

Contents

4.1 Introduction

This chapter aims to handle all data-related processes, shed light on the training and testing of models in detail, and then conduct comparative studies to identify the most efficient model.

4.2 Data Preprocessing

Data preprocessing is a crucial phase to ensure the quality and consistency of the dataset used for training the chatbot. For our project, the preprocessing phase involves several key steps to refine the dataset and address any issues that could impact the model's performance.

4.2.1 Initial Dataset Composition

Before preprocessing, our dataset consisted of question-answer pairs, each associated with a specific category and intent. The dataset, as shown in Figure 4.1, includes 1,184 question-answer pairs, distributed across 8 categories and 12 intents. To ensure the model performs effectively, it is essential to augment and preprocess the data. By expanding the dataset and refining the existing pairs, we can improve the model's ability to handle a wider variety of inputs and generate more accurate, contextually relevant responses.

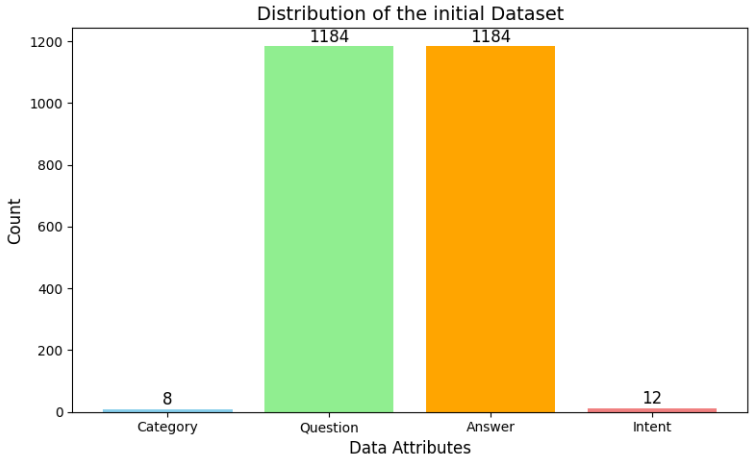

Figure 4.1: Distribution of Data Attributes in the Initial Dataset

Table 4.1: Data Attributes in the Initial Collection

Data Attributes	Count
Category	8
Question	1184
Answer	1184
Intent	12

4.2.2 Duplicate Removal

After collecting the data through web scraping from various sources, we performed an initial cleaning to eliminate any duplicate entries. This step was essential to ensure that each entry in the dataset was unique, thereby preventing potential bias during the training process and ensuring that the model doesn't overfit to repetitive patterns. By maintaining a diverse and representative dataset, we aimed to improve the overall robustness and generalization of the model, allowing it to perform more accurately across various queries.

4.2.3 Data Augmentation

To enhance the robustness and generalization capability of our model, we applied data augmentation techniques. These methods were particularly useful in increasing the variety within our dataset, ensuring that the model was exposed to a wide range of examples, thus preventing overfitting.

Techniques Used

— Easy Data Augmentation (EDA):
We employed four core operations from EDA to increase the variety of our data: Synonym Replacement, Random Insertion, Random Swap, and Random Deletion.

- **Synonym Replacement**: Randomly selected non-stop words in a sentence were replaced with their synonyms. This helped to generate semantically similar but syntactically diverse data, making the model more versatile.

  Original sentence: "Capgemini Engineering offers Model-Based Systems Engineering (MBSE) services."
  Augmented sentence: "Capgemini Engineering provides Model-Based Systems Engineering (MBSE) solutions."

- **Random Insertion**: Random synonyms were inserted into sentences, creating additional variations. *Original sentence:* "The SDA team specializes in developing software solutions for clients."
  Augmented sentence: "The SDA team specializes in software creating solutions for clients development."

- **Random Swap**: Words were randomly swapped within the sentence to further diversify word order.

  Original sentence: "Capgemini's approach focuses on delivering innovative solutions."
  Augmented sentence: "Capgemini's solutions focus on delivering innovative approaches."

- **Random Deletion**: Words were randomly removed, which created more concise variations of sentences, allowing the model to train with both detailed and brief data points.

  Original sentence: "Capgemini Engineering collaborates with clients to deliver complex digital transformation projects."
  Augmented sentence: "Capgemini collaborates clients deliver complex digital projects."

These methods were implemented using Python libraries like `random` and `nlpaug` to generate variations, and `json` for structured data handling.[7]

— Generative Models:
We also leveraged generative language models such as BERT, RoBERTa, and T5 to augment data in a more class-preserving manner. These models encoded class labels with text sequences to generate new samples that aligned with the desired categories.

By applying these data augmentation techniques, including both traditional EDA methods and advanced generative models, we significantly enriched our dataset. This provided a solid foundation for training the AI model to recognize and generate accurate responses in various Capgemini Engineering scenarios.

4.2.4 Final Dataset after Preprocessing

The table below representsThe final dataset, after preprocessing, includes cleaned and balanced Questions and Answers pairs, ready for training. The total number of entries in the dataset is 1500.

Table 4.2: Final Distribution of the Dataset

Data Attributes	Count
Category	8
Question	1500
Answer	1500
Intent	12

JSON Code

```
{
    "category": "HR",
    "question": "Where can I find the HR offices?",
    "answer": "They are on the 4th floor.",
    "intent": "location request"
}
```

4.3 Model Training

4.3.1 Training Configuration

The model was configured with the following parameters:

- **Rank ($r = 16$):** Balances model capacity and efficiency, allowing the model to capture sufficient complexity with manageable computational cost.

- **Scaling factor (`lora_alpha = 16`):** Controls the magnitude of low-rank updates, matching the rank to ensure stability during training.

- **Target modules:** `["q_proj", "k_proj", "v_proj", "o_proj"]` represent the query, key, value, and output projection matrices. Fine-tuning these enhances the model's attention mechanism.

- **Dropout rate (`lora_dropout = 0.05`):** Regularizes the model, reducing overfitting while preserving learning capacity.

- **Bias:** Set to `"none"`, meaning no extra bias terms are added to the low-rank adapters, simplifying the model and improving efficiency.

- **Task type (`"CAUSAL_LM"`):** Indicates fine-tuning for causal language modeling, suitable for sequence generation tasks.

Table 4.3: LoRA Configuration Parameters

Parameter	**Value**
r	16
lora_alpha	16
Target Modules	`["q_proj", "k_proj", "v_proj", "o_proj"]`
lora_dropout	0.05
Bias	`"none"`
Task Type	`"CAUSAL_LM"`

- **Training Arguments:**

Table 4.4: Training Configuration Parameters

Parameter	**Value**
Output Directory	`"Text-to-Text-generation"`
Per Device Train Batch Size	32
Per Device Train Batch Size	8
Learning Rate	2×10^{-4}
FP16 Training	True
Logging Steps	50
Evaluation Strategy	Steps
Save Strategy	Steps
Evaluation Steps	50
Save Steps	50
Load Best Model at End	True
Group by Length	True
Report to	`"wandb"`

—Reporting and Monitoring with Weights & Biases (W&B):

Figure 4.2: Weights & Biases Logo

To ensure effective tracking and visualization of the training process, we integrated our model training with Weights & Biases (W&B). It is a comprehensive platform designed to facilitate the tracking, visualization, and management of machine learning experiments. It offers tools for logging metrics, saving model checkpoints, visualizing training progress, and sharing results with collaborators. By integrating W&B into our training process, we could gain valuable insights into the model's performance and make informed decisions based on the visualized data.

Integration in Our Training Setup: In our training configuration, we specified the **report_to** parameter as "wandb" to enable logging of all relevant metrics and outputs to the W&B platform. This configuration allowed us to monitor the training process in real-time, visualize metrics such as loss and accuracy, and systematically manage multiple training runs.

4.4 Results and Discussion

The results of the training and evaluation are presented in the following plots. These metrics provide insights into the model's performance improvements over the training steps. The models used in this analysis are **LLaMA 2, Flan-T5-Base, Phi 2.7B** and **Gemma 2B** respectively.

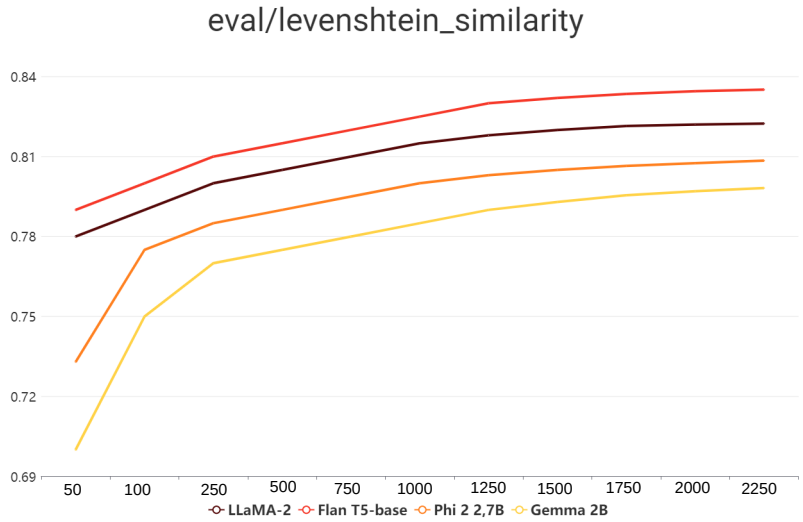

Figure 4.3: Levenshtein Similarity score Across the Four Models

Lets analyze the training results for the four models:**LLaMA 2**, **Flan-T5-Base**, **Phi 2.7B**, and **Gemma 2B**, based on the provided Plot, using two metrics: Levenshtein similarity and loss metrics. The Levenshtein similarity measures how close the generated responses are to the expected outputs by calculating the minimum number of single-character edits required. This metric provides insight into the linguistic accuracy of the models. On the other hand, the loss metrics track the model's performance over time, highlighting the improvements in prediction accuracy as training progresses. By comparing both metrics across the different models, we can assess their effectiveness in handling the task, pinpoint areas of strength, and identify potential weaknesses that may require further fine-tuning.

Levenshtein Similarity

- **Flan-T5-Base** opens with the highest Levenshtein similarity, showing that it starts off producing sequences that closely match the reference sequences. It consistently maintains a lead over the other models, indicating strong initial learning and gradual improvements.

- **LLaMA 2** also performs well, with a steady rise in similarity, closely following Flan-T5-

Base. This suggests that it progressively adapts to the task.

- **Phi 2.7B** and **Gemma 2B** trail behind, with Gemma 2B initially starting slower but steadily catching up as training progresses.

In this scenario, **Flan-T5-Base** stands out as the best performer in both accuracy and training efficiency, closely followed by **LLaMA 2**, with **Phi 2.7B** and **Gemma 2B** requiring more fine-tuning to match the performance of the leading models.

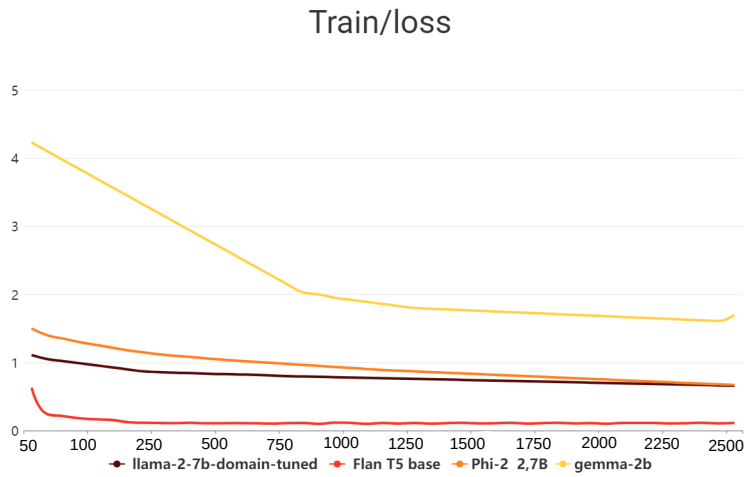

Figure 4.4: Training Loss Progression for Four Models

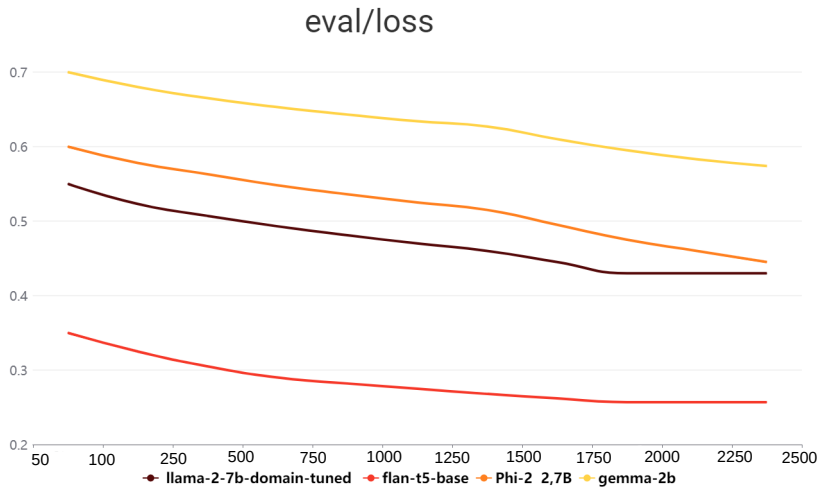

Figure 4.5: Validation Loss Comparison Across Four Models

Training and Validation Loss

- **Training Loss:** All models exhibit a steady decrease in training loss, with **LLaMA 2** and **Flan-T5-Base** showing faster reductions early on. However, **Gemma 2B** maintains a slightly higher loss, suggesting it struggles to minimize training loss as effectively as the others.
- **Validation Loss:** Similarly, in the evaluation loss plot, **Flan-T5-Base** and **LLaMA 2** demonstrate the best generalization, with their validation losses decreasing consistently. **Gemma 2B** shows the highest validation loss throughout, indicating potential overfitting or slower learning.

Summary

- **Flan-T5-Base** leads in both Levenshtein similarity and training efficiency, followed closely by **LLaMA 2**.
- This indicates that **Flan-T5-Base** is more effective at minimizing errors during training
- **Phi 2.7B** and **Gemma 2B** lag behind, with **Gemma 2B** showing a slower convergence.

BLEU Score

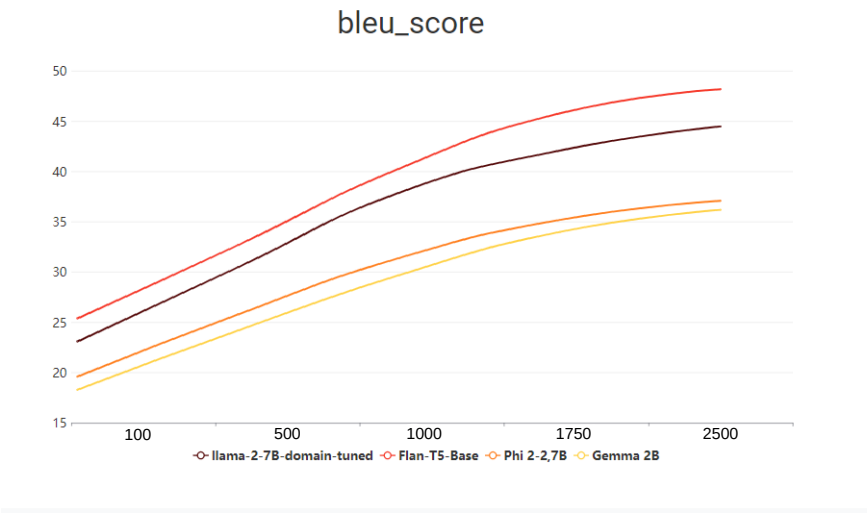

Figure 4.6: The BLEU score plot Across Four Models

- The BLEU score plot demonstrates that **Flan-T5-Base** starts with a higher BLEU score compared to **LLaMA 2**, indicating better initial content overlap with the reference sequences.

- Both models show improvement over time, with **Flan-T5-Base** consistently achieving higher BLEU scores, signifying its ability to produce outputs with superior content alignment.
- **Phi 2.7B** and **Gemma 2B** lag behind in BLEU scores, although they show some improvement, they remain consistently lower compared to **Flan-T5-Base** and **LLaMA 2**.

As for the training, the learning process shows a similar trend to the evaluation process which consists of the four models showing considerable improvement and achieving very promising results both in accuracy and effectiveness.

4.4.1 Metrics summary

Table 4.5: Comparison of LLaMA-2-7B-Domain-Tuned, Flan-T5-Base, Phi-2.7B, and Gemma-2B metrics

Metric	**LLaMA-2-7B-DT**	**Flan-T5-Base**	**Phi-2.7B**	**Gemma-2B**
Learning Rate	0.000016	0.000016	0.000016	0.000016
Train Loss	0.6600	0.1129	0.6700	1.7000
Eval Loss	0.4300	0.2470	0.4450	0.5740
Levenshtein Similarity	0.8224	0.8351	0.8085	0.7982
BLEU Score	44.5	48.2	37.1	36.2

Based on the table above, Flan-T5-Base clearly has the upper hand when it comes to generating text. Not only is the Levenshtein similarity higher indicating a certain level of accuracy when generating new answers, showing in a clear way that the difference between the generated and reference text is clearly shrinking adding the clear aspect of learning which is exactly what we want given the computational limitations at hand .

4.5 Conclusion

In this chapter, we detailed the model development phase and their training. Then, we evaluated them to select **Flan-T5-Base** as the most efficient model for deployment in the following chapter. The training showed a general and reoccurring trend of learning and generalization. As we have already detailed, all metrics prove that the model is capable of generally constructing text (answer) from textual descriptions (question). In the next chapter, we will be delving deep into the deployment process. Discovering the tools and technologies used to access the model for subsequent use in real time.

Chapter 5

Implementation of the Solution

Contents

5.1 Introduction

This chapter focuses on the deployment and implementation of the chosen model. It will discuss the development of the user interface using modern web technologies and the deployment process to ensure our solution is accessible and user-friendly for systems engineers.

5.2 User Interface Development

5.2.1 Frontend Technologies

To create an intuitive and interactive user interface, we leveraged the following frontend technologies:

- **JavaScript**: A versatile programming language that enables dynamic and interactive web content. JavaScript is essential for managing the application logic, handling user interactions, and communicating with the backend server[8].

Figure 5.1: JavaScript logo

- **React.js**: A popular JavaScript library for building user interfaces, React.js offers a component-based architecture. This modular approach allows for the creation of reusable UI components, making the development process more efficient and the interface more maintainable[9].

Figure 5.2: React logo

- **HTML & CSS**: Standard technologies for structuring and styling web pages. HTML provides the structure of the web application, while CSS is used to style and layout the components, ensuring the application is visually appealing and accessible[10].

Figure 5.3: HTML and CSS logos

5.2.2 User Interface Features

The user interface was designed to provide a seamless experience for users, allowing them to input textual requirements and receive corresponding Answers. Key features include:

- **Text Input**: Users can enter natural language requirements in a text input field.
- **Response Display**: The generated Answer is displayed within the application.

5.3 Backend Development and Deployment

5.3.1 Backend Technologies

For the backend, we developed a server capable of processing the user input and generating the corresponding answers. The backend technologies include:

- **FastAPI**: FastAPI is flexible and suitable for a range of web development tasks, including: RESTful APIs: FastAPI is a master at developing RESTful APIs for web and mobile apps.

Figure 5.4: Flask logo

- **Docker**: A containerization technology that packages the application and its dependencies into a single container. Docker ensures that the application runs consistently across different environments, simplifying the deployment process and enhancing scalability[11].

Figure 5.5: Docker logo

- **Hugging Face** : Hugging Face provides tools and platforms specifically designed for deploying machine learning models. It offers a variety of services to make it easier for developers and researchers to host, manage, and serve machine learning models, particularly in the areas of Natural Language Processing (NLP) and related domains.[12]

Figure 5.6: Hugging Face logo

5.3.2 Deployment Process

The deployment process ensures that the fine-tuned model and the entire system are accessible, scalable, and easy to use. We used a combination of Docker for containerization and Hugging Face Spaces for model hosting, along with additional steps for integrating and monitoring the application. The key stages of the deployment process include:

1. **Model Upload to Hugging Face Hub:** The fine-tuned model is uploaded to the Hugging Face Model Hub, which acts as a central repository for machine learning models. This enables easy access to the model and provides an API endpoint for interaction. Using the `transformers-cli`, we push the model to the Hugging Face Hub, ensuring it is properly versioned and documented for future updates.

2. **Backend API Development:** A FastAPI-based backend is developed to handle user requests. FastAPI, known for its high performance and ease of use, allows us to build APIs that accept input text from the user, pass it to the fine-tuned model hosted on Hugging Face, and retrieve the model's response.

3. **Containerization with Docker:** The entire backend, including the API and dependencies, is packaged into a Docker container.

Here's a thorough look at the interface :

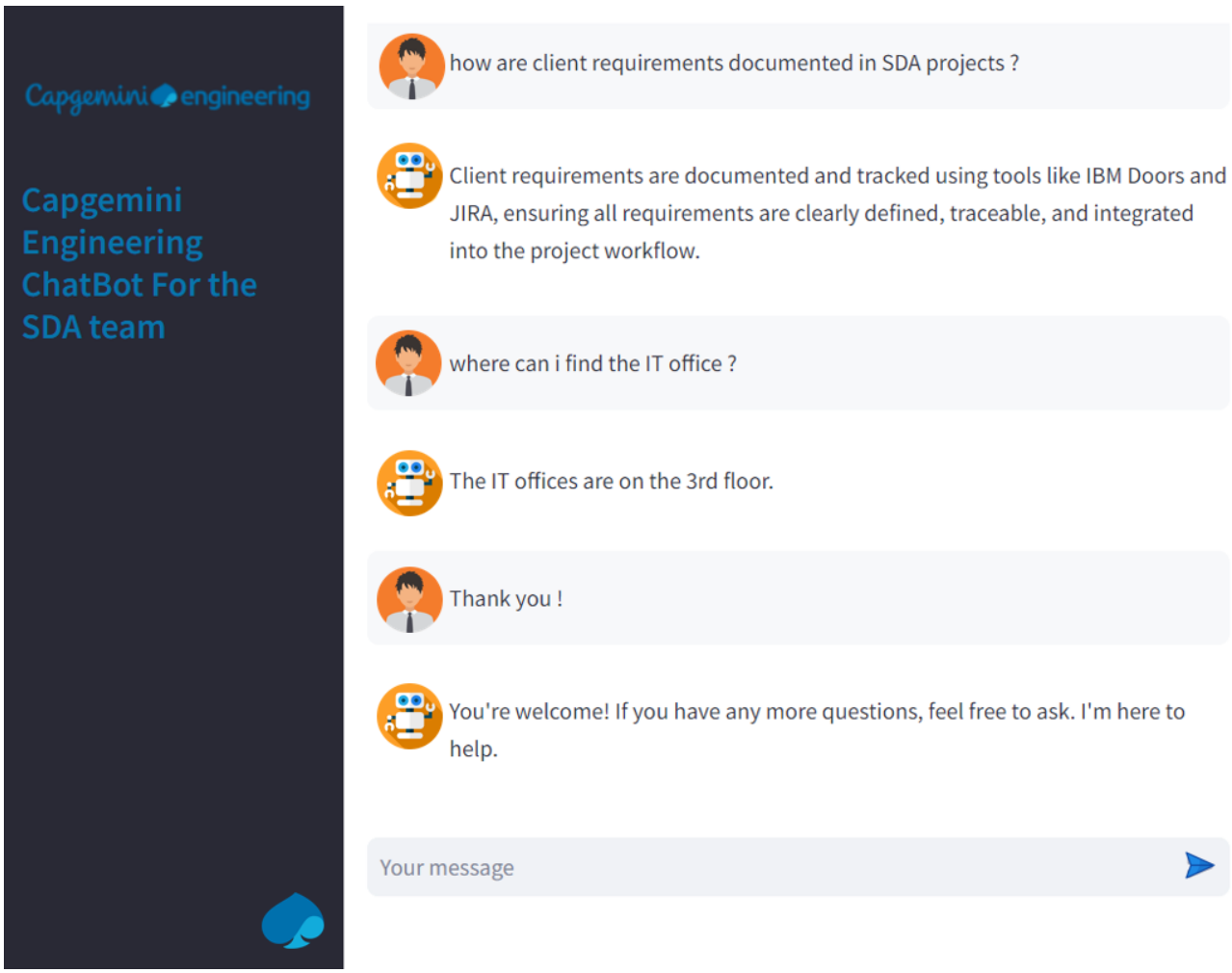

Figure 5.7: Chatbot User Interface

5.4 Conclusion

This chapter outlined the technical steps for deploying the fine-tuned model, integrating the frontend and backend, and using Docker and Hugging Face Spaces for containerization and hosting. The deployment ensures that the solution is accessible, scalable, and maintainable, offering a user-friendly interface for interacting with the machine learning model and generating responses to the user in Real Time.

Conclusion and Perspectives

In my final year project, I had the opportunity to develop an innovative solution to automate the generation of responses for a question-answer chatbot through the integration of Artificial Intelligence (AI). This project, conducted within Capgemini Engineering, addressed the need to modernize and enhance the process of providing information about the company's departments, projects, and teams. The chatbot aims to improve user engagement and offer a richer, more interactive onboarding experience for new employees, clients, and interns.

Leveraging state-of-the-art Natural Language Processing (NLP), Machine Learning (ML), and advanced Large Language Models (LLMs), the chatbot delivers accurate and personalized responses to a wide range of queries. The project employed an iterative development approach, incorporating research, design, and testing phases to ensure its effective integration into future VR applications.

Key achievements of the project include:

- **Successful AI Integration:** The incorporation of NLP and ML techniques has successfully automated the response generation process, achieving the primary goal of enhancing automation through AI.

- **Efficiency and Accuracy:** Preliminary results indicate notable improvements in the chatbots efficiency and accuracy in handling queries.

Throughout the project, I gained valuable experience with new tools and frameworks, bridging my systems engineering background with advanced AI techniques. This experience underscores the growing intersection of AI with various disciplines, shaping the future of engineering and technology.

The project presented several challenges, such as issues with data collection and preprocessing, model selection difficulties, and GPU-related problems during fine-tuning. Overcoming these obstacles involved acquiring new skills and utilizing advanced tools, ensuring the project's successful completion.

Perspectives

Looking forward, there are several promising directions to enhance the projects capabilities and impact:

- **Integration with VR Applications:** Embedding the chatbot system into VR applications could provide a more immersive and interactive user experience.
- **Compatibility with MBSE Tools:** Expanding compatibility with additional Model-Based Systems Engineering (MBSE) tools could improve the overall workflow and value of the solution.
- **Enhanced User Interface:** Developing a more intuitive and user-friendly interface will facilitate better adoption and usability among systems engineers with varying levels of expertise.
- **Advanced Customization:** Offering customization options for generated responses and AI model parameters will provide greater flexibility, addressing the specific needs and preferences of different engineering teams.

This project has demonstrated that AI can significantly improve the efficiency of systems engineering practices, paving the way for AI-powered tools to transform complex system modeling. Addressing these future directions will further develop the system into a comprehensive and versatile tool, advancing the boundaries of AI-driven automation in systems engineering and beyond.

Bibliography

Bibliography

[1] J. BRERETON, "Introducing jira software: the #1 software development tool used by agile teams," October 6, 2015.

[2] A. J. Trappey, C. V. Trappey, M.-H. Chao, and C.-T. Wu, "Vr-enabled engineering consultation chatbot for integrated and intelligent manufacturing services," *Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu, Taiwan*, 2022. 6 February 2022.

[3] E. Electro, "Steps of functional analysis," June 2022.

[4] Turing, "Fine-tuning llms: Overview, methods, and best practices." https://www.turing.com/resources/finetuning-large-language-models, 2023. Accessed: 2024-09-12.

[5] M. Carlo, "Rag vs fine tuning: How to choose the right method." https://www.montecarlodata.com/blog-rag-vs-fine-tuning/, 2024. Accessed: 2024-09-12, Updated: Jul 31, 2024.

[6] R. Theja, "Google colab notebook for evaluating qa systems using llamaindex," May 2023.

[7] S. ES, "Data augmentation in nlp: Best practices from a kaggle master." https://neptune.ai/blog/data-augmentation-in-nlp-best-practices-from-a-kaggle-master, 2023. Accessed: 2024-09-12, Published: 1st September, 2023.

[8] MDN Web Docs, "Javascript," 2024.

[9] React Documentation, "React a javascript library for building user interfaces," 2024.

[10] MDN Web Docs, "Html and css," 2024.

[11] Docker Documentation, "Docker - build, ship, and run any app, anywhere," 2024.

[12] H. Face, "Hugging face hub documentation." https://huggingface.co/docs/hub/index, 2024. Accessed: 2024-09-12.

[13] Saurabhk, "5 data augmentation techniques for text classification." Medium, 2020. Available at: https://medium.com, Published: Nov 20, 2020, Accessed: 2024-09-12.