Republic of Tunisia
The Ministry of Higher Education and Scientific Research

The University of Carthage

The Higher Institute of Information and Communication Technologies

END-OF-STUDY PROJECT REPORT

Submitted in Partial Fulfillment of the Requirements for the
BACHELOR DEGREE IN COMPUTER SCIENCE

Field of Study : Software Engineering and Information Systems

---

Design and Development of an AI-Powered Feasibility Analysis Suite with Integrated Chatbot Assistance for Public-Private Partnership (PPP) Evaluations

---

By
YASSINE HACHANI AND AHMED HAJJEM

Conducted within....

Publicly defended on May 25, 2024 in front of the jury members:

Chairman:	Name SURNAME, University Relations Leader, IBM
Reporter:	Name SURNAME, Teacher, ISTIC
Examiner:	Name SURNAME, Teacher, ISTIC
Professional Supervisor:	Name SURNAME, Engineer, VERMEG
Academic Supervisor:	Name SURNAME, Teacher, ISTIC

Academic Year: 2023-2024

Republic of Tunisia
The Ministry of Higher Education
and Scientific Research

The University of Carthage

The Higher Institute of Information and
Communication Technologies

END-OF-STUDY PROJECT REPORT

Submitted in Partial Fulfillment of the Requirements for the
BACHELOR DEGREE IN COMPUTER SCIENCE

Field of Study : Software Engineering and Information Systems

---

Design and Development of an AI-Powered Feasibility Analysis Suite with Integrated Chatbot Assistance for Public-Private Partnership (PPP) Evaluations

---

By
YASSINE HACHANI AND AHMED HAJJEM

Conducted within....

Authorization of graduation project report submission:

Professional Supervisor:	Academic Supervisor:
Issued on :	Issued on :
Signature:	Signature:

Dedicace

A mes parents,
A ma famille,
A mes amis

Dedicace

A mes parents, A mes frères et sœurs, A mes amis

Remerciements

Je tiens à remercier

Key Terms

- Attention: Statistical apparatus for evaluating the impact of each token fed through an LLM.
- **Cohere:** is a Canadian multinational technology company focused on enterprise artificial intelligence, specializing in large language models.
- **Deep Learning**: The field within ML that is focused on unstructured data, which includes text and images. It relies on artificial neural networks, a method that is (loosely) inspired by the human brain. Deep learning has revolutionized various domains, including computer vision, natural language processing, and speech recognition, by achieving state-of-the-art performance in many tasks.
- **Embeddings:** Numerical representations of words that capture their meanings and relationships in terms of context.
- **GPT (Generative Pre-trained Transformer):** A type of large language model that utilizes transformer architecture and is pre-trained on vast amounts of text data. GPT models are capable of generating coherent and contextually relevant text, making them useful for a wide range of natural language processing tasks.
- **Gemini:** Large language model under ongoing development, Google trained to be informative and comprehensive. It can access and process information from the real world through Google Search.
- **Gemini Pro:** Version within the Gemini family of large language models, created by Google DeepMind.
- **HuggingFace**:French-American startup in the field of Artificial Intelligence founded in 2015, developing tools to use machine learning.
- **Instruction Tuning:** Training a language model to answer different prompts to learn how to answer new ones. This process involves fine-tuning the model's parameters based on the specific prompts provided during training.
- **LLaMa.cpp:** Developed by Georgi Gerganov, LLaMa.cpp implements the Meta's LLaMa architecture in efficient C/C++. It is one of the most dynamic open-source communities around the LLM inference with more than 390 contributors, 43000+ stars on the official GitHub repository, and 930+ releases. LLaMa.cpp is widely used for LLM inference tasks due to its efficiency and active community support.
- **Machine Learning (ML):** A subfield of AI that specifically focuses on pattern recognition in data. ML algorithms learn from data to identify patterns and make predictions or decisions without explicitly programming them for specific tasks.

- **OpenAI:** is a company specializing in artificial reasoning, with a 'capped for-profit' mission, headquartered in San Francisco. Before March 2019, it was recognized as a nonprofit organization.
- **PEFT (Parameter-Efficient Fine-Tuning):** A library for efficiently adapting large pretrained models to various downstream applications without fine-tuning all of a model's parameters because it is prohibitively costly. PEFT helps streamline the fine-tuning process by focusing on specific parameters relevant to the downstream task, thus reducing computational resources and time.
- **Prompt:** The input a user provides to an LLM to elicit a response or carry out a task. It serves as a guiding instruction for the model to generate the desired output.
- **RLHF (Reinforcement Learning with Human Feedback):** A technique that tunes a model based on human preferences. RLHF leverages reinforcement learning algorithms to adjust the model's parameters in response to human feedback, aiming to improve its performance.
- **RAG:** Technique that complements text generation with information from private or proprietary data sources.
- **Transformers**: A neural network architecture that forms the basis of most Large Language Models (LLMs). Transformers are particularly effective in handling sequential data and have been instrumental in advancing various natural language processing tasks.
- **MMLU (Massive Multitask Language Understanding):** A new benchmark designed to measure knowledge acquired during pre-training by evaluating models exclusively in zero-shot and few-shot settings.
- **ARC (A12 Reasoning Challenge):** Conceived by Clark et al. in 2018 as a rigorous test for Large Language Models' (LLMs) question-answering capabilities.
- **HellaSwag benchmark:**Introduced by Zellers et al., stands for "Harder Endings, Longer Contexts, and Low-shot Activities for Situations with Adversarial Generations." It serves as a challenging evaluation for commonsense reasoning abilities in large language models (LLMs).
- **TruthfulQA:**TruthfulQA is a benchmark to measure whether a language model is truthful in generating answers to questions. The benchmark comprises 817 questions that span 38 categories, including health, law, finance, and politics. The authors crafted questions that some humans would answer falsely because of a false belief or misconception.
- **Winograd Schema Challenge (WSC):** Introduced by Levesque, Davis, and Morgenstern in 2011, is indeed a benchmark designed for evaluating commonsense reasoning abilities in natural language understanding systems. It consists of a set of carefully crafted pronoun resolution problems, which were specifically created to be challenging for statistical models relying solely on selectional preferences or word associations.

- **GSM8K:** A dataset of 8.5K high quality linguistically diverse grade school math word problems created by human problem writers. The dataset is segmented into 7.5K training problems and 1K test problems. These problems take between 2 and 8 steps to solve, and solutions primarily involve performing a sequence of elementary calculations using basic arithmetic operations ($+$, $=$, $\times$, $\div$) to reach the final answer. A bright middle school student should be able to solve every problem. It can be used for multistep mathematical reasoning.
- **Artificial general intelligence (AGI):** A theoretical field of AI research that attempts to create software that has human-like intelligence and is capable of self-learning.

Contents

List of Figures

List of Tables

General Introduction

Nowadays, public-private partnership is an increasingly popular procurement option used by governments over the world for multi-faceted project delivery. PPP is a cooperative venture between the public and private sectors for the delivery of a public service through appropriate allocation of resources, risks and rewards.

The evolution of ChatGPT has significantly impacted the world of work, revolutionizing communication and collaboration in the workplace. As an advanced language model, ChatGPT has transformed how professionals interact with technology, enabling more efficient and natural language-based interactions. With its ability to generate human-like responses and provide contextually relevant information, ChatGPT has become an invaluable tool for streamlining various work processes.

Using ChatGPT in public-private partnership work has indeed proven to be beneficial in various ways. By leveraging ChatGPT's capabilities, organizations like Jade-Advisory can effectively communicate large volumes of data and documents, resulting in significant time savings and improved efficiency. Additionally, ChatGPT can be utilized to assess project feasibility by analyzing and interpreting complex information, thereby facilitating informed decision-making.

By leveraging productivity improvements, Jade-Advisory has recognized the significance of integrating AI assistance for its repetitive work processes. This integrated AI assistance enables the organization to efficiently communicate with its extensive data and generate documents. The use of AI not only streamlines repetitive tasks but also enhances the overall productivity and accuracy of document generation. This approach allows Jade-Advisory to focus on more complex and strategic initiatives while ensuring that its data is effectively utilized to produce high-quality documents.

In this context , we want to share our final company-suggested project as well as the process that led to its realization. Our goal for this project is to create a simple and easy-to-use app. It will have a conversational ai assistance and will be used for making reports, evaluating projects, and managing resources. We're excited to share this user-friendly solution with you.

Chapter 1

Project context

1.1 Introduction

In this chapter, we are going to start by describing the host company in which our internship is taking place . We will at that point move our focus to the existing framework by presenting the current solution and its criticisms in order to extract our key objectives that will guide our project towards a transformative solution .

1.2 Host company presentation

In this section , the host company, presented by its logo within the **Figure 1.1** [1], Jade Advisory company is presented based on its accessible informations.

Figure 1.1: Jade Advisory logo

1.2.1 Jade Advisory presentation

Jade Advisory could be a consulting firm, wich was established since 2019 in Tunis and has included an office in London in 2020, specialized in advising Private and Public entities on structuring , offering and managing framework and PPP projects in Africa and the MENA region.It provides technical advice for the improvement of cost effective, sustainable and innovative solutions competently directing its clients in understanding all technical limitations of the ventures all through their whole life cycle. Its primary concern has continuously been to convey a fulfilling yield to its clients in each project.

Figure 1.2 illustrate the Distribution of Jade Advisory project over the Africa and the MENA regions.

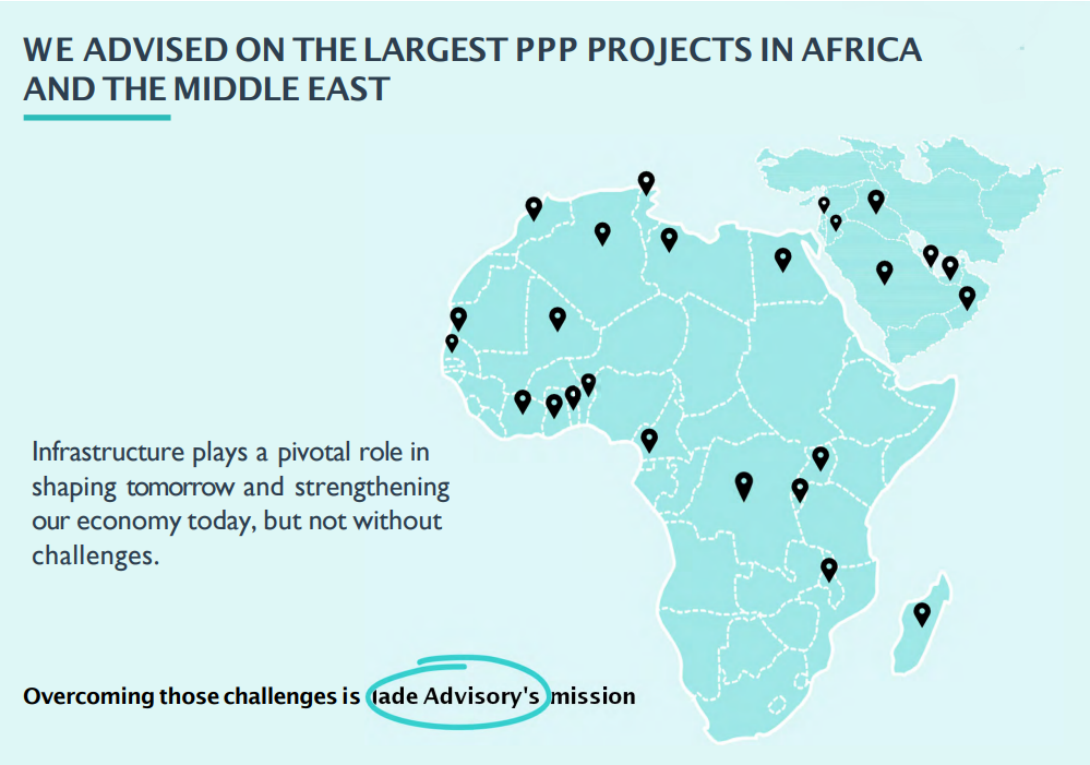

Figure 1.2: Distribution of jade advisory projects in the MENA

1.2.2 Jade Advisory core team

There are 5 people in the team who are experts in setting up projects that involve public and private partnerships in Africa and the Middle East. The different skills of its international team in strategy, consulting, and PPP services help them give advice that helps businesses grow and make more money. The senior team's skills in English, French, and Arabic help them communicate easily and understand different markets.

1. **Name**: Khaled Amri
   Role: Managing Director
   Background: Creator of Jade Advisory. Khaled has over 21 years of experience working as a PPP specialist in Europe, the Middle East, and Africa. He used to work at Ernst and Young MENA as a Director in the PPP Team, the French Railway Authority on High Speed Rail PPPs, Commerbank, and General Company Investment Bank in Paris. He has a Master's degree in civil engineering and another Master's in Project Finance and Structured Finance from Ecole of Ponts ParisTech.

2. **Name**: Mohamed Amine Sdiri
   Role: Director
   Background: Mohamed Amine has worked as a Civil Engineer for 11 years in different areas such as international development, government projects, public-private

partnerships, and transportation systems. He has a postgraduate diploma in management and international relations from Sciences Po Paris and a Master of Engineering from ENIT Tunisia .

3. **Name**: Farouk Bouhafs
   Role: Senior Consultant
   Background: Farouk has a master's degree in Management and Strategy from IHEC Carthage in Tunisia. He has more than 5 years of professional experience in development projects, public private partnerships, project finance and infrastructure advisory. Before joining Jade, Farouk worked as project manager for a consulting company that focuses on helping countries grow and develop.

4. **Name**: Houyem Rais
   Role: Consultant
   Background: Houyem has a bachelor's degree in business administration, with a focus on finance and business analysis, from Tunis Business School TBS . She is a Thomas Jefferson scholarship program (TJSP) alumnus. She started working at Jade Advisory in January 2023.

5. **Name**: Mohamed Chiheb Tlili
   Role: Associate Consultant
   Background: Chiheb holds a Master's degree in finance from Mediterranean School of Business. He recently started working as an Associate Consultant at Jade Advisory.

1.2.3 Sectors of activity

Jade Advisory mainly focuses on different areas, as shown in the following **Figure 1.3** .

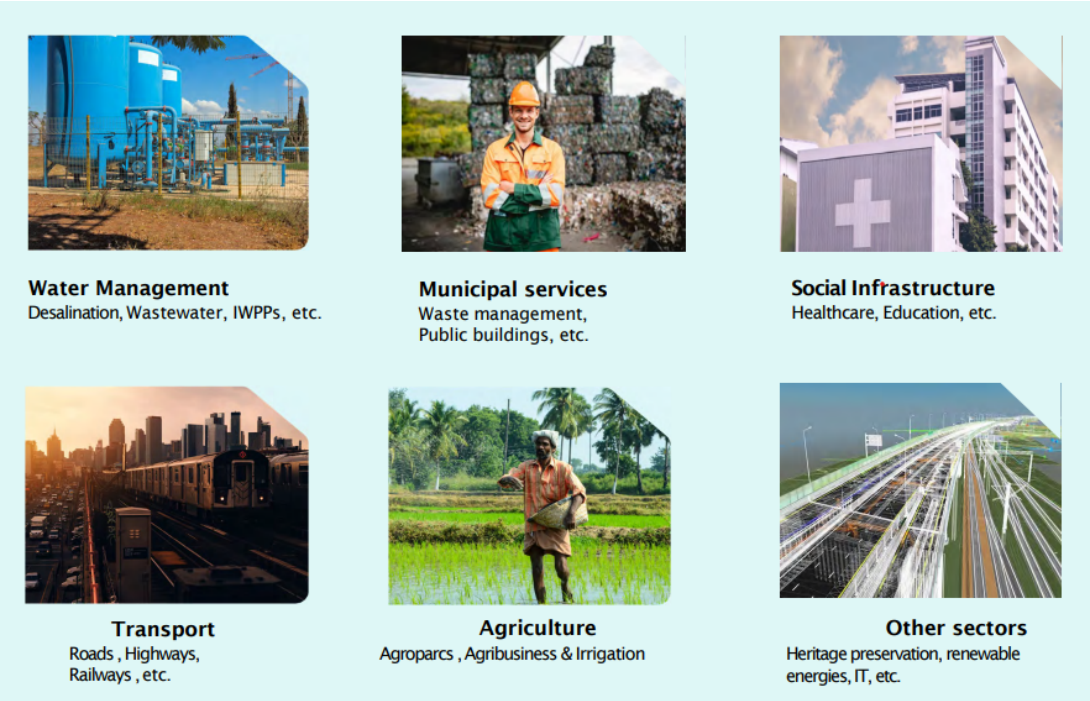

Figure 1.3: Jade Advisory working areas

Infrastructure Advisory in the transport sector

Jade Advisory focuses on providing expert advice on infrastructure for the transport industry. Its serves many different types of clients, such as Private Investors, Governments, International Financial Institutions, and Development Agencies. This includes transportation economics, toll management solutions, lifecycle management, and commercial advisory for people who invest, own, develop, and build physical assets throughout a project's lifespan, as shown in **Table 1.1** .

Table 1.1: Transport Sector Solutions

Transportation Economics	**Toll Management Solutions**	**Infrastructure Project Lifecycle Management**
• Optimizing financial viability and return on investment (ROI). • Identifying financial risks associated with transportation initiatives and developing tailored mitigation strategies. • Financial modeling and analysis that empowers clients to make well-informed decisions regarding their transportation projects.	• Implementing solutions that maximize revenue collection and financial performance for toll-based transportation projects. • Streamlining toll collection systems and processes, enhancing operational efficiency and reducing costs. • Fostering profitable collaboration between public and private stakeholders.	• Crafting detailed financial roadmaps that guide clients through every stage of infrastructure projects, from inception to completion. • Aligning financing strategies with project milestones, ensuring that financial resources are allocated efficiently. • Developing proactive risk mitigation plans, safeguarding the financial health of infrastructure projects and enhancing investor confidence.

Infrastructure Advisory in the water sector

Jade Advisory uses its knowledge to help with water management. They provide specialized advice on infrastructure to Private Investors, Governments, and Development Agencies. Their main areas of work involve making sure water is used in the best way possible, including finding ways to make seawater usable and managing waste water. We

also make sure that our projects are financially sound and that we use water resources efficiently. This can be seen in **Table 1.2** .

Table 1.2: Water Sector Solutions

Water Supply Optimization	**Desalination Solutions**	**Wastewater Management**
• Evaluating water supply projects for economic viability, ensuring sustainable access to clean water for communities • Identifying financial risks in water supply initiatives and developing tailored mitigation strategies • Provide financial modeling and analysis to support innovative water management strategies, optimizing water supply systems for long-term success.	• Implementing cost-effective desalination solutions, providing access to freshwater resources in water-scarce regions. • Streamlining desalination operations, reducing costs and enhancing financial sustainability. • Fostering collaboration between public and private stakeholders to achieve financial success while ensuring reliable desalination services	• Evaluating the financial viability of wastewater management projects, ensuring they meet economic and environmental goals. • Developing financing strategies tailored to the unique needs of wastewater management initiatives. • Promote projects that achieve financial sustainability while advancing responsible wastewater management and environmental protection.

Infrastructure Advisory in the waste management sector

Jade Advisory helps with waste management. They give advice to different kinds of clients, like private investors, governments, financial institutions, and development agencies. This covers turning waste into resources, following environmental rules, being sustainable, and managing finances during projects. We can find more information in **Table 1.3** .

Table 1.3: Waste Management Sector Solutions

Waste-to-Resource Transformation	**Environmental Compliance and Sustainability**	**Infrastructure Project Lifecycle Management**
• Optimizing waste management projects for maximum resource recovery. • Identifying opportunities for cost-effective waste-to-resource transformation, enhancing financial performance for both public and private stakeholders • Providing innovative strategies to convert waste into valuable resources, contributing to a circular economy and financial sustainability.	• Optimizing waste management projects for maximum resource recovery. • Identifying opportunities for costeffective waste-to-resource transformation, enhancing financiaperformance for both public and privatstakeholders • Providing innovative strategies to convert waste into valuable resources, contributing to a circular economy financial sustainability	• Navigating complex environmental regulations and ensuring compliance, mitigating financial risks associated with non-compliance. • Promoting environmentally sustainable waste management practices that reduce long-term financial liabilities and enhance environmental stewardship • Financial modelling and analysis that aligns waste management projects with environmental sustainability

Infrastructure Advisory in the social infrastructures sector

Jade Advisory uses their knowledge to help with important social projects. They offer advice to both private investors and public organizations about infrastructure. This involves making healthcare, schools, and cultural centers better, and improving the community. It helps society grow and be financially stable.

Infrastructure Advisory in the agriculture

Jade Advisory helps the growing agriculture industry by giving advice on infrastructure. This helps Private Investors, Farmers, Smallholders, Rural Communities, Government Bodies, and Development Agencies. This includes the growth of agroparks,

agribusiness, irrigation, and rural development that helps farmers and communities become financially stable and excel in agriculture.

1.3 Project presentation

In this section, we will talk about current state, what's wrong with it, and suggest ways to make things better and help the business do well.

1.3.1 Study of the existing

Public private partnerships PPPs have become very important in the global infrastructure sector. They help reduce costs and address economic challenges in areas like transportation and energy.

However, using the traditional methods to see if PPP projects are possible can cause some problems. These methods can be tiring, costly, and rely too much on experts, old data, and basic financial models. They provide some useful information but may not be advanced enough to handle the complexity and uncertainty of infrastructure.

Additionally, as the number of project files and documents increased, organizations like Jade Advisory realized they need for better ways to work together and manage resources. That's why they started using platforms like OneDrive for easier teamwork and improved management. Jade Advisory decided to use ChatGPT as an AI assistant in their work, which can help them better conduct studies and manage project documents. This accreditation involves tasks like reviewing documents, feasibility studies, and asking for advice from experts.

1.3.2 criticism of the existing

Although public private partnerships PPPs offer promising solutions for building infrastructure, traditional feasibility study methods have many criticisms because they are inefficient and have limitations.

First, using old methods to do feasibility studies with lots of manual work is really hard. These methods are usually slow, require lots of work, and are prone to mistakes. They also cause delays and inefficiencies in decision-making. Relying on expert opinions can lead to subjective and biased results, instead of objective and thorough analysis.

Besides, traditional approaches prioritize economic and neglect other important things like social impact, environmental sustainability, and technical development. This limited perspective hinders a full understanding of project feasibility and overlooks potential risks and opportunities that may come from non financial factors.

Howerver, The absence of automation for everyday tasks like creating reports still causes problems with projects and makes stakeholders less productive.

Also, ChatGPT's lack of user interaction makes it less efficient and slows down decision-making. Participants may have trouble understanding and using the information given

by the AI assistant, which could decrease their effectiveness.

Moreover, the limited access to ChatGPT makes it hard for people with technical skills or disabilities to fully take part in the study. However, the way ChatGPT deals with issues relating to the project, wich is careful and respectful of privacy concerns. Without following strict safety rules, people involved may hesitate to work with AI, which can make the feasibility study process less trustworthy and efficient.

In summary, current methods for assessing the feasibility of PPPs have problems like being inefficient, subjective, and only focusing on economic factors. These limits, with difficulties linked to combining data, automating tasks, engaging users, making things easy to access, and protecting privacy. They emphasize the need for a big change in how we make decisions, moving towards more thorough and objective methods.

The **Table 1.4** shows some possible problems with OneDrive and ChatGPT.

Table 1.4: oneDrive and ChatGPT cons

Software	Cons
	• Weak data privacy. • Data Vulnerability. • Synchronization Limits. • Limited Backup Functionality.
	• Provides Inaccurate Information. • Biased Responses. • Limited Knowledge.

1.3.3 Proposed soultion

To overcome problems with current methods offeasibility assessment in PPPs, we suggest new solutions that use cutting-edge technology to make decisions faster, fairer, and more effective.

First of all, Our solution focuses on making it easy for Chat assistance and documents to work together smoothly, allowing data to be exchanged and synchronized seamlessly. This integration will take off the need for people to manually use ChatGPT, make the feasibility study process easier, and boost overall performance. besides, a strong data integration process will be put in place to protect the privacy and confidentiality of important project information.

Secondly, we will use technology to make tasks like making reports easier by automating them. Using machine learning algorithms, our solution will help automate comparing reports, saving time and reducing mistakes. This automation will increase productivity so that stakeholders can focus on important tasks that need human skill.

Thirdly, our solution will have an easy-to-use interface that works well with ChatGPT alternative. We focus on making it easy for people to chat with our assistance, ask questions, and interpreting insights. This better connection will bring together users and make sure that participants can fully utilize the power of AI.

Our goal is to make sure that our solutions can be used by all people with different skills and abilities. We will give training and support materials to help everyone effectively use the solution.

Our solution will also have strong privacy and confidentiality protections to keep sensitive business data safe. Use encryption, access controls, and audit trails to keep data safe and secure during the analysis process. Make sure to focus on protecting data. Build trust with stakeholders to successfully implement the solution.

In summary, our idea offers a complete way to fix the problems with current methods in PPPs. By focusing on combining data, automation, user engagement, accessibility, and privacy, we want to change how feasibility analysis works and help stakeholders make better choices.

1.4 Conclusion

We have introduced the host company, highlighting the key team members and main areas of work. Moreover, we have included a presentation of our project, in which we have discussed the current issues that have received criticism and then put forward our proposed solution. Finally, in the next chapter, we will look at the project's requirements.

Chapter 2

Needs Analysis

2.1 Introduction

The success of any work depends on the quality of its start. As a first step in our project, it is necessary to analyze the system requirements, which is the objective of this chapter.

In the first part, we will study the needs and present the goals of our project by specifying the functional and non-functional requirements, as well as the external units that interact with the system. In the second part, we will present the development tools that allowed us to carry out this project and the working method used to prepare this report. Finally, we will focus on defining the use case diagram..

2.2 Functional needs

To ensure that our solution meets the expectations of users, it is crucial to identify all the functional requirements that the system must fulfill. These requirements, which are presented in use case diagram in **Figure 2.1**, define the features that the system must provide to the user. Therefore, before starting the development of PPP AI assistance , it is imperative to ensure that all functional requirements have been clearly defined and documented.

- **Chat With Private Document:** This feature allows users to have private conversations with the AI assistant, ChatGPT alternative, while securely sharing and receiving private documents. Users can interact with document assistance in real time for help, ask questions and receive personalized recommendations, all in a secure and private environment.
- **Feasibility Document Generation:** This functionality led to viable generation for public-private partnership (PPP) projects. Using pre-defined machine learning models and parameters, the system analyses project data, financial metrics, and other relevant data to generate detailed feasibility reports. Users can customize document content and structure to their specific needs, simplifying document organization and ensuring accuracy and consistency in reports.
- **Authentication:** The authentication process provides users with authorized access, ensuring privacy and data integrity is protected by establishing a secure framework

for real-time communication. This provides easy access private conversations and sharing of secure documents between stakeholders. Users authenticate themselves for traceability, control and personalization purposes.

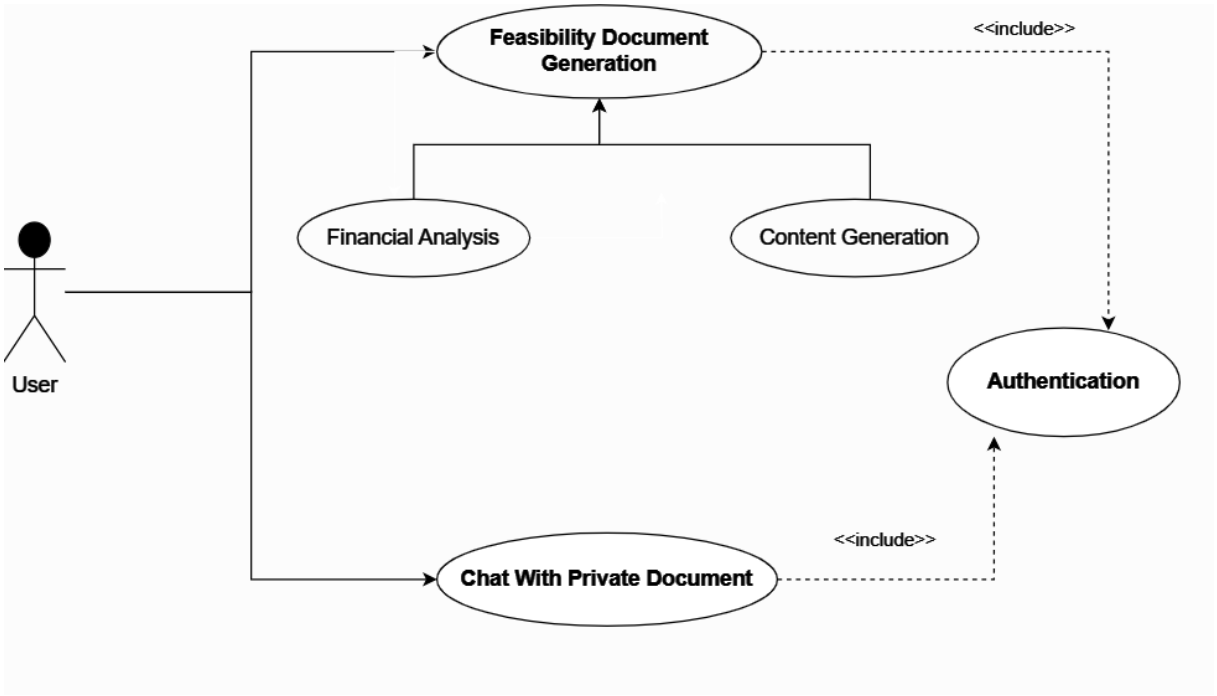

Figure 2.1: Use case diagram

2.3 Technical specification

2.3.1 Non-functionnal needs

The system must address non-functional requirements that are not essential to its operation but crucial for ensuring service quality and smooth system functioning. Non-functional requirements are internal system requirements essential for achieving our objective. To achieve this, the following requirements must be met:

- **ergonomics**: Our product must feature an easy-to-use, user-friendly interface that optimizes interaction between humans and the system, even for inexperienced users. Users should be able to navigate the system effortlessly and access its functionalities without requiring any technical training.
- **performance**: Its crucial that our system must be optimized for efficiency to ensure fast response times and efficient handling of requests by reducing latency qnd improving resource utilization
- **Reliability**: The system should be highly reliable, with minimal downtime and disruption.The AI assistance should possess contextual awareness of the PPP project and deliver precise results consistently.

- **Scalability**: The system should be scalable, capable of efficiently adapting to future demands and user growth without compromising performance.
- **Security** : Our system prioritizes robust security measures to safeguard sensitive data, prevent unauthorized access, and maintain compliance with industry regulations.

2.3.2 Hardware environment

In total, our hardware environment consists of two laptops, as shown in the following **Table 2.1**.

Table 2.1: Hardware environment

Specification	HP Pavilion Gaming Laptop	Dell laptop
Operating System	Microsoft Windows 11 Famille Unilingue	Not specified
CPU	Intel Core i5-10300H (2.50 GHz, 4 cores, 8 logical processors)	Intel Core
RAM	32 GB total physical memory + 36.5 GB virtual memory	Not specified
GPU	NVIDIA GeForce GTX 1650 Ti	MX330
Storage	500GB	500GB

2.3.3 Software environment

Our adopted software are Python, LangChain, Ollama, PostgreSQL, Docker, Streamlit and LangFuse, as shown in the following table 2.2.

Table 2.2: Software environment

Software	Description
	Python is a high-level programming language that is widely known for being beginner-friendly with an active community contributing to open-source projects.

Continued on next page

Software	Description
	PostgreSQL is an open-source relational database known for robustness and versatility.
	Ollama allows you to run open-source large language models locally.
	LangFuse is an open-source platform designed for engineering with Large Language Models (LLMs).
	Streamlit simplifies data science app creation by focusing on Python skills instead of web development.
	Docker is a platform as a service product that uses OS-level virtualization to deliver software in containers.

2.4 Conclusion

In summary, we've addressed both functional and non-functional needs, ensuring user satisfaction and system performance. Then we have discussed the Hardware and software environment considerations in order to satisfy the non-functional requirements. In the next chapter, we will delve into exploring the landscape of artificial intelligence.

Chapter 3

State of Artificial Intelligence

3.1 Introduction

In this chapter, we discuss the foundation of our AI assistance, the large language model, which is like the powerful engine of our system. Meanwhile, it is very important to understand the current state of artificial intelligence in order to use its capabilities properly. Therefore, we explore the landscape of artificial intelligence in its current form to contextualize the development of our AI assistance.

3.2 Large language model (LLM)

3.2.1 Definition

At its core, a large language model is a type of computer program that can understand and create human language using neural networks. The main job of a language model is to figure out the chances of a word coming next in a sentence. For instance, in the sentence "The sky is ....." the most common answer would be "blue". The model can guess the next word in a sentence by looking at a big collection of text. Basically, it learns to recognize patterns in the words. You get a pre-trained language model from this process

The following **Figure 3.2.1** [2] illustrates the layers within the field of Artificial Intelligence.

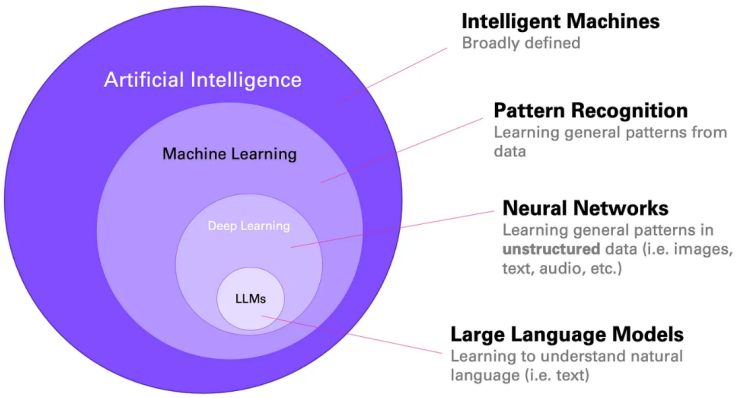

Figure 3.1: The field of Artificial Intelligence in layers

3.2.2 History

During the 1990s and early 2000s, the AI industry mostly worked on small projects that were not too complicated or time-consuming. This can be seen in **Figure 3.2.2** [3]. Let's quickly talk about the history of LLMs.

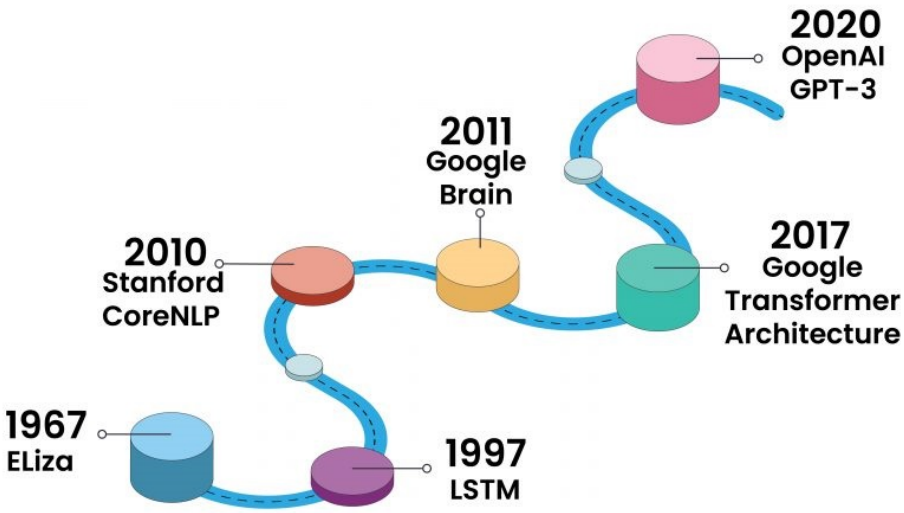

Figure 3.2: History of Large Language Models

When the ancients carefully recorded their knowledge on scrolls of papyrus and housed them in the legendary Library of Alexandria, they could not have even dreamed it possible that all that knowledge and more would be available at the fingertips of their descendants millenniums later. That's the power and beauty of large language models. Not only can

LLMs answer questions and solve complex problems, but they can also summarize huge volumes of work, as well as translate and derive context from various languages.

The beginnings of big language models can be traced back to experiments with neural networks and neural information processing systems in the 1950s to help computers process and use natural language. Researchers at IBM and Georgetown University collaborated to develop a system that can translate phrases from Russian to English automatically. Research in machine translation started from there and gained a lot of recognition.

The concept of LLMs started with Eliza, the first chatbot created by MIT researcher Joseph Weizenbaum in the 1960s. Eliza started the study of natural language processing NLP , laying the groundwork for advanced LLMs in the future. Then after about 30 years, in 1997, Long Short Term Memory LSTM networks were invented. Their arrival led to more advanced and sophisticated neural networks that could process larger amounts of data. Stanford's CoreNLP suite, launched in 2010, made it possible for developers to analyze feelings and identify entities in text.

In 2011, a smaller Google Brain with better features like word embeddings helped NLP systems understand context better. This was a big moment, with models becoming popular in 2017. Think, which is short for Generative Pre-trained Transformer, can create or "decode" new writing. Another example is BERT - Bidirectional Encoder Representations from Transformers. Wich analyze input text to make predictions or categorize it based on encoder parts.

From 2018 onwards, researchers concentrated on creating bigger and bigger models. In 2019, Google researchers introduced BERT, a large model with 340 million parameters. It can understand context in both directions and can be used for different tasks. By training BERT on many different types of data without supervision, the model learned the relationships between words. BERT quickly became the top choice for tasks involving understanding and using human language. Actually, it was BERT that was responsible for every English query on Google Search.

3.2.3 Types of Large Language Models (LLM)

LLMs can be divided into three types pre-training models, fine-tuning models, and multimodal models. Different options have different benefits, depending on what you want to achieve:

- **Pre-training models**: are trained on huge quantities of data, which helps them comprehend a broad range of language patterns and constructs. A plus is that a pre-trained model tends to be grammatically correct!
- **Fine-tuning models**: are pre-trained on a large dataset and afterward are fine-tuned on a smaller dataset for a specific task (use case). They're particularly good for sentiment analysis, answering questions, and classifying text.
- **Multimodal models**: combine text with other modes, such as images or video, to create more advanced language models. They can produce text descriptions of images and vice versa.

3.2.4 Phases of training Large Language Models (LLMs)

Large language models go through three main stages: pre training, fine-tuning for specific tasks, and learning from feedback from humans, as shown in **Figure 3.2.4**[4].

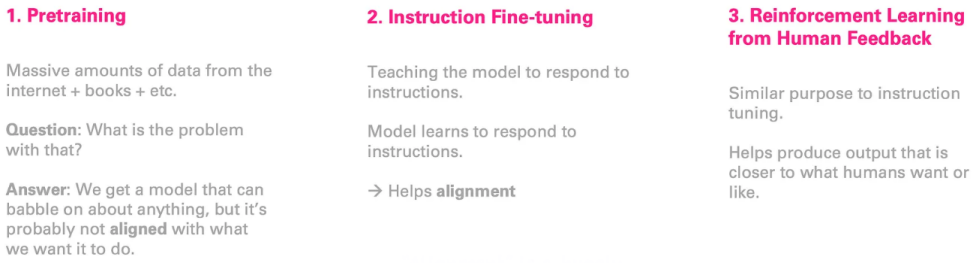

Figure 3.3: Phases of training LLMs

Pre-training

This step needs a massive amount of data in order to train for predicting the next word. During this time, the model learns how to use language correctly and also gains knowledge about the world and develops new abilities.

However, training a large language model only teaches it to talk a lot about a topic, which can make it produce impressive results but not very good at answering specific questions or following instructions. The LLM has not learned to be an assistant, but instead to complete input sequences. For example, when someone asks "What is your first name?", a pre trained LLM might answer with "What is your last name?" because it has been trained on lots of empty forms. The model is having a hard time understanding instructions because it is not used to seeing a pattern where an instruction is given and then a response is expected. At this point, the LLM does not match up with what people want, which is a important concern for LLMs. However, it can learn to follow commands with more practice. Despite some challenges at first, pre-trained LLMs can be controlled and taught to follow directions and understand human goals.

Instruction fine-tuning

This is where instruction tuning comes in to adjust instructions. We use the pre-trained LLM with its current abilities and continue to predict one word at a time. This time, we only use high quality training data with instruction and response pairs.

That way, the model un-learns to only complete sentences and learns to be a helpful assistant that follows directions and responds in a way that matches what the user wants. The size of this instruction dataset is usually much smaller than the pre-training set. high-quality instruction response pairs are very costly to make because they usually human annotated. This is very different from the cheap self-supervised labels we used in pre-training. This stage is also known as supervised instruction fine-tuning because of this reason.

Reinforcement Learning from Human Feedback (RLHF)

At its core, RLHF uses human feedback to create a dataset of human preferences. This helps determine the reward function for a specific output. People can give feedback in many ways:

- **Order of preference**: People rank outputs in order of preference.
- **Demonstrations**: Humans write preferred answers to prompts.
- **Corrections**: Humans edit a model's output to correct unfavorable behaviors.
- **Natural language input**: Humans provide descriptions or critiques of outputs in language. Once a reward model has been created, it's used to train a baseline model with the help of reinforcement learning, which leverages the reward model to build a human values policy that the language model then uses to produce responses. ChatGPT is a good example of how a large language model uses RLHF to produce better, safer, and more engaging responses

RLHF is a weighty step forward in language models, giving users a better and more reliable experience. However, there is a tradeoff, RLHF brings in the biases of the people who provided the data used to train the reward model. While ChatGPT aims to provide useful, honest, and self-confident answers, the way these answers are understood by the annotators can vary. RLHF improves consistency, but it reduces creativity and diversity of ideas. There is still a lot to discover in this area.

3.2.5 Most Large Language Model Application

At first look, LLMs can be used in any situation where a company needs to analyze, process, summarize, rephrase, edit, transcribe or etract insights from a set of data or input text. With adoption increasing, there are some useful ways to use language models that seem to be very helpful.

Translation With Language Models

One simple way to use LLMs is to change written words into a different language. A person can type in words to a chatbot and ask it to change them to a different language, and the chatbot will start translating the words automatically.

Content Creation

Another very common use for language models is creating content. LLMS allows people to create many different types of written content, like blogs, articles, short stories, summaries, scripts, questionnaires, surveys, and social media posts. The quality of these outputs relies on the provided in the initial prompt.

If LLMs are not used to create content, they can still be used to come up with new ideas. According to Hubspot, 33% of marketers who use AI use it to come up with ideas or inspiration for marketing content. The key benefit here is that AI can make the process of creating content faster. There are tools like DALL E, MidJourney, and Stable Diffusion that let you make pictures from written prompts.

Search

Many users will first have experimented with generative AI as an alternative search tool. Users can ask a chatbot questions in natural language and will receive an instant response with insights and facts on potentially any topic.

Using tools like Bard or ChatGPT to search for information gives you access to a lot of different information, but not all of it may be correct.

Language models often make mistakes and can come up with false information. It is important for users to verify any facts shared by LLMs to avoid being misled by misinformation.

Detecting and Preventing Cyber Attacks

Another interesting use of language models in cybersecurity is identifying cyberattacks. LLMs are able to analyze big sets of data from a company's network and can detect patterns that show a harmful cyber attack and send a warning.

So far, some companies are testing new technology to find and stop cybersecurity threats. For instance, earlier this year SentinelOne introduced a solution driven by LLM that can automatically search for threats and start automated responses of malicious activity.

Another way shown by Microsoft Security Copilot, lets users check their systems for known weaknesses and attacks, and can create reports about possible security issues in a few minutes so that people can react quickly.

Code Development

Generative AI tools can create both natural language and code in languages like JavaScript, Python, PHP, Java, and C#.

The language models can help non-technical users to create simple code. Can write basic code for simple projects, but has difficulties with larger and more complex tasks that are bigger in scope and scale.

Programmers need to carefully review their code for any problems with how it works or how secure it is before they finish working on it. This will help prevent any issues from happening after the code is in use. They can also be used to help fix problems in code or create documentation automatically, saving users time.

3.2.6 Large Language Model tools

large language models like LlaMA have various versions, such as llama2 7b, llama2 13b, and llama2 70b. These versions reflect the model's size and capacity, with larger models having more parameters and being able to handle more complex tasks. The purpose of versioning is to offer different sizes and capabilities of the model to fit different needs and uses. For instance, a small model like llama2 7b could work well for a chatbot, while a

larger model like llama2 70b might be better for generating content. Versioning allows developers to pick the best model for their needs, balancing resources and requirements.

Open-source LLM Models

Table 3.1: Overview of Various Language Models

Model	Developer	Key Features	Theoretical Context Window	Application
LlaMA 2	Meta AI	Ranges from 7 billion to 70 billion parameters, designed for reasoning, coding, proficiency, and knowledge tests.	4096 tokens	Reasoning, coding, proficiency tests
BLOOM	BigScience	A multilingual LLM with 176 billion parameters, covering 46 natural and 13 programming languages.	2048 tokens	Multilingual understanding, translation
BERT	Google	A transformer-based model focusing on bidirectional training, with a deep understanding of language context.	512 tokens	Natural language understanding, context analysis
OPT-175B	Meta AI Research	A model with 175 billion parameters, designed for zero- and few-shot learning, boasting a low carbon footprint for its training.	2048 tokens	Zero-shot learning, efficient training
Xgen-7B	Salesforce	Excels in processing up to 8,000 tokens, suitable for detailed conversations and comprehensive summarization.	8000 tokens	Conversations, summarization

Continued on next page

Model	Developer	Key Features	Theoretical Context Window	Application
Falcon-180B	Technology Innovation Institute	A causal decoder-only model with 180 billion parameters, supporting multiple languages and excelling in various tasks.	2048 tokens	Language tasks, multilingual support
Mistral 7B	Mistral AI	A model with 7.3 billion parameters, optimized for English language tasks and coding, efficient in resource usage.	4096 tokens	English tasks, coding
CodeGen	Not specified	Focused on program synthesis across multiple languages, transforming English prompts into executable code.	2048 tokens	Program synthesis, code generation
Mixtral 8x7B	Mistral AI	An enhanced version of Mistral with improved performance and efficiency for a broader range of applications.	4096 tokens	Broad applications, enhanced performance

3.3 Technologies and Models Choice

3.3.1 Comparison Table: LlamaIndex vs LangChain

LlamaIndex is made for creating search and retrieval application, as shown in **Table 3.3.1**. It helps you search for information and find the right documents easily using LLMs. LlamaIndex is the best choice for projects that focus on making a search and retrieval application , efficient, prioritizing efficiency and simplicity while concentrating on specific tasks. Our project aims to create an application that can work well with different software, is easily adjustable, and can grow as needed. Langchain is a great option for this. Nevertheless, each option is unique in how it works and what it's meant for, making sure it's a perfect fit for our particular requirements.

LangChain's robust ecosystem offers a lot of help and resources, making it a great place to create advanced chatbots. In fact, LlamaIndex is based on LangChain, which shows how important it is in developing language models. This, along with the direct connection to LangSmith, makes sure that LangChain provides a complete solution for making chatbots that can give top-quality, contuextually relevant responses.

Table 3.2: Feature Comparison: LlamaIndex vs. LangChain

Feature/Aspect	LlamaIndex	LangChain
Overview	A tool for efficient information indexing and retrieval, used with AI models for enhanced retrieval.	A framework for building language models, focusing on easy integration with external knowledge sources like RAG.
Pros	- Efficient indexing and retrieval - Scalable for large datasets - Integrates with various AI models	- Designed for language applications - Easy integration with tools like LangSmith - Supports RAG - Active community and updates
Cons	- Focused on indexing; additional tools needed for chatbots - More setup for language tool integration	- More complex initial setup - Requires familiarity with its ecosystem

3.3.2 Comparison of Open-Source LLM Models

The decision between Mistral and Mixtral models depends on the type of equipment you have and what the project requires. Mixtral 8x7B is great for NLP tasks that need a lot of parameters. It works well even when you have limited resources, like in **Table 3.3**. It performs really well. On the other hand, Mistral 0.2V 7B is a good choice for projects with standard hardware capabilities. It gives a good balance between performance and efficiency.

Table 3.3: open-source llm model Specifications

Model	Parameters	Hardware Requirements	Performan	Use Cases	Context Window
Mixtral 8x7B	56B	High	Very high accuracy	Demanding NLP tasks	Large
Mistral 0.2V 7B	7B	Moderate	Balanced performance	General NLP tasks	Moderate
Llama2 13B	13B	Moderate to High	High performance	Detailed language understanding	Moderate to Large
Llama2 7B	7B	Low to Moderate	Solid performance	Development and testing	Moderate

3.3.3 Performance Benchmarking of common Language Models

Table 3.4 [5] shows how different language models perform on different tests. These models come in different designs and sizes. They are tested on various benchmarks to see how well they can do tasks like reading, reasoning, and understanding language.

The performance metrics include Average Score, ARC AI2 Reasoning Challenge , HellaSwag, MMLU Mean Multi Labeling Unweighted , TruthfulQA, Winograd Schema Challenge Winogrande , and GSM8K General Language Understanding Evaluation Benchmark . The **Table 3.4** focuses on showing the strengths and weaknesses of various language models when it comes to understand different types of natural language tasks.

Table 3.4: Model Comparison

Model	Average	ARC	HellaSwag	MMLU	TruthfulQA	Winogrand	GSM8K
mistralai/Mixtral-8x7B-Instruct-v0.1	72.62	70.22	87.63	84.88	71.16	64.58	81.37
meta-llama/Llama-2-70b-hf	67.87	67.32	87.33	83.74	69.83	44.92	54.06
tiiuae/falcon-40b	58.07	61.86	85.28	81.29	56.89	41.65	21.46
mistralai/Mistral-7B-Instruct-v0.2	65.71	63.14	84.88	77.19	60.78	68.26	40.03
meta-llama/Llama-2-13b-hf	55.69	59.39	82.13	76.64	55.77	37.38	76.64
meta-llama/Llama-2-7b-hf	50.97	53.07	78.59	74.03	46.87	38.76	14.48
tiiuae/falcon-7b	44.17	47.87	78.13	72.38	27.79	34.26	4.62
Salesforce/codegen-16B-nl	42.59	46.76	71.87	32.35	33.95	67.96	2.65
Salesforce/codegen-6B-nl	40.00	42.32	68.59	25.93	34.47	66.46	2.20
Salesforce/codegen-6B-multi	32.43	27.22	41.11	25.71	45.65	53.91	0.99

3.3.4 Comparison of Embedding Models

We have chosen the BGE Embedding Model as our preferred model for embedding because it offers high efficiency, performance, and a large context window, as shown in **Table 3.3.4**. This choice is made because can handle difficult NLP tasks that need a deep understanding of context. The model's ability to work well with different types of

hardware helps it meet many needs for different applications, which is why it's the best choice even with different hardware requirements.

Table 3.5: Embedding Models Specifications

Model	**Parameters**	**Efficiency**	**Performance**	**Use Cases**	**Context Window**	**Hardware Requirements**
Ada002	Small	High	Quick, lightweight tasks	General NLP tasks	Small	Low-end GPUs/CPUs
BERT-base	110M	Moderate	Solid performance for a variety of tasks	Wide range of NLP tasks	512 tokens	Consumer-grade GPUs
GPT-2	1.5B	Low	High-quality text generation	Text generation, conversation	1024 tokens	High-end GPUs
Nomic Embedding Model	Varies	Varies	Exceptional across tasks	Large context window tasks	8194 tokens	Varies; scalable to hardware

3.3.5 Retrieval Augmented Generation (RAG)

RAG (Retrieval-Augmented Generation) empowers LLMs by fetching relevant information from external knowledge bases, enhancing their responses with factual grounding as shows the following **Figure 3.3.5**[6].

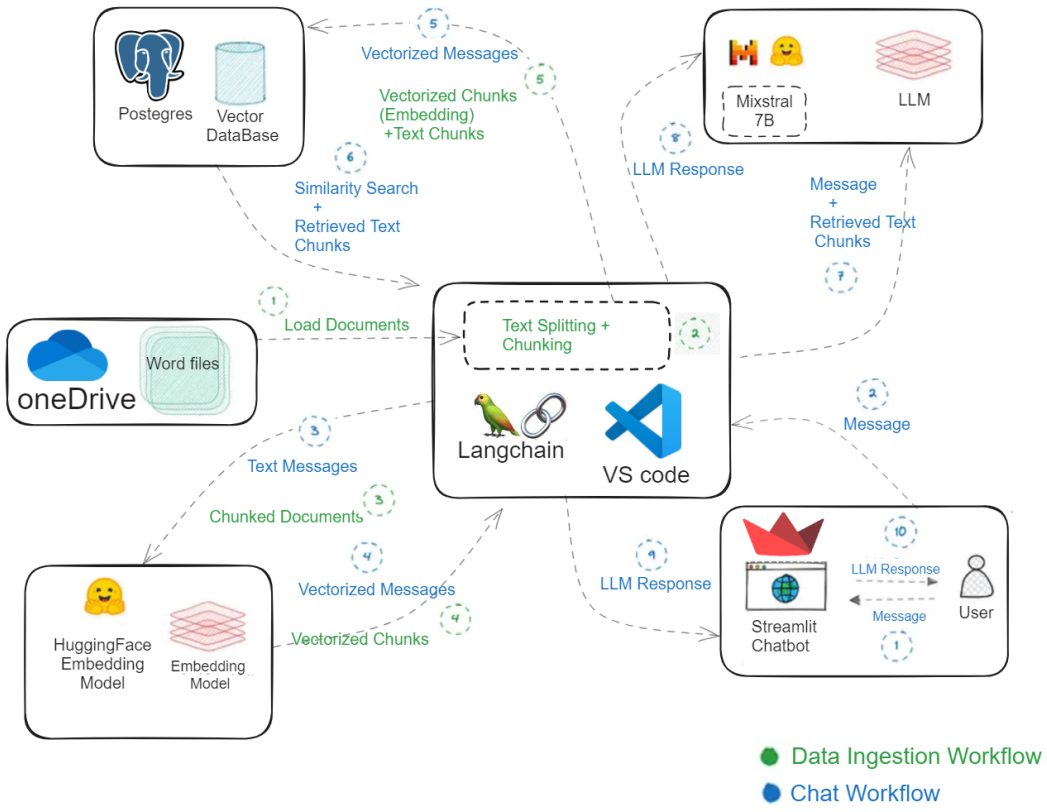

Figure 3.4: RAG pipeline

The process begins with loading a huge data base or corpus containing relevant information . This information base might contain of content from diverse sources such as books, articles, websites, or any other organized or unstructured data stores.

Once the data base is loaded , it s divided into reasonable chunks or sections to encourage effective recuperation . Chunking guarantees that the retrieval process is versatile and doesn t overload the framework with excessive data at once.

Each chunk of information is at that point encoded into a numerical representation, ordinarily utilizing methods like word embeddings or more progressed methods such as transformer based models like BERT (Bidirectional Encoder Representations from Transformers) . This vectorization process converts the literary data into high-dimensional vectors that capture semantic and relevant connections between words and expressions.

when a query is received , it's also vectorized utilizing the same encoding method as the information base. This grants for effective closeness computation between the query vector and the vectors representing chunks of information inside the information base.

Retrieval procedures like approximate nearest neighbor search or other similarity based methods are then employed to recognize the foremost significant chunks of information that match the query vector. chunks of information are recuperated , they are concatenated to make the setting or background data for the discussion . This setting gives the fundamental foundation for creating coherent and pertinent responses.

The concatenated context, along with the primary query , serves as input to a large language model such as GPT Generative Pre-trained Transformer or a comparative architecture .

large language model creates responses based on the input context and query . By leveraging the pertinent information recovered from the vectorestore. However, the responses are not only fluent and coherent but also grounded in retrieved context.

The last response , created by the language model is then displayed to the client or coordinates into the conversational interface.

at the end of the process, the generated response is displayed to the client or integrated into the conversational interface. In general, the Rag system consistently integrates the processes of data recovery and generation to create contextually pertinent and coherent responses in conversational applications.

3.4 RAG Applications

Retrieval-augmented generation models have demonstrated versatility across multiple domains. Some of the real-world applications of RAG models are :

- **Advanced question-answering systems**:RAG models can power question-answering systems that retrieve and generate accurate responses, enhancing information accessibility for individuals and organizations.
- **Content creation and summarization**: RAG models not only streamline content creation by retrieving relevant information from diverse sources, facilitating the development of high-quality articles, reports, and summaries, but they also excel in generating coherent text based on specific prompts or topics.
- **Conversational agents and chatbots**:RAG models enhance conversational agents, allowing them to fetch contextually relevant information from external sources. This capability ensures that customer service chatbots, virtual assistants, as well as other conversational interfaces deliver accurate and informative responses during interactions. Ultimately, it makes these AI systems more effective in assisting users.
- **Information retrieval**:RAG models enhance information retrieval systems by improving the relevance and accuracy of search results.

3.5 Conclusion

In conclusion, this chapter has provided a detailed explanation of LLM and a summary of the history of AI. It talks about different types of LLM, how they are trained, and where they are used. It also introduces various tools for LLM and compares Langchain with LlamaIndex, discusses other competing language and embedding models. Finally, this chapter has included the RAG pipeline of our chatbot and its main application.

In the next chapter, we will present the chatbot process of the augmented search generation in detail.

Chapter 4

Chat with PPP data

4.1 Introduction

This chapter explores the innovative application of Large Language Models (LLMs) combined with Public-Private Partnership (PPP) data in project management. To improve the feasibility analysis process, AI-driven chat capabilities are integrated with PPP data, offering stakeholders a dynamic, interactive, and informed decision-making tool. Additionally, a Retrieval Augmented Generation (RAG) model is employed to enhance the precision and relevance of the responses generated, specifically tailored to meet the unique needs and complexities of PPP projects.

4.2 Problem

PPP projects create a lot of information, like financial and performance reports. Studying all this old information to find helpful ideas is a big task, important for improving upcoming partnerships. Traditional data analysis methods may not be enough because of too much data, which can cause important information to be missed or not used effectively.

Furthermore, data from old projects becomes less useful without being able to apply lessons to new rules, market changes, and technology progress. As a result, important choices, making sure things go smoothly, and planning future projects are affected, which makes it harder for future PPP initiatives to be successful.

To solve this problem, we need a new idea that can analyze a lot of old information and find important information quickly. Chatbots, especially those powered by advanced AI algorithms, are becoming a very good solution. They can automate the analysis of big sets of data, provide immediate insights, and adjust learnings to the specific situation of new PPP projects. By using chatbots, people can make the most of their old data and make better plans for future projects. But we must choose the best type of chat bot to use based on our information.

4.3 The Dataset

Our dataset,a sample was shown at **Figure 5.3**, taken from PPP projects, is mostly semi-structured, with Word documents that have text, tables, and images. This type has unique features and information that make it difficult to manage and analyze data efficiently.

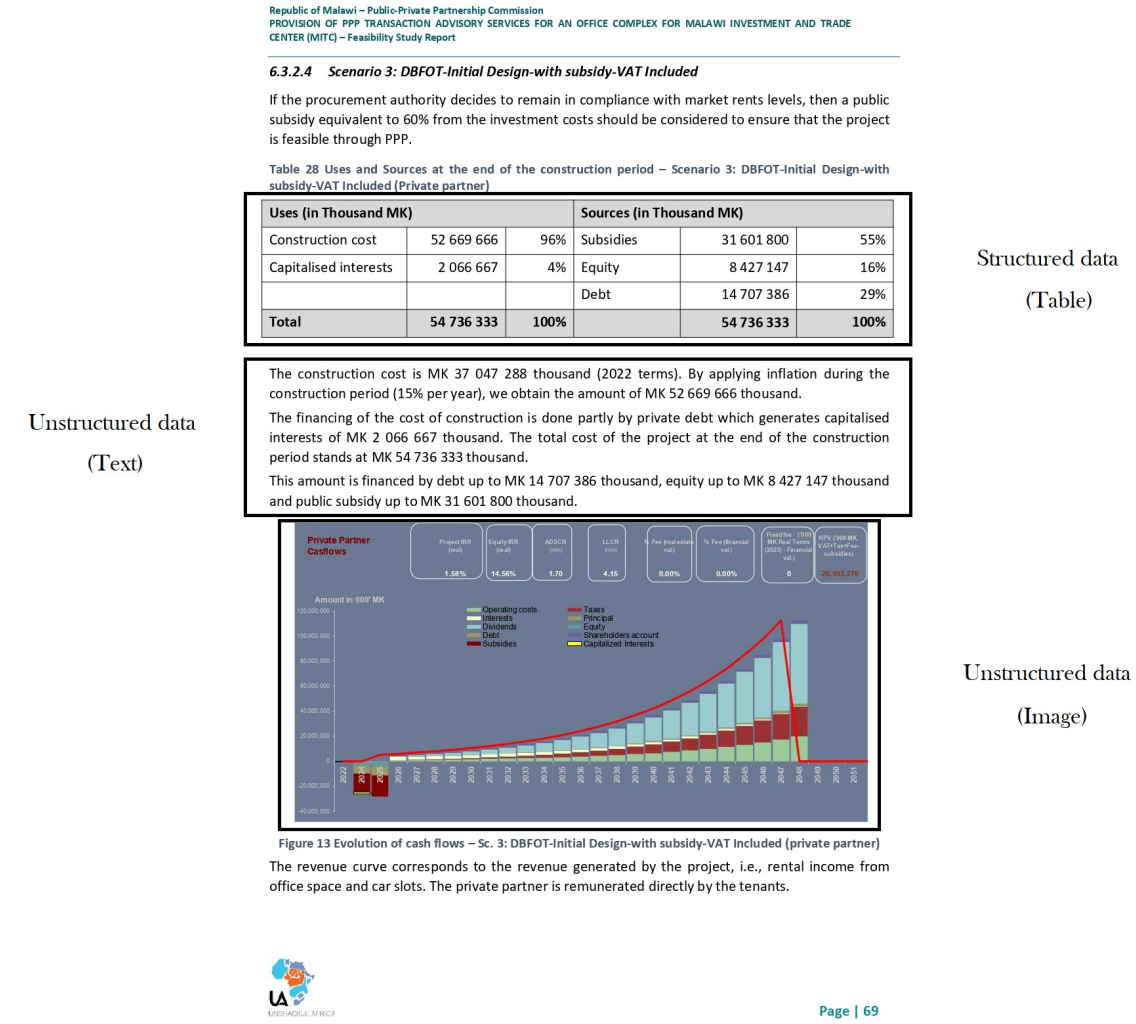

Figure 4.1: Example of a dataset page

Text: The textual data The textual data includes all the information about projects, contracts, reports on progress, and messages to people involved in the project. The text is full of details but it's hard for computers to understand because it doesn't have a clear structure. Understanding and interpreting detailed documents to find useful information can be difficult because of the complex language and jargon used in PPP projects.

Tables: Embedded within documents, tables give important, organized information about finances, schedules, and performance indicators. The issue comes from the different ways the tables are set up and used in the text. Automated tools need to figure out and take out important information without losing the meaning given by the surrounding text, which can be difficult because of the different layouts and designs used in documents.

Pictures: Pictures, like project drawings, graphs, and photos of the site, give important information about how the project is going and what the results are. Their study is limited because they require special processing methods. Taking information from pictures, especially when it has words or complicated visuals, needs high-tech OCR and image recognition tools, which may have trouble with picture quality, direction, and combining visual data with text and tables.

The variety and semi-structered data of our dataset show how complicated it is to analyze data from PPP projects. The difficulties include:

Data Integration: Combining information from text, tables, and pictures to better understand project progress and results.Context Preservation: Keeping the connections between different types of information, like the story behind a table or the details about an image, is very important for getting the analysis right.Format Variability: There are many different types of documents that need flexible tools to deal with them. Technical Language: The terminology and common terms in PPP papers need really smart AI tools to understand them correctly.

Dealing with these challenges requires advanced tools like chat bots that can handle complex PPP project data in a nuanced way. This helps in processing and analyzing data effectively.

4.4 Types of Chat-bots

In the context of managing and analyzing PPP data, it is important to know the different types of chatbots that you can use. These can be put into two groups: rule based chatbots and AI driven chatbots. Each one has its own specific uses and levels of difficulty and flexibility.

- **Rule-Based Chatbots:** These chatbots work based on a specific set of rules. They are designed for simple and predictable conversations. Rule-based chatbots excel in scenarios where queries fall within a well-defined range, making them efficient for FAQ-style interactions. However, using to analyze PPP data is difficult because the data is complex and can vary a lot. They are not able to handle unstructured data or give insights beyond their set rules.

- **NLP-Driven Chatbots:** AI-driven chatbots, powered by machine learning and natural language processing NLP , can understand and interpret the nuances of human language. Unlike their rule based counterparts, they learn from interactions to improve their understanding over time, making them capable of handling unstructured data. In the world of PPP projects, AI-powered chatbots can look at a lot of

insights, find patterns, and get useful information from previous project documents. Their skill in handling and understanding lots of difficult data makes them great at giving personalized advice and predicting future outcomes for projects.

- **Hybrid Chatbots:** Using elements from two different methods, hybrid chatbots can answer simple questions using set rules and switch to more advanced answers using artificial intelligence for harder questions. This method ensures that we handle basic questions efficiently and consistently, while still having the ability to dig deeper into data and generate insights for PPP projects.

- **Retrieval-Augmented Generation (RAG) Chatbots:** RAG chatbots are a type of AI chatbots that are really good at finding information from a big database before coming up with answers. Chatbots use information from previous projects to give helpful advice and predictions about future projects. Their ability to access historical data archives makes them extremely useful for understanding complicated PPP data environments.

In the context of tackling the challenges presented by PPP data analysis, AI-driven and RAG chatbots hold the most promise. Their ability to process and learn from unstructured data, combined with the capability to retrieve and utilize specific information from large datasets, positions them as powerful tools for extracting insights from historical PPP project data. This, in turn, can significantly enhance decision-making and strategic planning for future projects.

4.5 Why RAG

Retrival Augmented Generation RAG mixes the Advanced text-generation skills of GPT and other large language models with information searching features to give correct and contextually relevant information. This new method helps language models better understand and respond to user questions by using the most up-to-date information available. As RAG grows, its many uses are going to change how well AI works and how useful it is.

In general, General-purpose language models are trained on a lot of data from everywhere. But doesn't mean it knows the answer to every question. Traditional LLMs lack important things like current or important information, specific context, fact checking, etc. That's why they re-called general purpose and need the help of other widely used techniques to make LLMs more versatile. So, we fix these problems by adding ways for language models to find information, so they can give better and more specific answers. This helps prevent them from making things up and makes them more trustworthy.

4.6 Our RAG Components and Pipeline

In this part, we will look at the detailed parts and steps that make up our Retrieval Augmented Generation RAG pipeline. We made this pipeline after lots of testing and evaluation to get the best results. This pipeline is crucial for our chatbot to move through and

communicate with the complex data linked to Public Private Partnership PPP projects. It is a smooth combination of different parts that work together to process PPP data sets, which contain text and tables within Word documents. **Figure 4.2** shows the pipeline.

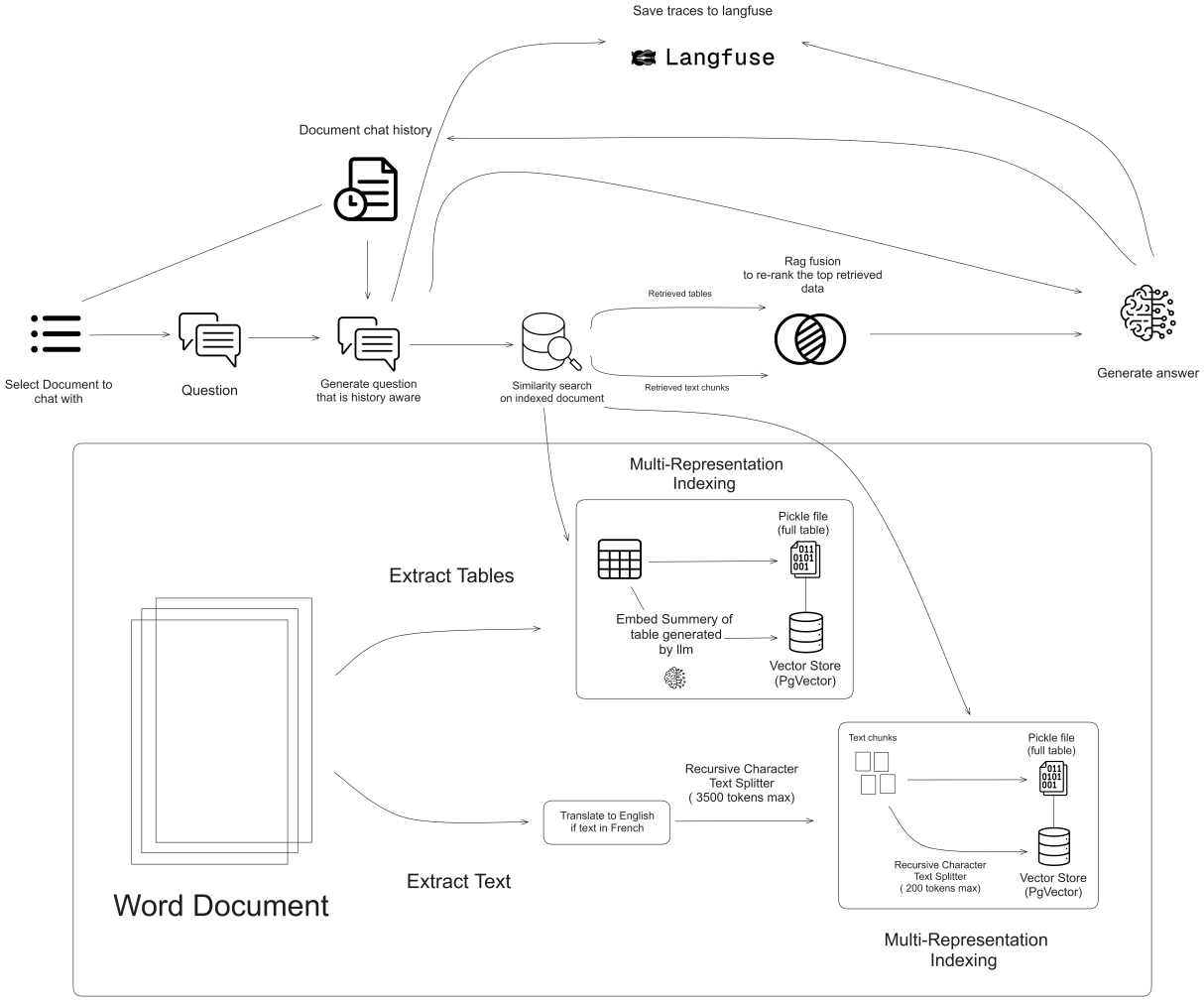

Figure 4.2: Our RAG pipline

Based on this pipline, in this chapter we will talk about:

- **Data Extraction and Preparation:** This step involves identifying and extracting out different types of information from Word documents. This includes text and tables. The process makes sure that all important information is gathered and ready for the next steps.

- **Data Embedding**: After extraction, the data goes through a process called embedding. This changes the information into numbers that AI can easily understand and work with. This embedding is very important for capturing the meaning of the data in a way that machines can understand and use it effectively.

- **History-Aware Conversation Management**: At this point, the system uses past interactions to keep track of the conversation flow. This makes sure that each

question is based on the history of the conversation, giving a smooth and relevant experience for the user.

- **Integrated Retrieval, Indexing, and Re-Ranking**: This system begins by retrieving data from out vector store based on the users question. The retrieval process is integrated with a creative technique known as RAG Fusion. During this phase, the retrieved data is re-ranked by importance, which enhances the quality and relevance of the information used for generating answers. This integrated approach ensures efficient data handling and improved response generation for the user.

- **Answer Generation**: The last stage of the process where the chatbot puts together the re-ordered information to create complete and relevant answers. The process uses language models and data to give accurate answers to user questions.

This advanced pipeline helps to change semi structured PPP project information into useful insights using a conversational AI interface. Each step is made to deal with the difficulty of the data and the details of how users interact, making sure that the chatbot is not just quick to respond but also very informative.

4.7 Data Extraction and Preparation

The text extraction stage is important in the RAG pipeline. Here, the AI system works with the Word document to find and separate all the text. This process is intricate and has many steps to make sure the text data is accurate and ready to be used.

The extraction process is divided into two distinct sub-tasks, one devoted to text extraction and processing and the other focused on tables extraction and processing This section allows essential control of each type of data, making the extraction process more accurate and efficient. By carefully analyzing textual content and tabular data, the system can gather broader perspectives and enhance its knowledge base, ultimately increasing its ability to provide contextual responses.

4.7.1 Text extraction and processing

The extraction process is divided into two parts, one for extracting and processing text, and the other for tables. This helps make sure the data is extracted accurately and quickly. By looking closely at written words and table data, the system can learn more and improve its knowledge. This helps it give better answers in different situations.

- **Full Text Extraction**: This process begins with a thorough scan of the entire Word document. Sophisticated algorithms are used to go through the document's complicated structure and capture all parts that have text. This means looking closely at all parts of the document, like the main text, titles, footnotes, and extra information on the side. Every paragraph, heading, and written note is carefully taken out, making sure no important information is missed.

This step is crucial because it sets the foundation for the kind of information the chatbot will have available. It is done carefully to capture the detailed story in PPP documents, like project goals, timelines, stakeholder duties, and legal rules.

- **Content Cleaning**: After getting the extracted data, an important cleaning process starts. It involves going through the text to find and delete parts that don't give important information, like the Table of Contents, index, bibliographies, and appendices. Automated scripts are designed to ignore certain parts and focus on the important details of the project.

  The cleaning also includes getting rid of any repeated headers and footers that show up on every page of the document. Moreover, acknowledgment sections and references are not necessary for the chatbot's data analysis and have been left out of the extracted content.

- **Handling Multilingual Text**: A special problem occurs when the documents are in languages other than English. For example, if the PPP documentation is in French, the system uses Argos Translate, a free, offline translation tool, to change the text into English. "We chose Argos Translate for privacy reasons and to be able to work without needing internet services."

  The choice to use machine translation instead of multilingual embeddings is based on a study by the GDELT Project in their article "Embedding Models: Multilingual Embedding Versus Machine Translation + English Embedding". The study in the article shows that translating non English text into English before embedding works better than using separate embedding models for each language. **The figure 4.3**, extracted from the article, outlines a comparison between Multilingual Embeddings and Machine Translation then English Embeddings over different dialects , counting Arabic, Chinese, Thai, Russian, German, Spanish, Korean, and Turkish with the same embedding model. As portrayed , the results clearly favor Machine Translation + English Embedding for accomplishing more precise clustering and regrouping of diverse sentences.

Figure 4.3: comparison between Multilingual Embeddings and Machine Translation then English Embeddings over different dialects

By translating all words to English and then using English language embeddings, the system makes sure that the language model can use its training effectively and give the best responses in the RAG pipeline.

4.7.2 Table extraction

In the RAG pipeline, extracting tables involves carefully finding and getting data from tables, as well as keeping the titles or captions that give information about the tables.

- **Identifying and Extracting Tables with Titles**: The first step is to scan the Word document for tables, which are important for analyzing PPP projects. Beside every table, the system also takes out the title or caption that goes with it, often found right above the table in the document. The title is important because it gives a summary or focus of the content in the table, serving as a key to understanding the data. By taking out the table and its title together, the system keeps all the information intact. This helps users to know the meaning and importance of the data.The process begins by scanning the Word document for tables, which are often pivotal in conveying structured, quantitative data essential for PPP project analysis. Alongside each table, the system also extracts the associated title or caption, which is frequently located directly above the table in the document. This title is key to understanding the table's content as it often provides a summary or highlights the focus of the tabular data. By extracting the table and its title in tandem, the system retains the full context, allowing users to understand the purpose and implications of the data within.

- **Transformation into HTML Format**: After a table and its title are extricated, they experience a change into an HTML (Hypertext Markup Language) format . HTML is chosen since of its various leveled structure, which adjusts well with the inalienable structure of tables. It permits for the representation of complex information in a settled , discernable arrange that's both human readable and machine

parseable.

This change is guided by standards laid out in research on the representation of tabular data for large language models. For example, the approach recommended within the research paper "Table Meets LLM: Can Large Language Models Understand Structured Table Data? A Benchmark and Empirical Study" [7] serves as a guide for this process. The paper explains ways to show tables better so that the language model can understand and work with them more easily, **The Figure 4.4** from this paper presents the final findings of different types of transformations and their result across different llm table understanding benchmarks.

Format	TabFact	HybridQA	SQA	Feverous	ToTTo
	Acc	Acc	Acc	Acc	BLEU-4
NL + Sep	70.26%	45.02%	70.41%	75.15%	**12.70%**
Markdown	68.40%	45.88%	66.59%	71.88%	8.57%
JSON	68.04%	42.40%	70.39%	73.84%	8.82%
XML	70.00%	47.20%	70.74%	73.14%	8.82%
HTML	**71.33%**	**47.29%**	**71.31%**	**75.20%**	12.30%
GPT-4 w/ HTML	78.40%	56.68%	75.35%	83.21%	20.12%

Figure 4.4: Results of different transformations on LLM table understanding benchmarks

When we transform tables to HTML format, each part of the table (cell, row, and column) is put inside special HTML tags. This makes a tree like structure that shows the table's layout accurately. The HTML format clearly separates data points, which helps the language model access and understand the table s contents accurately.

- **Preparation for LLM Interpretation**: The table is represented in HTML format to help the language model work with tabular data more effectively. This helps the LLM to better use the data from the table during conversations. It gets really good at using and making connections between the tabular data and the text or questions it comes across.

The RAG pipeline improves the chatbot's ability to use structured data by changing tables and titles into a format that works well with language models. This step makes sure that the chatbot can understand numbers and patterns as well as written words. This helps the chatbot give better and more detailed responses when analyzing PPP projects.

4.8 Data Embedding

Word embeddings are key concept in NLP, a fieldin machine learning. Word embeddings convert text data into numbers that machine learning algorithms can understand.

can also help us understand the meaning of words in relation to other words. For example, **Figure 4.5** shows a example of embeddings in three dimensions.

Before embedding, the document must be divided into smaller sections or chunks, so it can be analyzed and processed more easily. Chunking is play a pivotal role in helping with remembering and creating things quickly. These pieces are then dealt with separately, making it easier to find and create specific information.

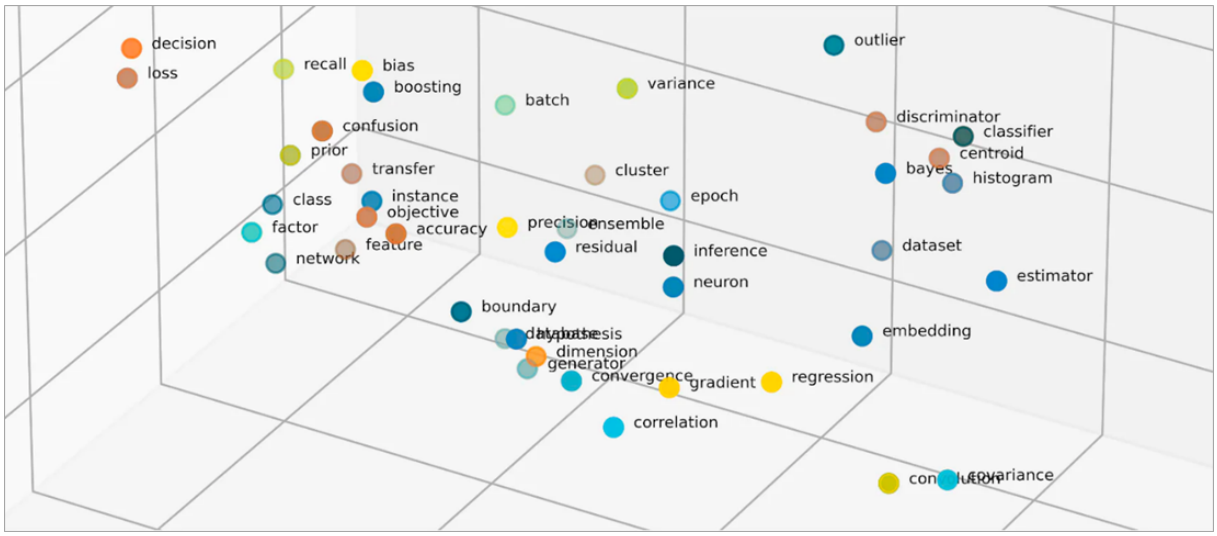

Figure 4.5: Three-dimension embedding sample

4.8.1 Choice of embedding model

Choosing the right embedding model is very important, especially when working with English documents. The model must be good at understanding the language and specialized terms used in legal and financial documents.

The BGE M3 model, mentioned in the article "OpenAI vs. Open Source Multilingual Embedding Models "[8], makes a strong argument for why it should be chosen. This model is very good at understanding many different meanings in English because it has been trained on lots of different types of data from lots of different areas. **The figure 4.6** shows the comparison between different models and their performance on Mean Reciprocal Rank (MRR) which is an evaluation benchmark for embedding models.

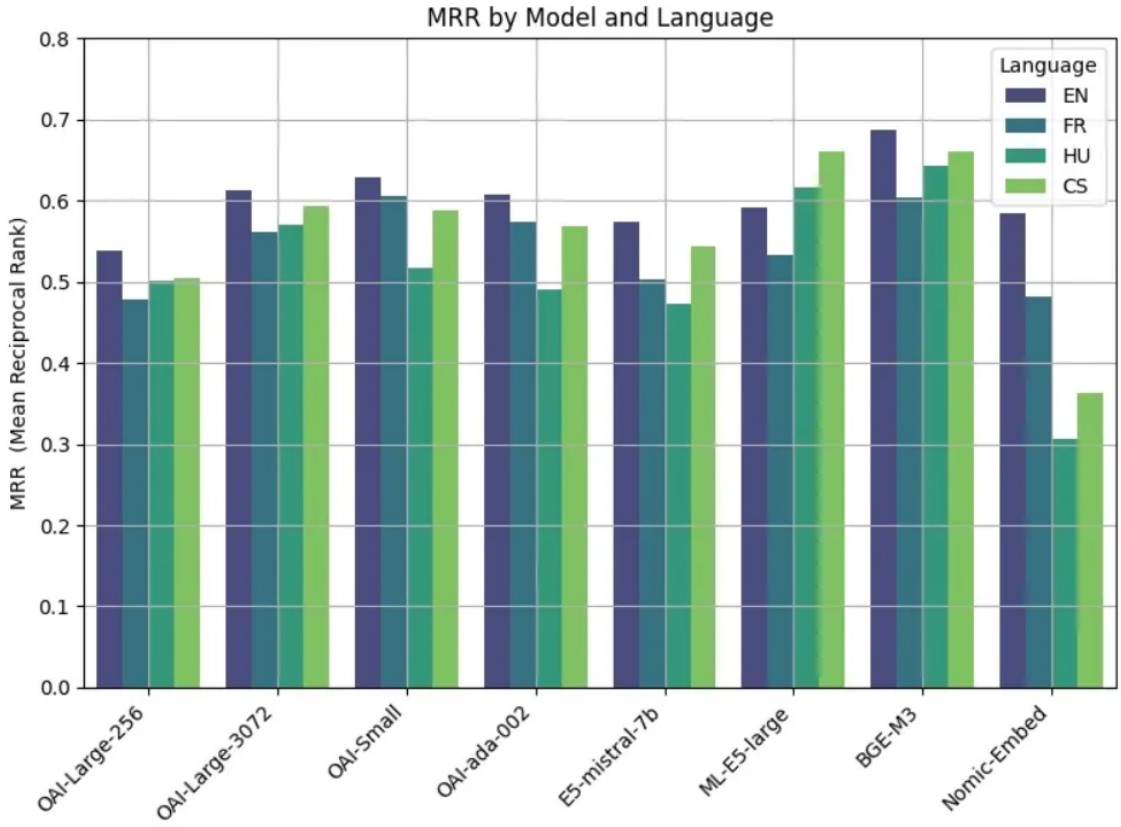

Figure 4.6: performance of the latest embedding models on Mean Reciprocal Rank (MRR) benchmark

Choosing BGE-M3 for English Language Embeddings: The BGE-M3 model is special because it is part of a group of models that are made to really understand language well, , stands out for several reasons:

1. **Model Specifications:** The BGE-M3 model, identified as BAAI/bge-m3, has a dimension of 1024 and can process sequences up to 8192 tokens in length. It is a multilingual model that benefits from a unified fine-tuning approach, incorporating techniques such as dense, sparse, and colbert tuning. These features are derived from its unsupervised learning capabilities, enhancing its versatility and adaptability across various text analyses.

2. **Contextual Understanding:** It has been trained to understand context well, which is important for reading PPP documents that often contains complicated language structures.

3. **Semantic and Syntactic Awareness:** The BGE-M3 model captures how words are related, which is important when dealing with technical language where accuracy is very important.

4. **Robust Training:** The training process for BGE-M3 has helped the model learn

different language patterns, so it can handle new words and ways of speaking easily.

5. **Efficiency in Processing:** As PPP documents can be very long and complicated, the ability of BGE M3 to process a lot of text quickly is a big benefit, making sure the system stays fast.

In summary, the strategic selection of BGE-M3 for English language embeddings helps us analyze PPP project documents better. It makes sure we can use a platform that gives us lots of specific information, context, and accurate insights necessary for informed decision-making.

4.8.2 Chunking and embedding for different type of data

Before embedding, documents are chunked (divided) into smaller segments. This step before the main process is important for controlling the system's workload and helps to find information in a detailed way. Breaking down the text into smaller parts helps the system find and organize information better, making it easier to search for specific content. This approach helps prevent losing important information when dealing with long texts. Each piece of information is carefully looked at.

In our RAG pipeline, Multi Representation Indexing is a very important step where we map different types of data, like text and tables, to their own vector representations. This process makes it easy to find information and ensures you can remember everything.

4.8.3 Tables Chunking and Embedding

The multi-representation indexing approach in our system is designed to handle the complicated table data in PPP documents. Here, the term multi-representation shows that the pipeline can store different types of data, such as vector summaries of tables and the tables themselves.

- **Process of Indexing Table Embeddings:** When we put the tables into the system, we create vector summaries of the important information in the tables. These summaries are stored in a database called PgVector which can handle vector data. This process involves making a map between a high-dimensional vector space and the original table. Each vector is a point in this space, representing the meaning and information in a table summary. By indexing these vectors, we make it easier to find things quickly when looking for specific information through semantic queries.
- here we choose the llmm (comp table)
- **Retrieval of Full Table Data:** The strength of multi-representation indexing is that it able to link these vectors back to all the data in the table. When a user needs specific information, the system can find the right data and get the complete table. This means that users will get a short answer to their questions and also be able to look at all the detailed information in the table to get a better understanding.

4.8.4 Text Chunking and Embedding

The sophisticated architecture of our RAG pipeline can handle different types of text data from PPP documents using a method called multi-representation indexing. This means the pipeline can organize various kinds of data, especially focusing on the vector representations of small text pieces and the longer text segments they come from(parent chunck).

- **Chunking for Text Embedding:** For this process, we are using Recursive text splitting which involves breaking down a big piece of text into smaller chunks. This is really helpful when working with long documents, because it helps the system to manage and process data while still keeping important information in context.

  The process starts by breaking the document into big chunks, each containing up to 3000 tokens. This initial size of each chunk is set in order to make sure that each part has enough information to accurately show the details of the text.

  After this first pass, these large chunks may still be too broad for the fine-grained analysis and retrieval purposes of our system. Therefore, the large chunks undergo a second round of recursive text splitting, where they are further divided into smaller chunks, each consisting of approximately 200 tokens. This size is optimized for the embedding process, where each small chunk is then transformed into a vector representation that captures the semantic richness of the text within that segment.

- **Multi-Representation Indexing:** In our pipeline, both levels of chunked data—large (3000-token) chunks and small (200-token) chunks—are important. The small chunks are embedded and the vectors produced from these embeddings are indexed. However, when it comes to retrieval, the system uses multi-representation indexing to ensure that it can return information from the larger chunks.

  The selection of recursive text splitting and multi representation indexing offers a highly productive approach to handling broad PPP documents. By breaking content into both large and small chunks and embedding these for point by point indexing , the framework adeptly balances comprehensive information analysis with the require for holding relevant information. This technique guarantees that the retrieval process is robust and exact.

4.9 History-Aware Conversation with Document

Within the domain of interactive AI frameworks , the capacity to review and construct upon past saved interactions is fundamental for keeping up a coherent and significant discourse . Typically where History Aware Conversation Management comes into play inside our Rag pipeline, especially within the context of engaging with PPP reports.

1. **Selective Document Interaction:** Inside our Rag pipeline, the selection of a particular document to interact with is essential. It engages users to lock in specifically with a specific PPP feasibility report from an established store associated to Jade's OneDrive collection and also save their chat history to our database. The user can

choose and search for a report, then they can start a unique conversational thread with that document. This interaction isn't generic but profoundly personalized, permitting the user to inquiry and speak with the content of the chosen report as in case it were a knowledgeable partner.

2. **Question Translation and Reformulation for Context Awareness:** After selecting the document, then the users can pose questions in various languages, the system adeptly translates these into English, using advanced models to ensure nuanced accuracy. This translation is then fused with the document's chat history, providing the large language model (LLM) with a rich contextual backdrop. The LLM may reformulate the inquiry for enhanced context specificity, drawing from previous dialogues and document details to craft questions that elicit more precise information. This consistent use of English across the board maintains the system's processing uniformity, ensuring each user receives context-aware, insightful responses from the conversational AI.

4.10 Integrated Retrieval and Re-Ranking

Within the Rag pipeline, the integrated retrieval and re ranking process plays a essential part in improving the system s response quality. This stage takes after the extraction and embedding stages where both tables and content chunks are indexed and made searchable.

4.10.1 Similarity Search on Indexed Documents

The Similarity Search stage could be a pivotal component of the Rag pipeline, where the system s capacity to accurately and productively coordinate user questions to the most significant information comes into play.

1. **Mechanism of Similarity Search:** Upon accepting a query , the system takes the user s input into a query vector utilizing the same embedding model applied to the text and table information during the indexing stage . This vector represents the semantic signature of the query.

2. **Finding the Best Match:** The multi representation indexed archives make a searchable space, our vector store (PgVector). Comprising of vectors for each chunk of content and table information. Each vector acts as a arrange in this high dimensional space, encoding the semantic essence of the corresponding text or table.

   The system conducts a search inside this vector space to distinguish the vectors that are closest to the query vector those with the highest degree of semantic similarity . This similitude is regularly computed utilizing metrics such as cosine similarity , which measures the cosine of the angle between two vectors, showing how closely aligned the implications of the query and the archive chunks are. For example, **Figure 4.7** shows a vector representation of kitten query and how it's close to other animals representation especially to cat and dog.

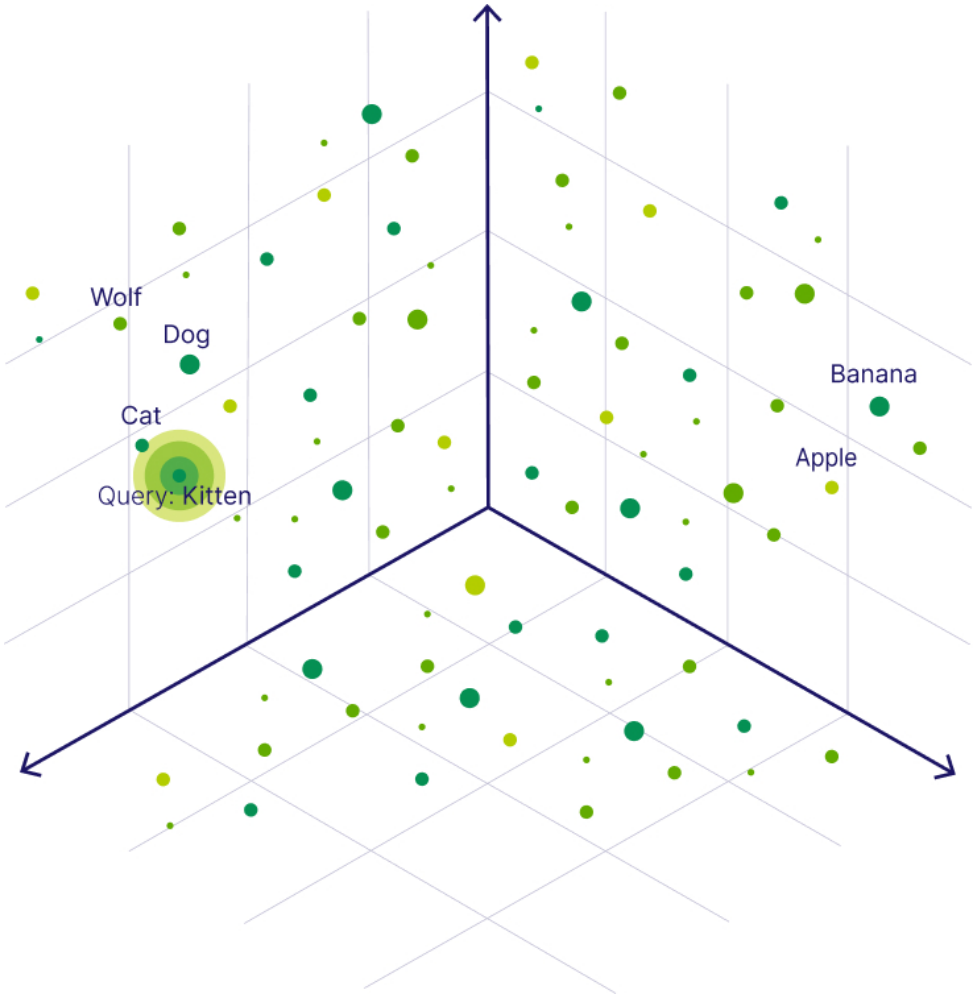

Figure 4.7: Vector representation of kitten query and how it's close to other animals representation in a 3 dimensional space

> The closest vectors are at that point mapped back to their original content chunks or table information . This retrieval is the primary step in giving a response to the user s query , sourcing the foremost semantically aligned pieces of data from the complete corpus of indexed documents .

The Similarity Search on Indexed Reports may be a confirmation to the power of vector space modeling in understanding and retrieving data. It empowers the system to filter through endless amounts of information with precision and return the foremost pertinent data, hence streamlining the user's experience in exploring complex PPP reports.

4.10.2 RAG Fusion for Contextual Alignment

RAG Fusion is an advanced technique in the RAG pipeline that harnesses the strengths of ensemble methods and reciprocal rank fusion for superior data retrieval.

At it's core RAG Fusion uses Reciprocal Rank Fusion (RRF) which is an integral component within the context of Rag fusion that essentially contributes to the efficiency and accuracy of the retrieval process within the pipeline. Its execution inside the Rag

system offers a method for combining the strengths of distinctive retrieval models to ensure the foremost important data is surfaced in response to a user's inquiry.

- **Fundamentals of Reciprocal Rank Fusion (RRF):** RRF is based on the principle that the significance of a document can be inversely relative to its positioning over diverse retrieval models. The method was detailed in a research paper by Cormack, Clarke, and Buettcher,titled "Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods"[9]. It s an successful data fusion strategy that totals different positioned lists into a single, more exact ranking.

- **How RRF Works:** In RRF, each document's rank from the individual result sets is converted into a score using the reciprocal of its rank. This means that if a document is ranked first in any retrieval model's list, it's assigned a score of 1; if it's ranked second, it gets a score of 1/2, and so on. These scores are then summed across all retrieval models for each document. The documents are then re-ranked based on these composite scores, with higher scores indicating higher relevance. This is the RRF equation

$$\text{RRFscore}(d \in D) = \sum \left[\frac{1}{k + r(d)} \right]$$

  where:

  - d represents a document within the set D of documents.
  - k is a constant that helps to balance between high and low ranking documents.
  - $r(d)$ is the rank or position of the document d.

- **Contribution to our RAG Pipeline:** The application of RRF in the RAG pipeline enhances the retrieval process by allowing the system to effectively combine and leverage the insights from our two different retrievals, the text retrieval and table retrieval . This is particularly useful in the RAG framework, where diverse data formats and complex query contexts can lead to varying retrieval performances across different models and with RRF we can have a unified set of score for each chunk returned whether it is a text chunk or a table.

The RRF method enhances the RAG fusion process by adeptly amalgamating the strengths of varied retrievals, ensuring the system's effectiveness over diverse datasets. Its application is pivotal in re-ranking the information chunks retrieved, which is essential for furnishing the most contextually relevant and reliable answers to complex inquiries. By reordering these data chunks, RRF helps to align the output more closely with the user's original intent.

The effectiveness of RRF in reordering content for LLMs can be further understood through the lens of the research paper "Lost in the Middle: How Language Models Use Long Contexts"[10]. This study underscores the challenges and solutions in how LLMs handle extended contexts. LLMs are often designed to prioritize information at the top and the end of a user query as being more relevant, which might not always align with

the user's needs. We can mitigate this by re-ranking the data chunks, placing those with the highest relevance at the start and end of our query. This tailored reordering can lead to more coherent and contextually rich responses from the LLM. **Figure 4.8** from this study shows how Changing the location of relevant information within the language model's input context gives different results.

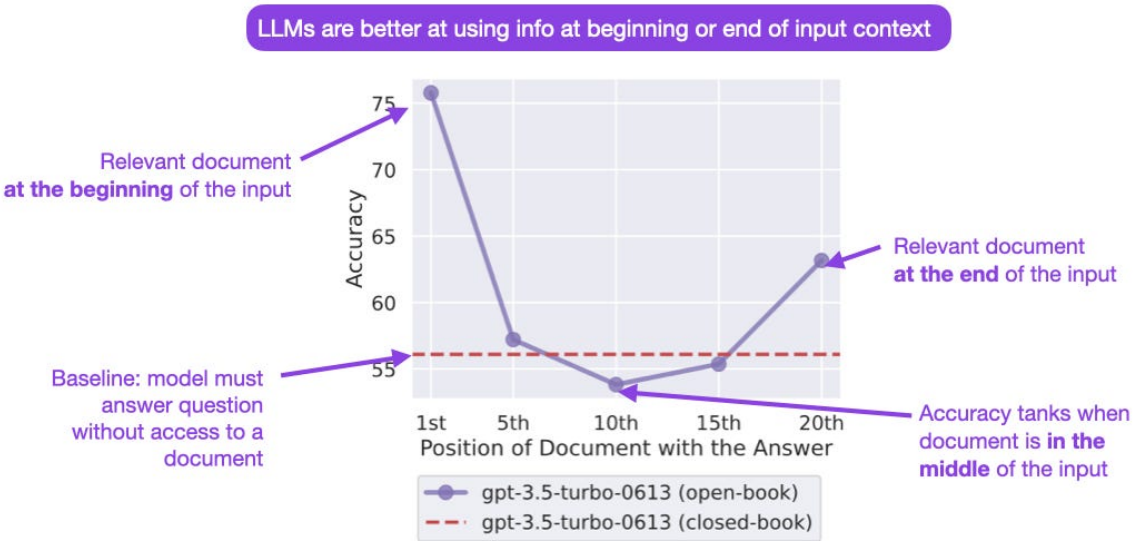

Figure 4.8: Result of changing the position of the passage that answers an input question

In essence, RRF acts as a strategic intermediary in the RAG fusion process, optimizing the arrangement of data chunks fed into the LLM. This optimization facilitates the generation of better-informed answers, effectively harnessing the long-context capabilities of LLMs to provide users with precise and contextually nuanced responses.

4.11 Answer Generation

4.12 RAG system Evaluation

4.13 conclusion

Chapter 5

Feasibility Document Generation

5.1 Introduction

The Feasibility PPP Document Generation tool revolutionizes the reporting and analysis process in Public-Private Partnerships by automating the extraction of data from Excel files and generating comprehensive reports. This innovative tool leverages artificial intelligence to transform quantitative data into descriptive narratives, significantly enhancing the efficiency and depth of PPP scenario assessments. Designed to streamline the creation of reports and improve decision making, it represents a pivotal advancement in leveraging technology for PPP project management. This chapter outlines the development, functionalities, and impact of the tool on simplifying complex data analysis within the PPP framework.

5.2 Objectives

The AI-powered PPP document generation tool is designed with strategic objectives to enhance Public-Private Partnership project management and reporting. By automating data extraction and analysis from Excel files through artificial intelligence, it transforms complex quantitative information into accessible, descriptive narratives. This key feature streamlines the creation of detailed reports, improving their quality by offering deep insights into the data, thus aiding in clearer understanding and decision-making.

Efficiency is a central objective, with the tool reducing manual effort and time traditionally required in the compilation of reports. It frees stakeholders to focus on the strategic aspects of project management. Design also supports informed decision-making by providing comprehensive but straightforward reports that highlight essential data insights, guiding stakeholders toward well-informed decisions.

Scalability and adaptability are intrinsic to the tool, designed to meet the varied demands of different PPP projects and sectors. This flexibility ensures that it can be a valuable resource in diverse scenarios, enhancing its utility and application.

The anticipated outcomes include increased report accuracy, efficiency in resource allocation, improved stakeholder engagement through clearer communication, and the provision of data-driven insights for better risk management and opportunity identification. Adopting this advanced technology also enhances an organization's competitive

edge, showcasing a commitment to innovation and operational efficiency in PPP projects.

In summary, this tool represents a significant step forward in leveraging technology to streamline PPP project evaluations, promising a combination of improved efficiency, accuracy, and strategic insight.

5.3 Document Generation Architecture

The system architecture of the AI-powered PPP Document Generation tool encapsulates a highly coordinated process that transitions data from initial input to final output. This process is delineated in the following components as shows the **Figure 5.3**:

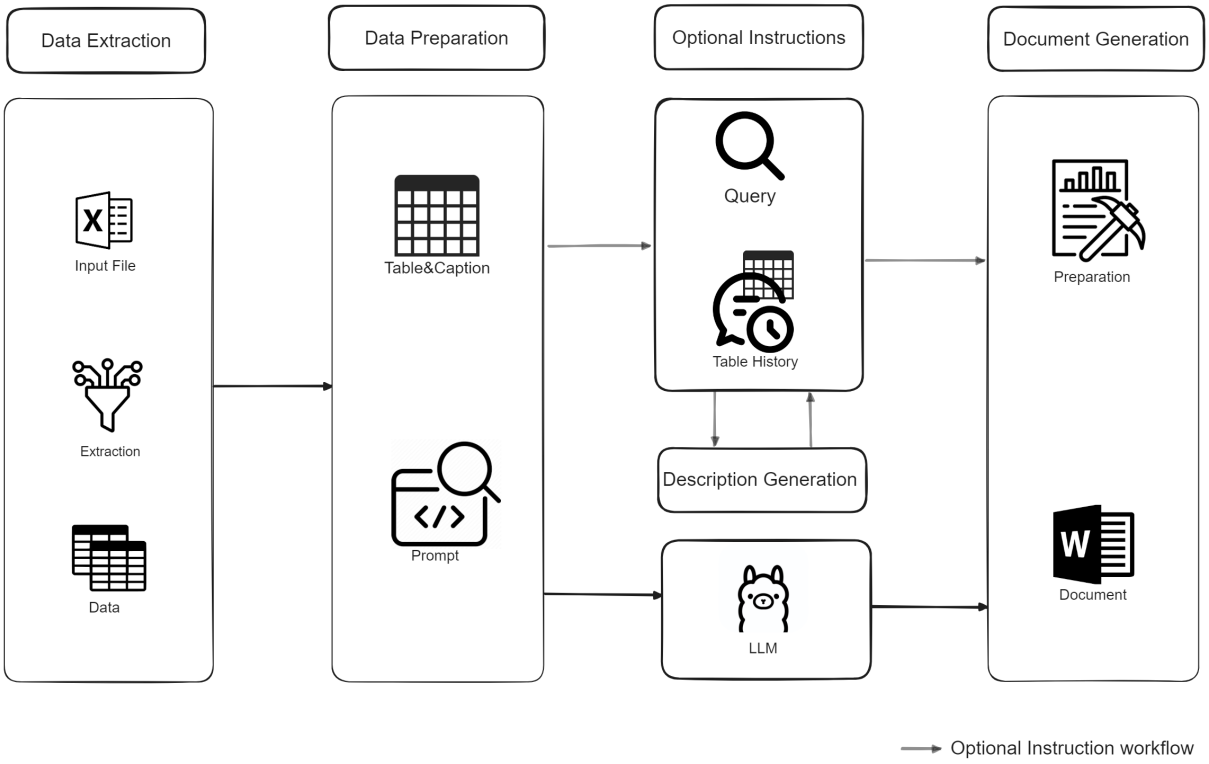

Figure 5.1: Document Generation Architecture

The workflow starts with the Data Extraction phase , a significant beginning point where raw information is gathered from an Excel file . This stage is foundational, focusing on identifying and gathering titles, tables, and the captions that illustrate the context of each table. Extraction tools are deployed to automate this process , guaranteeing that the data is pulled accurately and efficiently . The result are a compilation of natural data , which is an accumulation of figures, content , and names ready for the another stage of refinement.

Following extraction, the Data Preparation phase commences. Here, the raw data experiences a transformation into a structured format conducive to examination . This includes sorting, cleaning, and organizing the data into coherent tables accompanied by

descriptive captions. In pair , a well-crafted prompt functions as a directive for the consequent analytical process . The prompt encapsulates specific instructions and objectives that guide the dealing with and interpretation of the information , setting the stage for the application of advanced analytical procedures .

Consolidating a layer of adaptability into the workflow are the Optional Instructions . This intermediary step isn't always actuated but serves as significant part when utilized. It permits for the integration of custom queries and extra orders that relate to the historical perspectives of the information or particular analytical requirements . This flexible component guarantees that the following examination isn't as it were strong but too custom fitted to the interesting necessities of the project or the questions at hand.

With the information prepared and prompts in place , the Description Generation phase leverages the capabilities of a Large Language Model (LLM). The AI ingests baseline data from the prepared information , if provided , to synthesize a description. This story may be a nitty gritty , AI generated composition that dives into the insights inferred from the information. It contextualizes numbers and patterns inside the tables, weaving them into a description that's both informative and available to the intended audience.

The perfection of the workflow is the Document Generation phase . The wealthy , descriptive content delivered by the LLM is fastidiously gathered into a formal report , typically a Word file . This document encapsulates the analytical journey from raw data to Description, insights and presents it in an format that's cleaned and prepared for spread . The document not only describes the findings but is also structured in a way that coordinating visual components , such as merging cells and styling, which improve comprehension and give a visual outline of the literary investigation .

Additionnaly, these stages represent a spic-and-span and modern prepare , changing tables from an Excel spreadsheet into a narratively wealthy and visually supported document that is both comprehensive and prepared for professional PPP report.

5.4 Data Extraction

5.4.1 Input File

The foundation of our data-driven approach starts with the Input File , typically an Excel file **Figure 5.2**, containing different spreadsheet, each one includes sheet title, tables, and corresponding captions. These records represent the raw material from which valuable insights are to be extracted and afterward refined into a comprehensive report.

Figure 5.2: Input File

spreadsheet sample

The spreadsheet, as demonstrated on **Figure 5.3** is a collection of tables, each with a unique title wich is the title of the sheet and a caption that gives insights for the data. Besides, The expressive caption of ceratin table clarifies the content and significance of the table itself. However, The tables contain quantitative data that's fundamental for creating the PPP reports. The dataset is structured in a tabular format , with rows and columns representing the data points and categories, respectively . This structured dataset serves as the essential input for the AI powered report generation tool , empowering the extraction of valuable insights and the creation of detailed reports.

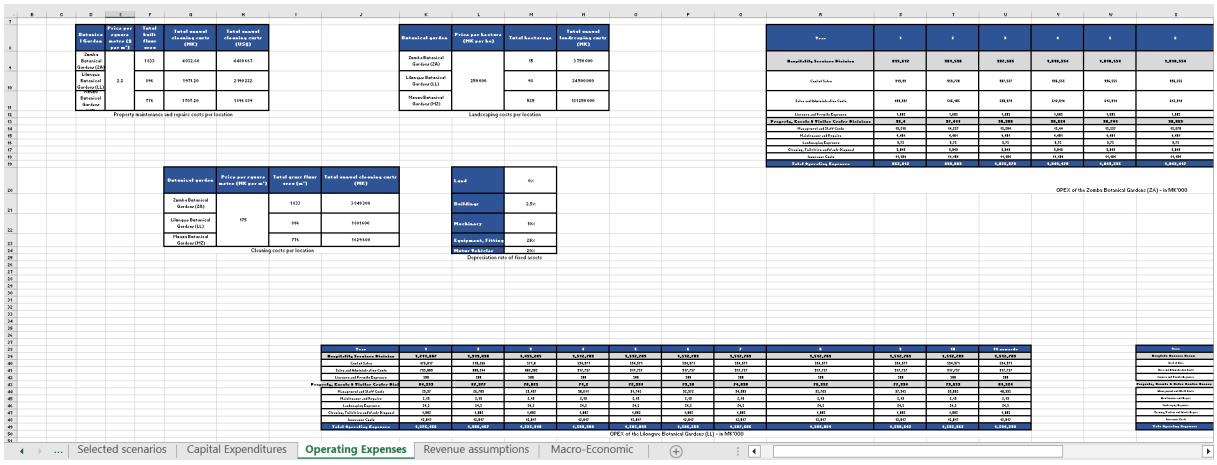

Figure 5.3: Sample of dataset

Obviously, When working with spreadsheets, it's easy to get lost in all the numbers and information. But there are some simple things we can do to make them easier to follow:

- **Don't Make Cells Too Complicated:** Sometimes, we might think it looks better to combine cells together, but it actually makes things harder to read. Instead of merging cells, just repeat the data in each cell. This way, everything remains clear and organized.
- **Fill in Empty cells:** If there are empty spaces in your spreadsheet, it can be confusing. To settle this, just put a dash '-' in those spaces. This indicates that there's no data there, so no one gets confused.

- **Make Titles and Captions Unique:** It's imperative to provide each part of your spreadsheet a different title. If you use the same title or caption more than once, it can be confusing. So, always make sure each title or caption is different from the others.
- **Keep Tables Separate:** Sometimes spreadsheets can get really enormous and untidy, with lots of different tables mixed together. This may make it difficult to understand what's going on. Attempt to keep each table on its own sheet, so it's easier to see what belongs together.

By following these simple tips, we could make our spreadsheets much easier for the extraction process.

5.4.2 Data Extraction Process

In this stage, the essential tool utilized for extracting tables from Excel files is eparse library, a robust library designed particularly for recognizing and parsing organized data inside spreadsheets. This tool proficiently handles the extraction of numerous tables from a single Excel file , recognizing particular table structures through their layout. Upon loading an Excel file into the system , eparse analyzes the spreadsheet to identify delineations between distinctive data squares , recognizing empty cells, corners, and the boundaries of each table. It can recognize tables that are closely positioned however separate , and indeed perceive subtables inside bigger tables based on varieties in organizing or column arrangements . This capability permits us to efficiently extricate each table as a discrete dataset.

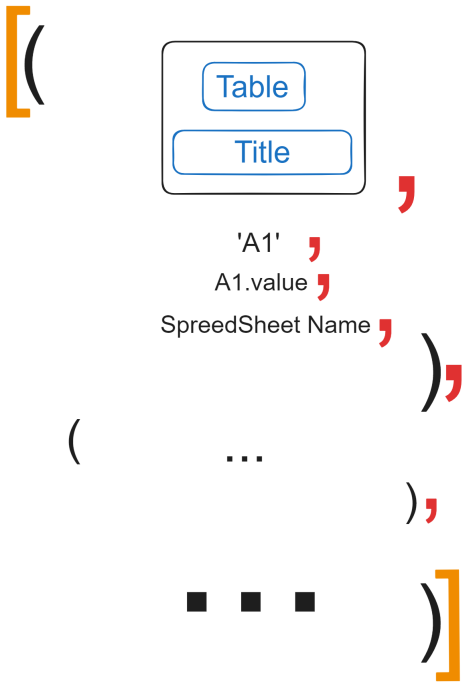

Figure 5.4: Extracted Data Structure

the accompanying **Figure 5.4** serves as a visual abstraction of the eparse tool's functionality in extracting table data from an Excel spreadsheet. Central to the figure may be

a representation of a spreadsheet table, indicated by the names Table and Title, demonstrating the distinguished table and its corresponding title. Flanking the table are curly braces, which, in programming contexts , typically encapsulate a related set of information or a code block . This suggests the encapsulation of the extracted table data for processing. the table representation, the cell A1 of the table and its value imply the extraction tool's capacity to reference and retrieve the value from a specific cell within the spreadsheet. This level of detail highlights the accuracy with which eparse can explore and translate spreadsheet contents. SpreadSheet Name underscores the tool's capacity to not only extract information from the sheets but also to distinguish and utilize the spreadsheet's title , possibly for organizational or referencing purposes.

the ellipses at the base of the figure suggest the continuation of the process , suggesting that what is represented is a part of a larger sequence of steps of the extracted data. The overarching curly braces enveloping the whole outline emphasize the cohesive and organized extraction process performed by eparse, from distinguishing person cells to recognizing whole spreadsheets by name.

This graphic metaphorically encapsulates the capabilities of the eparse tool, effectively summarizing the extraction process's scope from a high-level perspective.

5.5 Data Preparation

When we get our data ready, we make sure to put it in order by the names of the sheets from the Excel file as shown in **Figure 5.5**. This helps us a lot because it's like putting our music playlists into different genres—it's way easier to find what you need when everything's sorted out.

Imagine each sheet in Excel is like a different folder on your computer. We take the tables from each sheet and label them so we know exactly where they came from. It's like having a well-organized file cabinet, so when you need a specific document, you know right where to go.

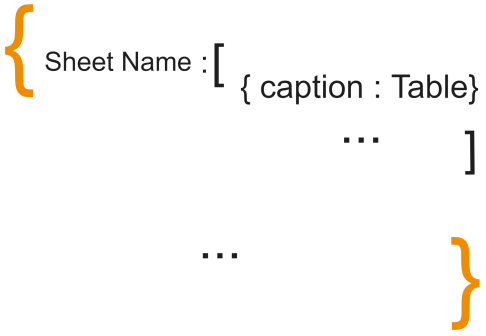

Figure 5.5: Organized Data

5.6 Optionnal Instructions

In this part of the report, we talk about two cool features of our system that let users have more control over what they're working with.

5.6.1 Interacting with the Order of Tables

First, we look at how users can arrange the order of the tables. This is pretty handy because sometimes you want the tables to show up in a particular way, like having the most important one first.

Imagine moving songs around in a playlist, we might have something similar as shows where you can just drag the tables into the order you like.

5.6.2 Viewing Tables and Adding Instructions to Prompts

Second, we talk about how users can see the tables right before they turn into text and add any last-minute details they think are important.

Table Preview

In our system, the capability to see tables,as shows the table, is a basic feature that enhances the user's understanding of the information being analyzed. This view only mode permits clients to see the tables as they are, without the capacity to associated specifically with the information through sorting or filtering . By showing the information in a clear, inactive format , clients can look at the structure and substance of the tables effectively . This straightforward visualization helps users assess the data more precisely and helps in planning the information for further examination or reporting . The center on a non-interactive display ensures clarity and minimizes the potential for perplexity , making it simpler for clients to interpret the information as intended.

Description Generation

Now we're getting to the really exciting part the Description Generation. This is where we take everything we've prepped and ask the language model to start making sense of it.We have our prompt ready, which is like a little message we send to the language model. This message isn't just a "hello" though; it's packed with useful stuff:

- **Commands:** We include clear commands in the prompt that tell the AI exactly what to do with the table, like "Explain the trends in this sales data" or "Summarize the key findings."
- **Archive of Prior Descriptions:** If we've done this before for similar tables, we let the AI know what it said last time. It's like reminding a friend of a story they told so they don't repeat themselves.
- **Title of the Table:** Knowing the title helps the AI get the context, it's the difference between just seeing random numbers and knowing these numbers are all about, say, "Monthly Ice Cream Sales."

- **Table:** Of course, we include the table data because the AI needs to see the numbers or info it's going to talk about.
- **Custom Request Details:** And lastly, if the client threw in any extra details, like wanting to focus on a certain column or needing the description to be super simple, we put that in the prompt too.

By letting clients interact with the order of tables and see them up close , furthermore include extra instructions , we make beyond any doubt that the system does what they need . It's all around giving individuals the control to get their tables just right before they turn them into description.

5.7 Document Generation

The document generation process in our system is designed to effectively convert extracted tables from Excel into well formatted Word documents .docx , ensuring that the data presentation is both professional and accessible . This prepare is crucial for delivering significant insights in a clear and organized way . Here's how we manage each viewpoint

5.7.1 Exporting Tables into a Word File

The process of exporting tables into a Word file includes a few steps to guarantee the clarity and readability of the report:

- **Cell Cleaning:** As part of preparing the tables for send out, any cells containing the symbol '-', which regularly indicates empty cell or unessential information, are cleaned to seem empty. This guarantees clarity and anticipates misinterpretation of the information.
- **Merging Cells:** To preserve the integrity and readability of the tables, our system naturally combines adjacent cells that share the same value. This not only preserves the format frequently utilized in Excel to demonstrate related information but also improves the visual appeal of the table within the report format.
- **Styling and Fonts:** The aesthetic presentation of the tables is carefully managed by applying consistent styling and textual styles. This includes setting suitable textual style sizes, styles, and colors that adjust with the generally report design. This guarantees that the tables are not only readable but also visually integrated with the rest of the report.

5.7.2 Descriptions Generation

In our system,when creating word document, we categorize tables into two types when generating Word documents those that already have an existing description , and those that do not . For tables missing a past description , our system leverages AI to generate informative summaries for each one. This prepare involves the creation of detailed prompts that coordinate the AI to completely analyze the displayed information . The prompts are carefully crafted to emphasize important data pointsdata points and inspire particular insights from the AI. This strategy guarantees that the resulting descriptions are both

exact and meaningful , providing clear and valuable summaries tailored to each table's content.

5.7.3 Report Organization

The organization of the report is essential for facilitating understanding and investigation of the information:

- **Title of the Report:** Each report starts with a clearly defined title, which sets the context for the consequent content.
- **Sections and Tables:** The report is organized into sections, each dedicated to a specific aspect of the analysis. Within each section, tables are displayed alongside their respective captions. This organizational strategy makes a difference in exploring the report and understanding the stream of data.
- **AI-Generated Descriptions:** Each table is accompanied by an AI-generated description placed immediately below the table. These descriptions serve to explain the significance of the information within the table, providing insights or summarizing patterns straightforwardly inside the report. This integration of descriptive content with financial tables improves the document's utility as a comprehensive analytical tool.

5.8 Testing and Results

5.9 Limitations and Challenges

5.10 Appendices

5.11 Future Work

5.12 Conclusion

Conclusion Générale

Nous espérons que ce document vous a aidé à rédiger votre rapport convenablement. Ce document vise à aider les étudiants à rédiger des rapports en Latex.

Bibliography

[1] Jade-Advisory, "Le site du jade-advisory," fev 2024. www.jade-advisory.com.

[2] A. Stöffelbaue, "Le site du medium," fev 2024. www.medium.com.

[3] superteams, "Le site du superteams," fev 2024. www.superteams.ai.

[4] A. Stöffelbaue, "Le site du medium," fev 2024. www.medium.com.

[5] huggingface, "Le site du knowledge.dataikuhuggingface," fev 2024. www.huggingface.com.

[6] knowledge.dataiku, "Le site du knowledge.dataiku," fev 2024. www.knowledge.dataiku.com.

[7] arxiv.org, "Table meets llm: Can large language models understand structured table data? a benchmark and empirical study," feb 2024. https://arxiv.org/abs/2305.13062.

[8] Y.-A. L. Borgne, "Openai vs open-source multilingual embedding models," feb 2024. https://towardsdatascience.com/openai-vs-open-source-multilingual-embedding-models-e5ccb7c90f05.

[9] S. B. G. V. Cormack, C. L. A. Clarke, "Reciprocal rank fusion outperforms condorcet and individual rank learning methods," feb 2024. https://plg.uwaterloo.ca/ gvcormac/cormacksigir09-rrf.pdf.

[10] A. P. Kevin Lin, John Hewitt, "Lost in the middle: How language models use long contexts," feb 2024. https://arxiv.org/pdf/2307.03172.pdf.

Annexe 1