Angela Colas
Carlos Eduardo Jedwab
João Pedro Aras

Synthetic text generation and retrieval text reduction based on embedding using natural language processing

São Paulo, SP
2024

Angela Colas
Carlos Eduardo Jedwab
João Pedro Aras

Synthetic text generation and retrieval text reduction based on embedding using natural language processing

Final thesis presented to the Department of Computer Engineering and Digital Systems at the Polytechnic School of the University of São Paulo to obtain the title of Engineer.

University of São Paulo – USP
Polytechnic School
Department of Computer Engineering and Digital Systems (PCS)

Supervisor: Prof. Dr. Edson Satoshi Gomi

São Paulo, SP
2024

Acknowledgments

This project would not have been possible without the invaluable guidance and support of Prof. Dr. Edson Satoshi Gomi, who served as the thesis advisor. His expertise, patience, and insights greatly contributed to shaping the direction and success of this work. Professor Gomi's encouragement and constructive feedback were instrumental in overcoming challenges and achieving meaningful results throughout this research.

Additionally, special thanks are extended to Vinícius Bitencourt Matos, a member of the current SeSO development team. His assistance, particularly in understanding the practical intricacies of the SeSO system and its implementation, provided essential contributions to this project. Vinícius's collaboration and shared knowledge played a key role in refining methodologies and ensuring the project's alignment with Petrobras's operational goals.

This research is deeply indebted to their collective efforts, and their support is sincerely appreciated.

Abstract

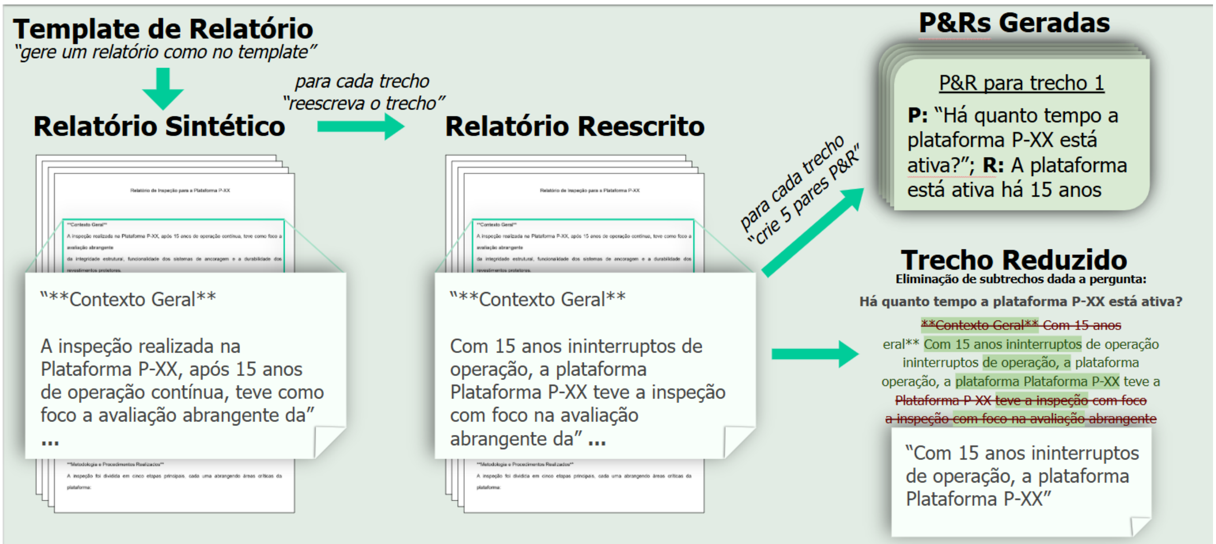

Figure 1 – Data Flow Diagram: Synthetic and augmented reports are generated and stored. Questions and answers are generated for evaluation. Upon user input, relevant passages are retrieved and summarized for answer generation.

The Semantic Search on Offshore Engineering (SeSO) system was developed by Petrobras to assist employees in retrieving and understanding information from dense and complex reports regarding offshore platform operations. Despite its functionality, SeSO faces two significant challenges: limited availability of real-world failure reports and issues with handling large, detailed documents during retrieval. This project seeks to enhance SeSO by addressing these challenges through the generation of synthetic failure reports, dataset augmentation via rewriting techniques, and the implementation of summarization methods to optimize retrieval for large passages.

The development process is structured around several key stages. First, synthetic reports are generated to expand the dataset, simulating realistic operational scenarios. These reports are further augmented through rewriting techniques to introduce linguistic and contextual variability. Questions and answers are then generated from both real and synthetic reports to test the retrieval system's performance. Upon user input, relevant passages are retrieved and summarized to provide concise and accurate answers, ensuring effective handling of lengthy documents and tables.

This workflow supports SeSO in delivering accurate and efficient question-and-answer capabilities while addressing operational constraints. The proposed methodologies not only improve data availability and retrieval quality but also lay the groundwork for future enhancements to Petrobras's Q&A system.

Contents

List of Figures

1 Introduction

Petrobras (PETROBRAS. . . ,) is a leading Brazilian company in the oil and gas industry, specializing in offshore oil extraction. The operation of its offshore platforms involves numerous potential failure points, both structural and mechanical. Such failures, when they occur, are documented in detailed reports that describe the problem and its potential causes. These reports are numerous and information dense, and so, employees are constantly in need of searching specific information among the vast amount of data. With the aim of speeding up the answering of employee's questions, Petrobras invested on developing an automated question-and-answer (Q&A) system named Semantic Search on Offshore Engineering (SeSO (GONCALVES et al., 2021)).

In essence, SeSO works similarly to a Retrieval-Augmented Generation (RAG (AWS, 2020)). The user can input a question in natural language, and the model finds relevant passages within the known reports, and then use the result as a basis for actually generating an answer. In its latest stage, system is functional, but does not always yield the best quality results. Both the retrieving and answer generation steps were in need of improving, and two main challenges were found in the way.

First, relating to the retrieval of passages, a smarter model is being designed. However, training and testing is limited by the scarcity of failure reports available. Due to the infrequency and complexity of certain failure events, the number of existing reports is often insufficient to build comprehensive models capable of supporting the Q&A system effectively. With that in mind, one of the objectives of this research is to explore the feasibility of using Artificial Intelligence (AI) to generate synthetic failure reports, thereby overcoming the limitations posed by the shortage of real-world data.

The other half of the system also faces an important challenge. Since the answer generation is based on giving an LLM a prompt containing both the question and the relevant context passages, if the full prompt becomes excessively large — for example when one of the relevant passages is a giant 1000 lines table — the LLM can forget portions of the input due to token limit constraints, resulting in incomplete or inaccurate outputs. To address this issue, a key objective of this research is to investigate whether using summarized versions of these passages can still produce high-quality answers. That is, whether we can find a strategy to condense text and tables without loosing vital information for question answering.

1.1 Motivation

In the oil and gas industry, ensuring operational safety and efficiency is critical due to the high stakes associated with resource extraction and management. Petrobras frequently encounters structural and mechanical failures in its offshore platforms. Thoroughly documenting and analyzing these failures is essential for identifying root causes and implementing effective preventive measures. However, the limited number of available reports—due to the infrequency and complexity of such events—poses a significant challenge for developing predictive models and supporting informed decision-making.

Generating synthetic failure reports using AI presents a viable solution to this data scarcity issue. However, one of the main challenges faced in this research involves the token limit constraints of LLMs when using large input prompts. Providing full-length reports as input often causes the model to truncate or disregard parts of the input, resulting in incomplete or less reliable outputs. This limitation undermines the effectiveness of AI models in producing high-quality synthetic data. Thus, the motivation behind this study is twofold: to explore the feasibility of generating synthetic reports to overcome data limitations, and to investigate whether using summarized versions of reports can still yield high-quality outputs. Addressing these issues will enhance the Q&A system's ability to provide reliable responses even in data-scarce scenarios, ultimately improving Petrobras's operational efficiency and safety.

1.2 Justification

The importance on this Thesis lies in the need to support Petrobras's internal Q&A system, which plays a critical role in managing and disseminating operational knowledge based on confidential reports. The scarcity of failure reports currently limits the system's ability to provide comprehensive responses, creating a barrier to effective failure detection and risk management. By generating high-quality synthetic reports, this research aims to expand the dataset available for AI training, thereby enabling the development of more robust models to support Petrobras's Q&A system.

Moreover, addressing the prompt size constraints of LLMs through the use of summarized inputs offers a novel solution to the limitations faced when using large prompts. This research not only contributes to the advancement of synthetic data generation but also has practical implications for other industries facing similar data scarcity and input size challenges. Ultimately, the results of this study will help optimize Petrobras's operational decision-making processes, promote more proactive risk management, and enhance overall platform safety and efficiency.

1.3 Objectives

The primary objective of this project is to enhance the efficiency and reliability of Petrobras's Semantic Search on Offshore Engineering (SeSO) question-and-answer (Q&A) system. This enhancement is pursued by addressing two critical challenges identified in the system's current implementation: the scarcity of real-world failure reports and the limitations imposed by prompt size constraints during answer generation. To achieve these overarching goals, the project is structured around the following specific objectives:

1. **Synthetic Generation of Failure Reports:**
   - *Initial Data Expansion:* Generate synthetic failure reports to initially expand the existing dataset, compensating for the limited availability of real-world reports.
   - *Report Augmentation:* Further increase the dataset size by augmenting both real and synthetic reports using rewriting techniques, ensuring diversity while maintaining the integrity of the information.
   - *Q&A Generation and Evaluation:* Create a substantial number of question-and-answer pairs derived from the synthetic reports. Utilize these pairs to evaluate and improve the quality of the retrieval system, ensuring that the augmented data effectively enhances the Q&A performance.
2. **Enhancement of Passage Retrieval:**
   - *Summarization Techniques for Future Integration:* Develop and assess summarization methods to condense lengthy passages and complex tables. Although primarily aimed for future adaptation, these techniques will provide valuable insights and foundational work for integrating efficient summarization into the real SeSO retriever based on the project's findings and diverse methodological approaches.
3. **Quality Assessment:**
   - *Evaluation Metrics and Procedures:* Establish comprehensive evaluation metrics and procedures to assess the coherence, relevance, and reliability of answers produced by the enhanced Q&A system. This includes comparing responses generated from synthetic data against those derived from real reports to ensure the synthetic augmentation effectively supports operational inquiries.

By systematically addressing these objectives, the project aims to deliver a robust and efficient Q&A system that significantly improves information retrieval and answer accuracy. This will support Petrobras employees in making informed decisions, thereby enhancing operational safety and effectiveness.

2 Overview of SeSO

2.1 SeSO's algorithm

2.1.1 Indexing block

The existing reports are all pdf files, some of them digitally made, but others scanned copies. Before any question answering process, we must address how to interpret the information within these reports, as described in the diagram 2

- First, the pdfs are transformed into a more structured format, in SeSO's case, xmls for text and tables, and image files for images within the reports. This is done via a software called Tornado.
- Secondly, the text and tables from each report is split into multiple smaller passages. These passages are what we will future aim to retrieve as possible candidates for having the answer to a given question from the user.
- The passages are then indexed with Elasticsearch (ELASTICSEARCH. . . , 2015). Once indexed, Elasticsearch can be queried using keyword-based queries, phrases, or filters. In other words, the index of the passages will be the test point to determine its relevancy to a given user question.

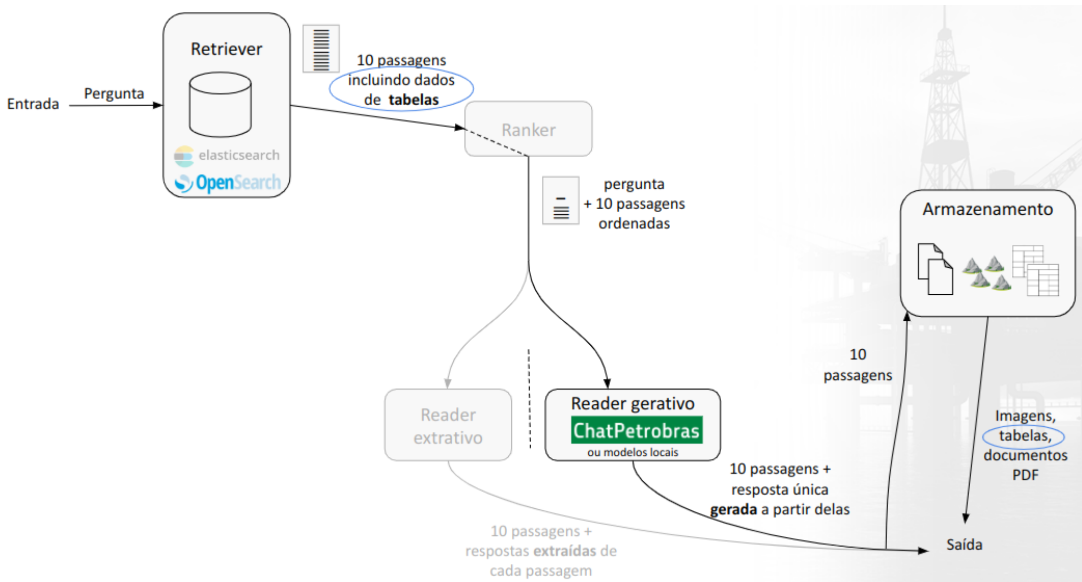

Figure 2 – Diagram describing the SeSO indexing step

2.1.2 Inference block

The actual answering of a user's question process is called the inference. The user must input its question and the system uses it and the indexed passages to generate an answer, as described in the diagram 3.

- First, SeSO needs to retrieve 10 of the most relevant passages that possibly contains the answer for the question. Note that it is possible for no passage to contain the answer, multiple passages to contain different parts of the answer, or a single passage contain the answer.
- It is possible for the retriever to return many passages and another block, the ranker, selects a smaller subset via scoring each passage and then selected the top 10 only. In some versions, for simplicity, the retriever can also just returns those 10 passages directly without the need for the ranker. (Note: the specific number of 10 passages was an abstract choice)
- With those context text extracts plus the original question, an LLM tried to generate an answer.

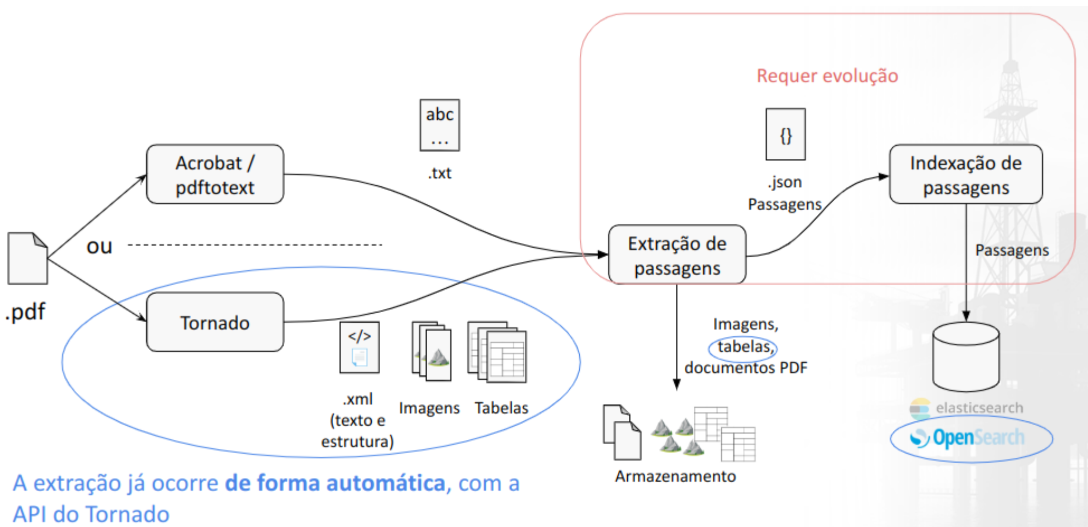

Figure 3 – Diagram describing the SeSO inference step

2.2 SeSO's challenges

While SeSO has successfully laid the foundation for an automated question-and-answer system tailored to Petrobras's needs, it faces significant challenges that hinder its optimal performance. These challenges are centered around two main components of the system: the retrieval of relevant passages and the generation of accurate answers.

Firstly, as outlined previously, the scarcity of available failure reports poses a critical issue for the retrieval step. The limited dataset reduces the model's ability to effectively retrieve relevant passages, particularly for rare or complex failure scenarios. This highlights the necessity of exploring alternative approaches, such as the generation of synthetic failure reports, to augment the dataset and improve retrieval performance.

Secondly, the process of answer generation encounters limitations stemming from the token constraints of large language models (LLMs). When context passages, especially those containing extensive data such as large tables, exceed the LLM's token limit, parts of the input may be truncated, leading to incomplete or unreliable answers. Summarizing these passages without losing critical information is, therefore, an essential avenue of research to mitigate this problem and ensure the system's reliability.

3 Key Concepts

3.1 Large Language Models (LLMs)

Large Language Models (LLMs) (LANGUAGE..., 2020), such as OpenAI's GPT series (CHATGPT..., 2022), have revolutionized natural language understanding and generation through self-attention-based transformer architectures. These models excel in capturing long-range dependencies in text and can generate coherent, contextually relevant outputs. Their pretraining on massive datasets provides broad contextual awareness, while fine-tuning on specific domains allows for highly specialized capabilities. In applications like Petrobras's Q&A system, LLMs serve dual roles: generating synthetic data to augment limited datasets and enhancing the retrieval of semantically relevant content for user queries.

However, the inherent token limits in LLMs pose a challenge. These models can process only a fixed number of tokens (e.g., 2048 tokens in GPT-3, 4096+ in GPT-4) at a time, necessitating careful input management to ensure key content is not truncated. Techniques such as retrieval text reduction and strategic prompt engineering can help maintain input coherence and contextual richness within these constraints.

There are different types of LLMs created by several companies. Among the best known today in 2024 we can cite Chat GPT, Claude, Gemini and Mistral. For its multilingual skills and its ability to see and be able to read PDFs which sets it apart from its competitors (MICHARD, 2024) we used the Chat GPT model in the development of our tools.

Below is a comparison table of the different models used in this project:

	gpt-3.5-turbo	gpt-4o-mini	gpt-4
Tokens	4,096	128,000	128,000
Princing in/out	0.15/0.60	0.50/1.50	2.50/10
Speed	Fast	Fast	Moderated
Tasks	Writing texts, solving simple to moderate problems.	Writing texts, solving simple to moderate problems.	Writing text, solving more complex problems, especially calculations.

We primarily used the 4o model for performing complex tasks that required high quality, which was essential for the continuity of the project. We aimed to leverage the strengths of each model based on specific needs. For example, for report generation , we favored the 4o model, as these reports were necessary at every stage of development and had to meet impeccable quality standards. Additionally, one group member already had a personal account allowing access to this model.

For text rewriting, we used the 4o Mini model, which offered a good balance between cost and performance. During the processing of text excerpts provided by the Retriever, we opted for the 3.5 Turbo model. Since this step was limited to summarizing data, the performance of 3.5 Turbo was sufficient. Although its cost-to-performance ratio was less favorable compared to 4o Mini, we occasionally used 3.5 Turbo because we already had credits available for this model, whereas using 4o Mini required purchasing additional credits.

The 4o model was not used for these simpler tasks because its cost was too high, particularly for processes generating a large number of tokens during each execution. A test was conducted using the 4o model on a report, but the quality difference compared to 4o Mini was not significant. Therefore, we opted for 4o Mini, which was more affordable and sufficiently effective for these use cases.

Finally, for the part dedicated to summarizing complex data, we once again used the 4o model. This step involved demanding tasks of extraction, analysis, computation, and restitution, which justified the use of this more powerful model.

3.2 RAG

The RAG (Retrieval-Augmented Generation) model is an advanced natural language processing technique that combines data extraction and text generation. It uses artificial intelligence models to retrieve relevant information from databases and then integrate it into contextualized responses written in a natural way.

Unlike conventional generative AI, which relies solely on its pre-trained knowledge, the RAG enriches its responses with specific and up-to-date external data, while clearly attributing the sources used. This transparency allows users to check source documents for accuracy or detail, thus increasing confidence in the answers provided. (AWS, 2020)

In addition to providing accurate results, the RAG consolidates the information extracted to generate unique explanations or instructions, adapted to the context. This makes it a powerful and reliable solution, combining the best of generative AI and information-mining-based AI. (COHESITY, 2021)

3.3 LangChain

LangChain (LANGCHAIN. . . , 2022) is a robust library designed to facilitate the seamless integration of LLMs into complex data processing workflows and decision-making pipelines. It offers tools to connect LLMs with tasks such as information retrieval, content generation, natural language understanding, and various AI-based applications.

Architecture of LangChain

- **Chains**: Enables the construction of sequential workflows where the output of one model serves as the input for subsequent processing or decision-making steps.
- **Memory**: Implements short-term memory mechanisms to maintain contextual consistency during multi-step interactions.
- **Agents**: Dynamic entities that decide which tools or methods to use automatically to solve specific tasks. For example, agents can retrieve data from a database, use an external API, or directly process text with an LLM.
- **Retrievers**: Specialized modules for semantic search, integrating large volumes of data with language models. They support strategies like vector search, embedding-based retrieval, and full-text search.
- **Toolkits**: Allow integration with external tools (e.g., APIs, databases, text files) to enhance the system's capabilities with additional contextual data.

Advanced Features

- **Prompt Templates**: Defines templates for creating robust and efficient prompts, ensuring the model receives the necessary context for accurate responses. It supports dynamic variables like `{user_input}` or `{contextual_data}`.
- **Embeddings**: Integrates embedding algorithms for semantic search and relevance ranking. This enables precise comparisons between text snippets to retrieve meaningful information from large datasets.
- **Data Augmentation**: Enriches model capabilities by integrating external data sources, such as tables, documents, or APIs. For example, synthetic reports can be enhanced with live data from external databases.
- **Evaluation Tools**: Provides mechanisms to assess the quality of generated responses, including relevance, coherence, and adherence to prompt requirements.

3.4 Embeddings and Semantic Similarity

Embeddings are numerical representations of textual elements such as words, phrases, or sentences, mapped into a high-dimensional space where semantic relationships are encoded. These embeddings enable machines to perform mathematical operations to compare and relate words based on their meanings, rather than just their syntax or form.

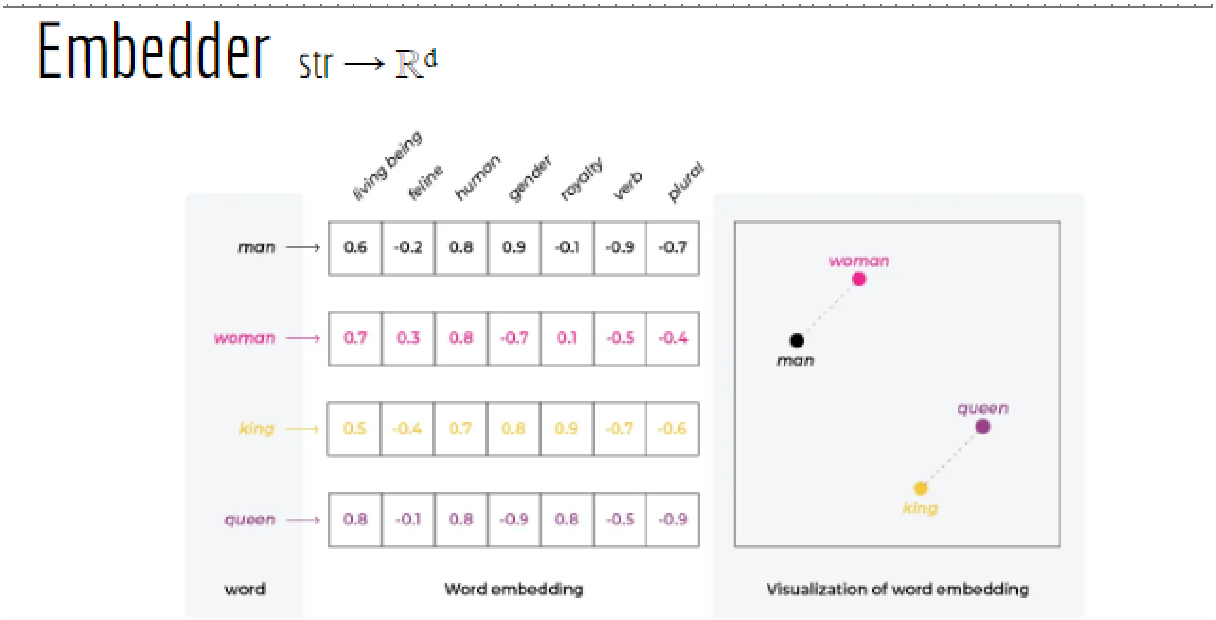

Figure 4 – Illustration of the embedding process and visualization. Words are mapped into a vector space, encoded with semantic attributes, and visualized in two dimensions.

Word-to-Vector Mapping

The table on the left of Figure 4 demonstrates how specific words (*man*, *woman*, *king*, *queen*) are transformed into vectors based on multiple semantic dimensions such as *living being*, *gender*, *royalty*, and *plural*. Each dimension captures an attribute of the word, assigning numerical weights that represent its contribution. For example:

- **Man**: $[0.6, -0.2, 0.8, 0.9, -0.9, 0.7, -0.7]$
  - High values for *living being* (0.6) and *human* (0.8), reflecting its connection to humanity.
  - A negative value for *plural* (-0.7), indicating its singular nature.
- **Woman**: $[0.7, 0.3, 0.8, -0.7, 0.1, 0.5, -0.4]$
  - Similar to *man* but differs significantly in the *gender* feature (0.3), capturing the contrast between masculine and feminine entities.

This transformation allows for precise mathematical operations to analyze word relationships.

Embedding Table

The middle table in Figure 4 highlights how embeddings encode subtle differences and similarities. For instance:

- The vectors for *king* and *queen* are very similar in most dimensions, particularly *royalty* (0.8 for both), but differ in *gender* (-0.1 for *queen*, -0.4 for *king*). This reflects their analogous roles while capturing their distinct gender attributes.
- The relationships between *man* and *woman* show a similar pattern, with differences in *gender* (0.3 vs. -0.2) but alignment in *human* (both 0.8).

Visualization in Two Dimensions

The right panel in Figure 4 visualizes the word embeddings in a simplified 2D space. Words with closer semantic meanings, such as *man* and *woman* or *king* and *queen*, are placed near each other. This proximity reflects shared features. Additionally:

- The relative positions of *king* and *queen* versus *man* and *woman* illustrate the concept of analogical relationships. For example, the directional vector difference between *king* and *queen* closely matches the difference between *man* and *woman*. This property enables embeddings to solve analogies like:

  "man" : "woman" :: "king" : "queen"

Applications and Benefits of Embeddings

By leveraging embeddings, machines can compute semantic similarity by measuring distances or angles between vectors. For instance:

- The cosine similarity between *king* and *queen* would be high due to their vector alignment in key dimensions such as *royalty*.
- Differences in attributes like *gender* or *plural* enable systems to differentiate subtle nuances.

This capability powers applications like search engines, recommendation systems, and natural language processing tools. By capturing both relationships and attributes, embeddings bridge the gap between human language and computational understanding.

3.5 Similarity Metrics: Euclidean and Cosine Similarity

In the context of embeddings, similarity metrics are critical for quantifying the relationships between vectors. These metrics enable the understanding of how closely related two vectors are in a multidimensional space, forming the foundation for various natural language processing tasks. Two widely used similarity metrics are **Euclidean distance** and **cosine similarity**, and they can be illustrated using the chart below.

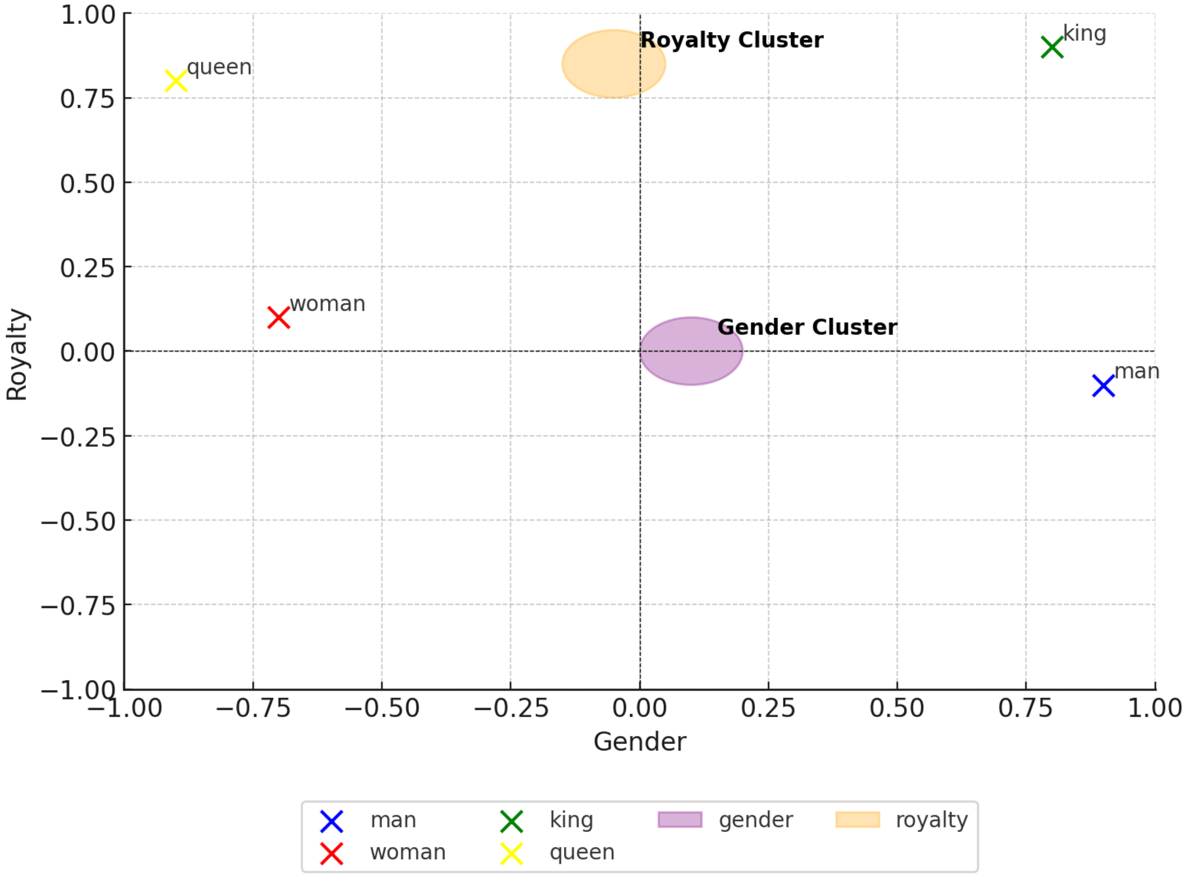

Figure 5 – Visualization of embeddings for "man," "woman," "king," and "queen," along with their respective clusters ("Gender" and "Royalty").

Euclidean Distance

Euclidean distance measures the straight-line distance between two points in the embedding space. It is defined as:

$$d_E(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$

Where:

- $\vec{a}$ and $\vec{b}$ are vectors representing two embeddings.
- n is the dimensionality of the embedding space.

In the chart (Figure 5):

- The Euclidean distance between *man* and *woman* is smaller than the distance between *man* and *queen*, reflecting their closer semantic relationship in terms of gender.
- Similarly, *king* and *queen* are positioned closer together within the "Royalty" cluster, illustrating their shared semantic features.

Cosine Similarity

Cosine similarity evaluates the orientation of two vectors by measuring the cosine of the angle between them. It is given by:

$$\mathrm{sim}_{\cos}(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|}$$

Where:

- $\vec{a} \cdot \vec{b}$ is the dot product of the vectors.
- $\|\vec{a}\|$ and $\|\vec{b}\|$ are the magnitudes of the vectors.

In the visualization (Figure 5):

- Points closer within the same cluster (e.g., *king* and *queen*) have vectors that are more aligned, resulting in a higher cosine similarity.
- Despite having a larger Euclidean distance, *man* and *king* might share a moderate cosine similarity due to their alignment in certain dimensions like "human" or "leader."

Use of Both Metrics

While Euclidean distance captures the absolute differences between embeddings, cosine similarity emphasizes their proportional relationships. The chart effectively demonstrates both concepts:

- **Euclidean Distance:** Measures the physical separation between words, useful for tasks like clustering and anomaly detection.
- **Cosine Similarity:** Captures semantic relationships and is widely used in recommendation systems and semantic search.

Insights from the Visualization

- The "Gender" cluster groups *man* and *woman*, while the "Royalty" cluster groups *king* and *queen* showing how embeddings encode shared features and relationships.

- The directional difference between *man* and *woman* is similar to the difference between *king* and *queen*, illustrating how embeddings support analogy tasks.
- Words far from a cluster center can be identified as semantically distinct or less related.

4 Methodology

The chosen methodology reflects the need to generate synthetic data that compensates for the scarcity of real-world failure reports, as well as to optimize the retrieval process to ensure that the Q&A system retrieves the most relevant information for complex queries. The two-phased approach aligns with the primary objectives: expanding the dataset with synthetic failure reports and enhancing the retrieval system through text reduction.

4.1 Phase 1: Synthetic expansion of the Reports Dataset

Given the limited availability of real failure reports, synthetic text generation serves as a solution to augment the dataset. This phase is designed to create realistic scenarios that mimic the characteristics of operational failures documented by Petrobras. The following steps outline the methodology for this phase:

1. **Initial study:** Study the overall structures of the existing reports, design a few templates around the structures. This step includes capturing common essential fields on the reports dataset such as failure description, causes, and context, as well as common structure elements such as sections, metrics, tables, etc.

2. **Synthetic Report Generation:** Use Large Language Models (LLMs) prompting to generate texts and tables that fit the templates. It must produce reports that align with the structural and contextual characteristics of real reports.

This phase enriches the dataset, thereby enhancing the Q&A system's ability to respond to complex queries that require detailed operational context.

4.2 Phase 2: Synthetic Q&A generation

For a more complete evaluation of the previous phases, as well as for the testing and validation of the next phase, we need a big number of Q&A examples. Therefore, we must produce our own dataset of questions and answers.

1. **Q&A Generation**: Using our dataset, generate some number n of question and answer pairs for each passage of each report.

4.3 Phase 3: Report Reformulation System

With the aim of expanding the dataset even further, we propose an augmentation phase where original and/or synthetic reports can be transformed with rewriting techniques, as to convey equal information with slightly differentiated text.

1. **Report Augmentation:** Using either the existing reports and/or the synthetic ones from the previous step, generate different versions of each, where the text conveys the exact same information but with modified phrasings.

2. **Quality Assessment:** Generated reports undergo evaluation using metrics such as readability, coherence, and similarity to real-world reports. This step may require iterative adjustments to the model's parameters to ensure the quality and relevance of synthetic data for training the Q&A system.

4.4 Phase 4: Reduction of Prompt Size

This phase involves the development of a passages reduction algorithm to be deployed between the retrieval and answer generation steps of SeSO. For that, we proposed the following steps:

1. **Implement passage extractive summarization (LIU; LAPATA, 2019)** Develop a stand alone code capable of receiving a question and one passage and assessing weather the passage should be shortened, if so by how much, and then proceeds to detect and remove irrelevant text (for text) or columns (for tables) that won't help with the answer.

2. **Quality Assessment:** Compare different approaches and their results when it comes to removing actually irrelevant information.

3. **Deployment:** Implement the passage processing code in the SeSO application.

The implementation leverages a layered retrieval approach comprising three main components: **Splitter**, **Embedder**, and **Scorer**. Each component plays a crucial role in efficiently identifying and prioritizing relevant information from lengthy documents. We will go into more detail in the Development chapter, but here is a quick overview:

- **Text Splitting (Splitter)**: Divides passage into smaller, manageable segments using various splitting techniques like overlapping and non-overlapping methods to preserve context.

- **Embedding Creation (Embedder)**: Converts text segments into high-dimensional vectors that encode semantic meaning for accurate comparison.

- **Similarity Scoring (Scorer)**: Calculates the similarity between embeddings to rank and retrieve the most relevant segments.
- **Filtering (Filterer)**: Removes the least important segments from the passage.

By reducing the text in this layered approach, the system minimizes token consumption and enhances response relevance, making the Q&A system more effective under constraints of data volume and prompt size.

4.5 Evaluation Procedures

To ensure the effectiveness of the proposed methodology, a series of tests were conducted, simulating real Q&A queries. These tests are designed to evaluate the retrieval system's accuracy, efficiency, and ability to handle various semantic relationships between queries and document segments. The evaluation includes:

- **Relevance Testing**: The retrieval scores are analyzed to validate that the system ranks relevant information higher than less pertinent content, ensuring that retrieved data aligns with the user's query.
- **Efficiency of Retrieval**: Token usage and prompt sizes are monitored to confirm that the retrieval text reduction effectively optimizes performance while preserving answer quality.
- **Synthetic Report Validation**: Responses generated using synthetic reports are compared to those based on real data, assessing the reliability and quality of synthetic data in supporting operational inquiries.

This structured methodology provides a comprehensive and replicable approach for creating an efficient, data-optimized Q&A system. By following this design, Petrobras gains an enhanced ability to address operational issues with quick, accurate responses, even with limited access to real-world data.

5 Requirement Specification

This chapter outlines the updated requirements for the project, incorporating the progress made in the development stages and the insights gained from the implementation process. These requirements are structured into functional and non-functional categories, reflecting the project's current focus on enhancing the Q&A system through synthetic text generation and retrieval optimization.

5.1 Functional Requirements

The system must fulfill the following functional objectives:

1. **Synthetic Report Generation:**
    - Generate synthetic reports for the initial expansion of the dataset, simulating real operational scenarios.
    - Include diverse linguistic and structural variations to enrich the dataset with broader coverage of failure scenarios and inspection contexts.
2. **Data Augmentation and Reformulation:**
    - Implement text rewriting techniques to produce alternative versions of existing synthetic and real reports, ensuring variability while preserving information integrity.
    - Create augmented datasets that expand the linguistic and structural diversity of the training data.
3. **Q&A Dataset Generation:**
    - Generate question-and-answer pairs from synthetic reports to facilitate the evaluation and validation of the retrieval system.
    - Develop questions that are contextually relevant and answers that are explicitly derived from report passages.
4. **Retrieval Summarization:**
    - Design and test summarization techniques to reduce the size of lengthy text passages or tables without losing critical information.
    - Integrate summarization methods as a foundational experiment for potential future adaptation in the SeSO retrieval module.

5. **Evaluation and Deployment:**
    - Conduct thorough quality assessments of the generated synthetic data and retrieval outputs, using predefined metrics.
    - Prepare the synthetic text generation and summarization modules for seamless integration into the SeSO framework.

5.2 Non-Functional Requirements

The non-functional requirements ensure the effectiveness, reliability, and usability of the system beyond its core functionalities. These include evaluation and quality assessment, which are detailed further below.

1. **Data Integrity:** All generated synthetic reports and reformulated texts must preserve the original information's integrity, ensuring no critical data is altered or lost during generation or rewriting processes.
2. **Scalability:** The system should handle a minimum of 100 synthetic reports and at least 200 to 500 question-and-answer pairs while maintaining computational efficiency.
3. **Diversity:** The augmented datasets must exhibit high linguistic and contextual variance, enhancing their value for training and evaluating the retrieval system.
4. **Efficiency:** API usage and computational operations must remain cost-effective and optimized for large-scale text generation and summarization tasks.
5. **Evaluation and Quality Assessment:**
    - *Summarization Effectiveness:* The summarization methods must achieve high similarity between answers derived from summarized and full text segments. Using a threshold of 0.9 cosine similarity as a benchmark, the summarization should maintain critical information while effectively reducing text length.
    - *Synthetic Data Validation:* Synthetic reports, including reformulated versions, should undergo rigorous quality checks to ensure readability, coherence, and relevance. Metrics such as accuracy and Mean Reciprocal Rank (MRR) will evaluate the retriever's performance across real and synthetic datasets. The measure of these metrics with the new dataset must be better than or equal to the one measured with the initial dataset.
    - *Statistical Analysis of Results:* The system should demonstrate statistically significant improvements in summarization and retrieval quality over baseline methods, confirmed through tests such as chi-squared analysis.

- *Balanced Dataset Utilization:* Both original and rewritten reports must contribute equally to the retriever's performance, ensuring no bias toward one dataset version during testing phases.

6 Development

The development phase of this project was focused on implementing the synthetic text generation process and optimizing the retrieval system to handle complex queries in a data-constrained environment. This phase not only involved the technical integration of various components, but also addressed the confidentiality challenges associated with handling sensitive operational reports of Petrobras. The focus was on creating a solution that could enhance the Q&A system's ability to provide precise and context-rich answers, even with limited access to real-world data.

6.1 Phase 1: Expansion of the Reports Dataset

6.1.1 Synthetic Report Generation for Retriever Optimization

As explained earlier, a significant part of the work involves expanding the database of reports. This step aims to address an issue raised by Petrobras collaborators.

The Retriever of the system does not always manage to retrieve relevant passages for a query, or it ranks them with low relevance, resulting in these passages being excluded from the top positions by SeSO. For instance, for a question about inspections "on 100% of the lines," the relevant passages are correctly identified. However, if the wording changes to "on all the lines," these same passages no longer appear among the top results. The creation of synthetic data seeks to address this issue by:

- Adding different linguistic variations;
- Enriching the database with scenarios and concepts less frequently found in the original data.

For confidentiality reasons, the original Petrobras reports could not be used with public LLM systems. A prior step was necessary to generate anonymized data. The traditional Retriever, which uses the BM25 algorithm, does not require pre-training on data. Similarly, the new approach using embeddings with OpenSearch does not require prior model training. This allows the anonymized reports to be directly used with the Retriever, enabling direct comparison of the results with those obtained using the real reports. We therefore divided our generation into two parts:

- **Implementation of Synthetic Reports for Offshore Inspection:** Creation of a new anonymized report database based on the real reports;
- **Report Reformulation System:** Creation of a report reformulation code that takes reports as input and provides a reformulation of the reports as output.

6.1.2 Implementation of Synthetic Reports for Offshore Inspection

The objective of this process was to generate synthetic reports replicating the structure, complexity, and content of real offshore inspection reports. These reports encompass routine checks, integrity assessments, and failure scenarios, thereby enhancing the SeSO system's capabilities in retrieving contextually relevant responses to complex queries.

Template Analysis and Development

The process began with an in-depth analysis of 39 original reports provided by Petrobras. This analysis aimed to identify recurring structures, critical data points, and essential sections, leading to the classification of six distinct report models. Each model served as a blueprint for synthetic report generation. The templates developed included the following components:

- **Initial Data Table:** Details such as contract information, platform identifiers, and field locations.
- **Objectives and Operational Contexts:** Comprehensive descriptions of inspection goals and the operational environment.
- **Detailed Inspection Procedure Descriptions:** Step-by-step accounts of inspection methodologies and measurements.
- **Integrity Analysis:** Sections highlighting potential issues like corrosion, structural damage, and wear.
- **Recommendations and Preventive Actions:** Actionable suggestions based on identified issues.

Prompt Engineering

Custom prompts were crafted for each of the six identified report models to guide the GPT-4o model in generating synthetic content aligned with the predefined templates. The prompt engineering process involved:

- Crafting initial prompts tailored to each report model, specifying the required sections and their content.
- Iteratively refining prompts based on the quality of the generated reports. Adjustments were made to ensure coherence, technical accuracy, and adherence to operational standards.
- Incorporating additional constraints and examples in the prompts to simulate complex scenarios, describe observed failures, and propose actionable recommendations.

The original reports were anonymized, retaining only their structural elements and excluding any proprietary or sensitive content. GPT-4o was tasked with inventing the context and data necessary to populate the synthetic reports, ensuring the results were realistic but entirely fabricated.

Report Generation and Dataset Expansion

For each model, the GPT-4o model was tasked with generating at least 10 synthetic reports, resulting in a total of 60 reports. The process involved:

- Generating reports using the refined prompts, ensuring diversity within each type while maintaining consistency with the original templates.
- Applying data augmentation techniques to introduce variability. This included modifying data points, altering scenario descriptions, and varying linguistic styles.

Figure 6 presents a visual summary of the categorized report types and their relevance to the SeSO system. The categorization was designed to ensure adequate representation and balance within the synthetic dataset.

Figure 6 – Categorized report types and their relevance. Image credit: Nicolas Marotti Moreira.

6.1.3 Process of Report Generation

The generation of synthetic reports followed a systematic and reproducible workflow, leveraging the templates and prompts detailed earlier while introducing specific steps to ensure clarity and consistency for replication:

1. **Analyzing and Structuring Input Data:** The process began by examining 39 real offshore inspection reports to identify common structures, critical components, and essential elements. These reports were categorized into the following types:
   - **Service Reports:** Prototypical reports already processed in the dataset presentation, totaling 13 examples.

   - **Anomaly Reclassification Reports:** Sets of recommendations and analyses, including compiled inspection reports, with 5 examples.
   - **Measurement Reports:** Reports focusing on specific components, totaling 2 examples.
   - **Intervention Reports:** Technical intervention reports, such as shallow dive operations, totaling 6 examples.
   - **Inspection Reports:** Documents similar to anomaly reclassification reports but focused on specific line inspections, totaling 7 examples.
   - **Miscellaneous Reports (categorized as Öthers)̈:** Including descriptive memorials, executive procedures, and periodic reports, totaling 6 examples.

   Only structural details were retained, and sensitive information was anonymized. Each report was categorized into one of these predefined models, as outlined in the Template Analysis section.

2. **Developing and Refining Prompts:** For each report model:
   - An initial prompt was designed based on the identified structure, specifying required sections (e.g., Initial Data Table, Integrity Analysis).
   - Example: A prompt for an integrity analysis report might include: *"Generate a report containing an Initial Data Table, followed by a detailed Integrity Analysis highlighting corrosion and structural wear."*
   - Generated outputs were reviewed for alignment with the intended structure. When outputs deviated, additional constraints or examples were incorporated into the prompt, iteratively refining the instructions.

3. **Generating Synthetic Reports:** Using the refined prompts, GPT-4o was tasked with generating synthetic reports. Parameters such as `temperature = 0.7` and `max tokens = 800` were used to balance creativity and relevance. Synthetic reports were either based on a single type, a mixture of types, or entirely invented models, ensuring comprehensive coverage of possible scenarios. Each model required the generation of at least 10 distinct reports, resulting in a total of 60 synthetic reports.

4. **Ensuring Data Variability:** To enrich the dataset:
   - Scenario descriptions were varied to introduce diverse operational contexts (e.g., inspections in different environmental conditions).
   - Data points, such as metrics and identifiers, were altered within plausible ranges while maintaining consistency with the model's structure.

- Linguistic variations, such as rephrased recommendations or different narrative styles, were introduced to simulate multiple authorships.

5. **Final Compilation:** The resulting reports were reviewed to confirm they adhered to the predefined structural templates and exhibited sufficient variability. The anonymized structures ensured no proprietary data from the original reports was carried into the synthetic dataset.

6.1.4 Example of a Synthetic Report Prompt and Generated Result

GPT Prompt:

```
Generate a synthetic offshore inspection report with the following structure:
- Title: "Inspection Report for Platform P-XX"
- Initial Data Table with key operational metrics (e.g., pressure, depth,
  temperature)
- Context and Objectives of the inspection
- Detailed procedural descriptions with observations and data measurements
- Integrity analysis highlighting identified issues and providing
  recommendations
- Annexes containing supplementary tables and material lists
```

Generated Report:

Inspection Report for Platform P-XX

Initial Data Table

Metric	Value	Unit
Operational Pressure	150	bar
Depth	3500	meters
Temperature	85	°C
Inspection Date	2024-11-23	
Weather Condition	Clear skies	

Context and Objectives

The objective of this inspection was to assess the structural and operational integrity of Platform P-XX. The inspection aimed to ensure compliance with safety standards and operational requirements, as well as to evaluate the performance of key components, including anchoring systems, pipelines, and mechanical structures. Specific attention was given to identifying potential safety risks, ensuring uninterrupted operation, and maintaining compliance with industry regulations.

Detailed Procedural Descriptions

Step 1: General Visual Inspection

A complete visual inspection of all accessible areas of Platform P-XX was carried out. The inspection began with a visual examination of the topside structures, including the helideck, accommodation modules, and mechanical components. Observations noted during this step included minor surface rust on several joints, which will require further monitoring but does not pose an immediate threat.

Step 2: Subsea Inspection with ROV

An ROV was deployed to assess subsea components, with a particular focus on the integrity of

Figure 7 – Generated synthetic report for Platform P-XX.

6.1.5 Integration of Checklist Tables in Synthetic Reports

The integration of checklist tables into synthetic reports mirrors the detailed format of the original Petrobras reports, which used structured questions to evaluate inspection components.

Purpose of Checklist Tables

- Standardize inspection data collection.
- Support query-based retrieval in the SeSO system.
- Highlight anomalies for further review.

Checklist Tables:

Tabela de Questões com Respostas Detalhadas

1. LIMPEZA DOS COMPONENTES INSPECIONADOS

1.1 Os componentes foram devidamente limpos antes da inspeção? - (X) Sim ( ) Não

1.2 Houve necessidade de remoção de resíduos específicos? - (X) Sim ( ) Não

1.3 Quantidade de resíduos removidos? - 20 kg

1.4 Quais métodos de limpeza foram utilizados? - Lavagem com pressão e solventes industriais

1.5 Os resíduos causaram danos estruturais? - ( ) Sim (X) Não

2. Fairleader

2.1 Estado de conservação

2.1.1 Existem sinais visíveis de desgaste? - (X) Sim ( ) Não

2.1.2 As superfícies estão livres de corrosão? - ( ) Sim (X) Não

2.1.3 Quais superfícies apresentaram corrosão? - Superfície externa e base

2.2 Lubrificação

2.2.1 O sistema de lubrificação está funcionando corretamente? - (X) Sim ( ) Não

2.2.2 Há vestígios de excesso de lubrificação ou vazamentos? - (X) Sim ( ) Não

2.2.3 Foi necessária alguma correção no sistema? - (X) Sim ( ) Não

2.2.3.1 Tipo de correção aplicada: Ajuste das vedantes

Figure 8 – Generated Checklist Tables

The checklist includes:

- Questions like "Was cleaning performed?" with yes/no answers.
- Corrective measures and observations.

- Enhancement of structured queries and anomaly detection.

6.1.6 Iterative Generation of Large Tables

Some synthetic reports include extensive tables, iteratively generated for realism. This approach ensures consistency across various sections like tension, coating measurements, and structural analyses.

Generating process

- Iterative prompts guided GPT-4 to fill templates.
- Categories like "Measurements" and "States" informed structure.

Large Tables:

Conexão Flex TDP	Conexão flexivel	1106	160	Operacional	17/11/2
Cabo Aço Tipo 1	Cabo reforçado para ancoragem	1309	216	Operacional	11/11/2
Manilha C3	Manilha de alta resistência	1359	199	Manutenção	26/11/2
Conexão Flex TDP	Conexão flexivel	1106	160	Operacional	17/11/2
Cabo Aço Tipo 1	Cabo reforçado para ancoragem	1309	216	Operacional	11/11/2
Manilha C3	Manilha de alta resistência	1359	199	Manutenção	26/11/2
Conexão Flex TDP	Conexão flexivel	1106	160	Operacional	17/11/2
Cabo Aço Tipo 1	Cabo reforçado para ancoragem	1309	216	Operacional	11/11/2
Manilha C3	Manilha de alta resistência	1359	199	Manutenção	26/11/2
Conexão Flex TDP	Conexão flexivel	1106	160	Operacional	17/11/2
Cabo Aço Tipo 1	Cabo reforçado para ancoragem	1309	216	Operacional	11/11/2
Manilha C3	Manilha de alta resistência	1359	199	Manutenção	26/11/2
Conexão Flex TDP	Conexão flexivel	1106	160	Operacional	17/11/2
Cabo Aço Tipo 1	Cabo reforçado para ancoragem	1309	216	Operacional	11/11/2
Manilha C3	Manilha de alta resistência	1359	199	Manutenção	26/11/2
Conexão Flex TDP	Conexão flexivel	1106	160	Operacional	17/11/2
Cabo Aço Tipo 1	Cabo reforçado para ancoragem	1309	216	Operacional	11/11/2
Manilha C3	Manilha de alta resistência	1359	199	Manutenção	26/11/2
Conexão Flex TDP	Conexão flexivel	1106	160	Operacional	17/11/2
Cabo Aço Tipo 1	Cabo reforçado para ancoragem	1309	216	Operacional	11/11/2
Manilha C3	Manilha de alta resistência	1359	199	Manutenção	26/11/2
Conexão Flex TDP	Conexão flexivel	1106	160	Operacional	17/11/2
Cabo Aço Tipo 1	Cabo reforçado para ancoragem	1309	216	Operacional	11/11/2
Manilha C3	Manilha de alta resistência	1359	199	Manutenção	26/11/2
Conexão Flex TDP	Conexão flexivel	1106	160	Operacional	17/11/2
Cabo Aço Tipo 1	Cabo reforçado para ancoragem	1309	216	Operacional	11/11/2
Manilha C3	Manilha de alta resistência	1359	199	Manutenção	26/11/2
Conexão Flex TDP	Conexão flexivel	1106	160	Operacional	17/11/2
Cabo Aço Tipo 1	Cabo reforçado para ancoragem	1309	216	Operacional	11/11/2
Manilha C3	Manilha de alta resistência	1359	199	Manutenção	26/11/2
Conexão Flex TDP	Conexão flexivel	1106	160	Operacional	17/11/2

Figure 9 – Iteratively Generated Large Tables

6.1.7 Number of Reports Generated and Justification

A total of 60 synthetic reports were generated during this process, compared to the 39 original Petrobras reports analyzed. This number was chosen to provide nearly double the volume of data, ensuring diversity and robustness for the training and evaluation of the SeSO system.

The increased number of synthetic reports allowed for greater variability in operational scenarios, including:

- Rare failure cases,
- Routine inspections, and
- Detailed integrity analyses.

This ensured that the SeSO system could handle a wide range of queries and situations effectively.

6.1.8 Anonymization Strategy and Justification

To safeguard proprietary information and comply with confidentiality requirements, the synthetic reports were carefully anonymized. Key steps in the anonymization process included:

- **Generic Identifiers:** Specific platform names, equipment, and operational incidents were replaced with generic identifiers (e.g., "Platform P-XX").
- **Fabricated Data Points:** All numerical and operational metrics were fabricated based on predefined distributions and ranges derived from an analysis of the original reports, ensuring no direct replication.
- **Abstraction of Context:** Operational details were generalized to reflect plausible scenarios without revealing specific strategies or sensitive practices.

6.1.9 Decision to Exclude Images

Images and visual elements were deliberately excluded from the synthetic reports. This decision was based on the functionality of the SeSO system, which focuses on text-based retrieval. Including images would not enhance the system's performance and could introduce unnecessary complexity in report generation. Instead, the reports emphasize textual descriptions and detailed tabular data, aligning with the retrieval-focused design of the SeSO system.

6.1.10 Technical Challenges and Solutions

The development process encountered several technical challenges, particularly related to the token limitations of LLMs and the need to maintain high-quality input data. Addressing these challenges involved:

- **Managing Prompt Size Limitations**: LLMs have constraints on the number of tokens they can process in a single prompt. To mitigate this, the retrieval system was optimized to reduce input size through retrieval text reduction, allowing the most relevant content to be prioritized and ensuring that essential context was preserved.

- **Optimizing Embedding Computation**: Given the need to process large volumes of text, the Embedder module was adjusted to balance computational efficiency and accuracy. Techniques such as batch processing and vector caching were implemented to minimize memory consumption without sacrificing the quality of the similarity analysis.

- **Ensuring Data Security during Testing**: Given the sensitivity of Petrobras's operational reports, data security was a primary concern during testing phases. All synthetic reports generated externally were validated for confidentiality before being integrated into the testing process, ensuring that no sensitive details were exposed during the iterative development process.

6.1.11 Testing and Validation

Testing and validation were integral to ensuring that the system met the stringent requirements of Petrobras's Q&A operations. This phase involved simulating realistic queries and measuring the retrieval system's performance using metrics such as relevance, accuracy, and response time:

- **Relevance Testing**: Synthetic and real reports were used to evaluate the retriever's ability to rank the most relevant information higher, ensuring that responses aligned closely with the user's questions.

- **Quality Assessment of Synthetic Reports**: The synthetic reports were evaluated against real reports using predefined metrics, including coherence, readability, and structural similarity. This step validated that the generated reports provided a realistic basis for improving the Q&A system's training.

- **Performance under Constraints**: Special focus was placed on evaluating the system's performance when operating close to the LLMs' token limits, ensuring that the retrieval text reduction approach maintained response quality even when input data was condensed.

The development phase concluded with a validated retrieval system capable of handling Petrobras's complex operational queries, backed by a robust methodology for generating and leveraging synthetic data to overcome the challenges posed by data scarcity.

6.2 Phase 2: Synthetic Q&A Generation

It is essential we create a sufficient amount of Question and Answers pairs extracted from the dataset we have. This would allow for the testing of the retriever, and the testing of the r and summarizer from the next phase in this project.

6.2.1 Naive Approach

In the rewriting phase, we created a system that changes passages into rewritten versions using specific prompts. For this approach, we reused that system but modified it to work differently. Instead of rewriting passages, we changed the prompt to create three questions and answers for each passage. This way, the system now generates three question-and-answer pairs for every passage in each report. This simple change makes it easier to extract useful information and organize it in a way that is more interactive and helpful for users.

GPT Prompt

```
prompt_template = """
Voce e uma IA projetada para gerar pares de perguntas e respostas.
Abaixo voce encontra um trecho retirado de um relatorio.
O assunto da pergunta deve estar contido no trecho, e deve ter especificacoes
    relacionadas ao contexto geral.
A resposta deve estar contida no trecho.
Nao invente informacoes, use apenas dados explicitamente citados no texto.
Caso a pergunta gerada nao seja respondida no texto, a resposta deve ser "O
    trecho encontrado nao responde a pergunta".

Formato de saida:
Pergunta 1: [Sua pergunta aqui]
Resposta 1: [Sua resposta aqui]
Pergunta 2: [Sua pergunta aqui]
Resposta 2: [Sua resposta aqui]
(continue esse formato ate a Pergunta 3 e a Resposta 3)

Trecho:
{{text}}
"""
```

Results

Unfortunately, the results of this approach fell short of expectations. A significant issue was that many of the generated questions explicitly referenced the context in which the LLM was working. For example, they often included phrases such as "In the text below..." or "According to the passage..." that directly acknowledged the presence of the input text. This style of questioning is fundamentally flawed for our purposes, as it reveals the existence of the passage or report, making it clear that the question is tied to a specific input text.

The goal of this system is to create questions that appear natural and self-contained, as if they were being asked without any prior knowledge of the passage or even the report itself. Questions should be framed in a way that reflects a broader understanding or curiosity about the subject, without directly pointing to or referencing the source text. This failure to decouple the generated questions from the input passage undermines the usability of the output for scenarios where the questions are meant to stand independently or simulate a real-world user query. Consequently, this aspect of the approach requires significant refinement to meet the desired standards.

6.2.2 New Approach

With that in mind, we have successfully refined the prompt to better address this issue. Our adjustments now guide the LLM toward generating the desired output without inadvertently reinforcing undesired behaviors. Previously, including instructions like "don't reference the text directly" often led to the opposite effect; ironically, highlighting that possibility made the LLM more likely to do exactly that.

GPT Prompt

After many iterations of editing the prompt to fine tune the generation, we have:

```
template_prompt = """
Voce e uma IA projetada para gerar pares de perguntas e respostas.
Abaixo voce encontra um trecho retirado de um relatorio.
O assunto da pergunta deve estar contido no trecho, e deve ter especificacoes
    relacionadas as infos gerais do relatorio.
Imagine que a pergunta nao sabe a principio qual relatorio tem a resposta,
    portanto use as infos gerais do relatorio para dar especificidade a que
    relatorio a pergunta esta tratando.
A resposta deve estar contida no trecho, nao nas infos gerais. As infos gerais
    sao apenas de referencia a resposta esta no trecho.
Nao invente informacoes, use apenas dados explicitamente citados no texto.
A pergunta gerada deve ter resposta no trecho.
```

```
A formulacao da pergunta nao pode referenciar o texto diretamente.
A pergunta deve ser altamente tecnica e generica, portanto evite numeros
    especificos.
A resposta deve ser curta, e priorizar respostas de numeros, datas e medidas
    na resposta se possivel.

Formato de saida:

Pergunta 1: [Sua pergunta aqui]
Resposta 1: [Sua resposta aqui]
Pergunta 2: [Sua pergunta aqui]
Resposta 2: [Sua resposta aqui]
Pergunta 3: [Sua pergunta aqui]
Resposta 3: [Sua resposta aqui]
Pergunta 4: [Sua pergunta aqui]
Resposta 4: [Sua resposta aqui]
Pergunta 5: [Sua pergunta aqui]
Resposta 5: [Sua resposta aqui]

Nao diga mais nada alem das perguntas e das respostas. Sem resumos ou analises
    iniciais ou finais.

Infos gerais do relatorio:
{{context}}
Trecho:
{{text}}
"""
```

Results

In Figure 10 we can see some sample results to exemplify the resulting table of Q&A.

Report	Text Element	Question	Answer
Detailed_Relatorio_1.pdf	e exposicdo ao inicio de corrosao. 5. "Testes Mecdnicos e Funcionais* - Sistemas avaliados: Geradores de emergência; bombas hidraulicas e valvulas de controle. - Resultado: Uma bomba apresentou queda de efici€ncia de 10%; e valvulas secundarias mostraram sinais de desgaste. *R esultados e Impactos Identificados* - *Linhas de Ancoragem*: - 25% das linhas requerem substituicdo ou reforco imediato devido a corrosao e fadiga. - *E struturas Metalicas**: - 85% estao em bom estado; enquanto 15% apresentam desgastes significativos. - *R evestimentos Protetores*: - 90% permanecem funcionais; mas 10% necessitam de reaplicacdo urgente. - *S istemas Mecanicos*: - 95% operam dentro dos padrées esperados; mas requerem manutencdo preventiva regular. *KR ecomendacoes Técnicas*	Qual foi a queda de eficiência observada em uma das bombas hidráulicas durante os testes?	10%.
Detailed_Relatorio_1.pdf	1. Substituir imediatamente as 3 linhas de ancoragem com corrosdo avancada. 2. Reaplicar revestimentos anticorrosivos nas areas identificadas com falhas; priorizando zonas de alto impacto salino. 3. Reforcar suportes estruturais em areas criticas para prevenir falhas futuras. 4. Realizar manutencao trimestral em sistemas mecdnicos com sinais de desgaste. 5. Implementar um programa de monitoramento continuo utilizando ferramentas automatizadas para prever falhas. **C onclusao** A Plataforma P-XX continua operacional; mas as acées corretivas identificadas sdo indispensaveis para garantir Sua seguranca e eficiência nos préximos anos. A execucdo das recomendacées apresentadas assegurara a conformidade com os padrées de seguranca e prolongara a vida Util da plataforma.	Quais linhas de ancoragem devem ser substituídas devido a corrosão avançada?	As 3 linhas de ancoragem.
Detailed_Relatorio_1.pdf	1. Substituir imediatamente as 3 linhas de ancoragem com corrosdo avancada. 2. Reaplicar revestimentos anticorrosivos nas areas identificadas com falhas; priorizando zonas de alto impacto salino. 3. Reforcar suportes estruturais em areas criticas para prevenir falhas futuras. 4. Realizar manutencao trimestral em sistemas mecdnicos com sinais de desgaste. 5. Implementar um programa de monitoramento continuo utilizando ferramentas automatizadas para prever falhas. **C onclusao** A Plataforma P-XX continua operacional; mas as acées corretivas identificadas sdo indispensaveis para garantir Sua seguranca e eficiência nos préximos anos. A execucdo das recomendacées apresentadas assegurara a conformidade com os padrées de seguranca e prolongara a vida Util da plataforma.	Que tipo de revestimentos deve ser reaplicado nas áreas com falhas?	Revestimentos anticorrosivos.
Detailed_Relatorio_1.pdf	1. Substituir imediatamente as 3 linhas de ancoragem com corrosdo avancada. 2. Reaplicar revestimentos anticorrosivos nas areas identificadas com falhas; priorizando zonas de alto impacto salino. 3. Reforcar suportes estruturais em areas criticas para prevenir falhas futuras. 4. Realizar manutencao trimestral em sistemas mecdnicos com sinais de desgaste. 5. Implementar um programa de monitoramento continuo utilizando ferramentas automatizadas para prever falhas. **C onclusao** A Plataforma P-XX continua operacional; mas as acées corretivas identificadas sdo indispensaveis para garantir Sua seguranca e eficiência nos préximos anos. A execucdo das recomendacées apresentadas assegurara a conformidade com os padrées de seguranca e prolongara a vida Util da plataforma.	Com que frequência deve ser realizada a manutenção em sistemas mecânicos com sinais de desgaste?	Trimestralmente.
Detailed_Relatorio_9.pdf	Relatério de Inspecao para a Plataforma P-MN *Contexto Geral* A Plataforma P-MN; localizada em uma das areas mais profundas e desafiadoras do litoral; passou por uma inspecao técnica minuciosa. Apdés 9 anos de operacao continua; os desafios enfrentados incluem alta corrosividade do ambiente; impacto de correntes oceanicas extremas e desgastes em componentes estruturais devido ao uso intensivo. **O bjetivos Especificos* 1. Avaliar a eficiência dos reparos estruturais realizados no ultimo ano. 2. Verificar o estado das linhas de ancoragem; considerando a fadiga acumulada. 3. Inspecionar os sistemas de transporte de fluidos e componentes metalicos expostos. 4. Mapear areas criticas que necessitem de intervencao imediata. **Metodologia e Procedimentos Realizados**	Quais são os principais desafios enfrentados pela Plataforma P-MN após um período de operação contínua?	Alta corrosividade do ambiente; impacto de correntes oceanicas extremas e desgastes em componentes estruturais devido ao uso intensivo.

Figure 10 – Example rows from the generated Q&A dataset

6.3 Phase 3: Report Reformulation System

The aim of the project was to create a code that automates the rewriting of texts and tables in documents, making this process usable by members of the Petrobras team. This code analyzes text and table elements while ignoring images. The main goal is to rewrite these elements meaningfully, while preserving the integrity of the information. The code developed is divided into three main steps, described below.

6.3.1 Analyze the Code from Petrobras Teams

The first step in implementing this project involved a detailed analysis of SeSO, the system already developed by Petrobras, to understand its operation and determine how the rewriting code could be integrated. The SeSO code was shared as a Git repository and consists of the following components:

- **SeSO**: Website code
- **SeSO**: Code for training dataset generation
- **SeSO**: Code for the ranker
- **SeSO**: Code for the reader
- **TBE**: Code for handling tables
- **Experimentos Busca Sementica**: Test code

For developing the document rewriting code, an RAG model from the "Experimentos Busca Sementica" project was used as inspiration. The elements reused in the code include:

- The function for processing PDF files in a given folder.
- The function creates a Retriever from the text and table elements.
- The function that provides an answer for a given question using a Retriever.

6.3.2 Access the Virtual Machine and Set Up the Working Environment

The developed code combines several computationally intensive operations, such as extracting large amounts of data, making calls to GPT models that require external APIs, and using a lot of memory for batch processing and generating and manipulating structured PDF documents. For these reasons, Petrobras provided us with virtual machines called TPN.

Since several modules and libraries were needed to run the code, Docker was used.

`Dockerfile`:

- Python 3.11 usage
- **Installation of required libraries**: `unstructured_inference` (for PDFs)
- Working environment configuration: `/usr/src/code`

`docker-compose.yml`: orchestrates container deployment
`requirements.txt`:

- **Document manipulation**: unstructured, pdfminer.six, fpdf2, langchain ;
- **OCR and image extraction**: `unstructured_pytesseract`, `pi_heif`
- **Vector databases and artificial intelligence**: `langchain`, `langchain_chroma`
- **ata model management**: `pydantic`

6.3.3 Writing of the code

The script processes PDF files located in a specific directory, extracts their content (text and tables), and then rewrites this content using a function (`rewrite_texts`) that calls an API (via an API key defined in the script). The modified content is then saved in a new PDF file in an output directory.

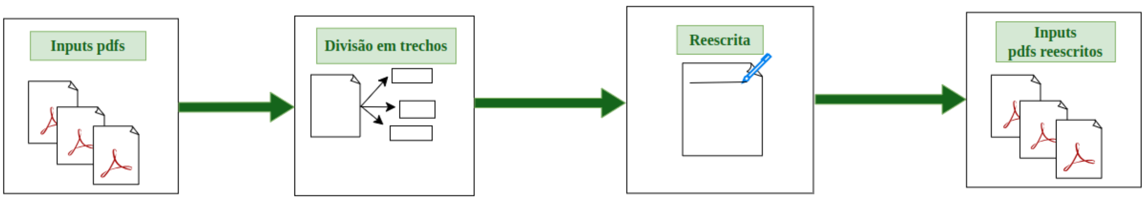

Figure 11 – Global squeme of the rewriting process

Processing of PDFs in input

The function `process_input_pdfs(path: str, file: str) -> tuple[list, list]` takes a pdf as input, merges it into separate elements and returns them as two separate lists.

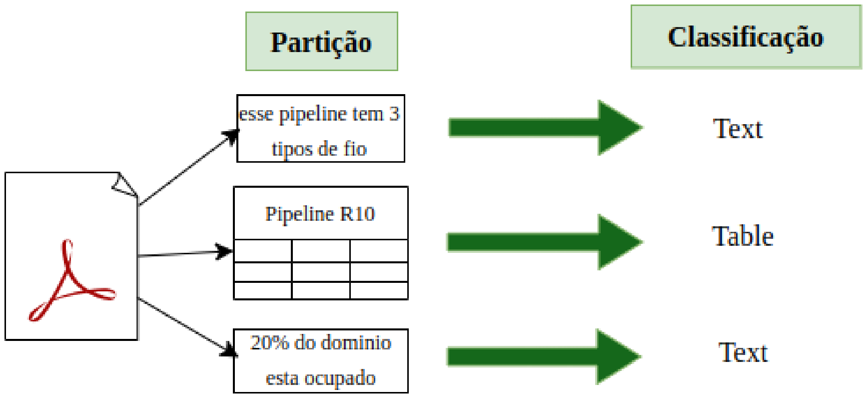

Figure 12 – Steps to process a pdf given in input

Partitioning

The `partition_pdf` function is used to segment the pdf file into structured blocks (text and tables) according to titles, while configuring size limits to group or divide text contents. By dividing documents into smaller and more defined pieces, it becomes easier to process them with natural language processing tools or other AI algorithms because these tools work better on organized data of appropriate size. This also makes it easier to manage the different sections of the document (tables and text) for differentiated treatment.

```
raw_pdf_elements = partition_pdf(
    filename= path + file,
    # Unstructured first finds embedded image blocks
    extract_images_in_pdf=False,
    # Use layout model (YOLOX) to get bounding boxes (for tables) and find
        titles
    # Titles are any sub-section of the document
    infer_table_structure=True,
    # Post processing to aggregate text once we have the title
    chunking_strategy="by_title",
    # Chunking params to aggregate text blocks
    # Attempt to create a new chunk 3800 chars
    # Attempt to keep chunks > 2000 chars
    max_characters=4000,
    new_after_n_chars=3800,
    combine_text_under_n_chars=2000,
    image_output_dir_path=path,
)
```

Classification of extracted elements

This part of the code goes through the elements extracted from the PDF (`raw_pdf_elements`) and classifies them into two categories: "table" type elements and "text" type elements, according to their type, then adds them to a `categorized_elements` list as objects containing the type and associated text.

```
    # Categorize by type
categorized_elements = []
for element in raw_pdf_elements:
    if "unstructured.documents.elements.Table" in str(type(element)):
        categorized_elements.append(Element(type="table",
            text=str(element)))
    elif "unstructured.documents.elements.CompositeElement" in
        str(type(element)):
        categorized_elements.append(Element(type="text", text=str(element)))
```

Code of rewriting

This part of the code reformulates text elements and tables from a document, retaining their content while changing their wording.

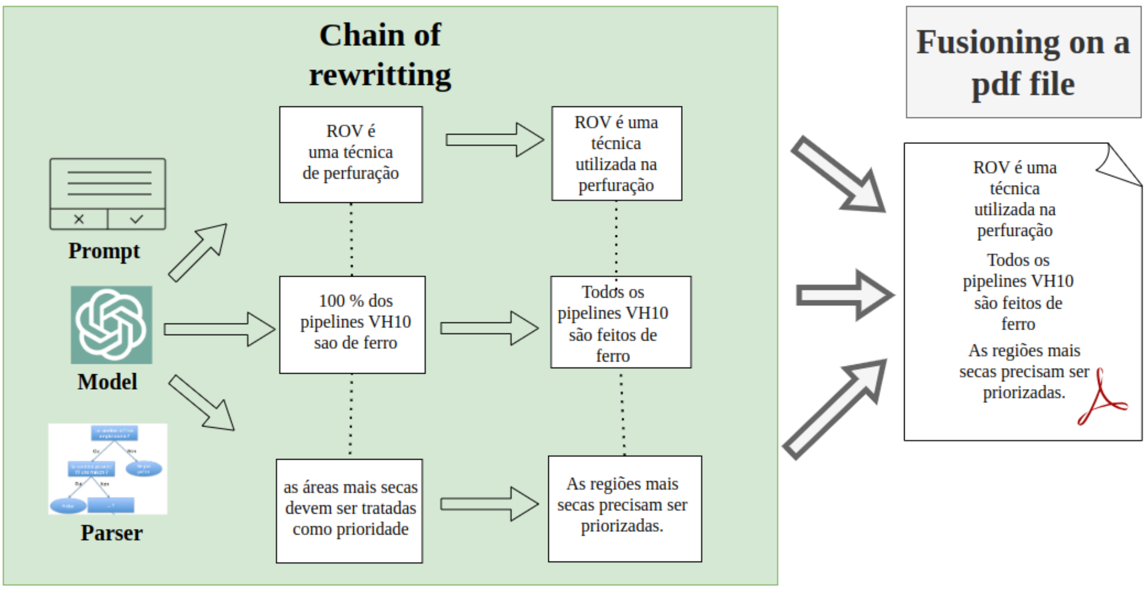

Figure 13 – Step of rewriting after division in pieces

Creation of the prompt

The prompt was to allow for the rewriting of texts in the given input language. The following requirements were followed (CORBASSON, 2023) :

Role	Rewriting Assistant
Context	Rewrite the element provided as input (table or text) to expand a database and allow the LLM using it to provide more context to the data and better understand it.
Format	Format and language similar to the one provided as input. Use of synonyms, utilization of synonyms.
Tone	Same tone as the one used in the input report.
Constraints	Each sentence must be different. Do not leave any identical sentences.

The prompt was made in English to be consistent with the code which is entirely written in English. The following prompt came out:

```
prompt_text = """Voc   um assistente especializado em reformular textos
    com o objetivo de ampliar um banco de dados \
que  ser  utilizado por um modelo de linguagem (LLM). \
Seu papel   reescrever trechos de texto fornecidos, garantindo que:
O texto seja reformatado para oferecer maior contexto e clareza, a
    linguagem seja ajustada para enriquecer \
a compreenso do modelo, garantindo respostas mais relevantes e precisas.
As  informaes  originais devem ser mantidas, sem alterar os dados
    apresentados.
Reformule frases para que nenhuma permanea idntica ao original, alterando
    sua estrutura gramatical, \
ordem das palavras e vocabulrio. Sinnimos apropriados devem ser utilizados
    para evitar repetio e criar diversidade no texto.
Todas as porcentagens presentes no texto devem ser convertidas em fraes
    expressas por palavras.
Exemplos: '20%' deve ser reescrito como 'um quinto'. '50%' deve ser
    alterado para 'metade'. '100%' deve ser substitudo por 'todos'. \
As frases devem ser adaptadas para explorar diferentes estilos de escrita,
    respeitando o tom e o formato originais.
Preserve a estrutura e o tom geral do texto fornecido.
Respeite as  instrues  especficas de estilo ou  organizao , caso estejam
    presentes.\
Fragmento de texto: {element} """
```

Call of the LLM model

This prompt is then given to the Chat GPT API configured upstream. The gpt-4o-mini model was chosen to generate a large amount of data at low cost for this amount and with decent speed and text creation and calculation performance.

The temperature value that controls how the model chooses among its possible options to generate a response was chosen to be 0.7. This ensures that the model uses synonyms while respecting the meaning and structure of the original text and introducing clear adjustments such as converting numbers to letters. **Text Application**

The function applies the summary string to each element of text extracted from documents and produces a reformulated version.

Once the template is configured, it is applied to all text and table elements extracted in the first step.

LangChain's `batch` method allows you to process a set of inputs in parallel, by successively applying a prompt, a model, and a parser to generate results.

In this code:

- The prompt is used to structure or guide the model's responses;
- The model generates the summaries based on the instructions provided by the prompt;
- The parser `StrOutputParser()` formats and extracts the results into readable text.

In practice, this pipeline is applied to data (tables and texts) via `summarize_chain` using batch, multiple elements are summarized simultaneously, with a maximum concurrency limit (`max_concurrency` = 5) to optimize speed while controlling processing load.

```python
# Summary chain
model = ChatOpenAI(model="gpt-4o-mini", temperature=0.7, api_key=API_KEY)
summarize_chain = {"element": lambda x: x} | prompt | model |
    StrOutputParser()

# Apply to tables
tables = [i.text for i in table_elements]
table_summaries = summarize_chain.batch(tables, {"max_concurrency": 5})

# Apply to texts
texts = [i.text for i in text_elements]
text_summaries = summarize_chain.batch(texts, {"max_concurrency": 5})
```

Creation of the pdf

Then, the FPDF library is used to generate an output pdf from the text reformulations obtained in the previous step.The text has been maintained with a similar font type and size to the original report. The cell width is set to 100 (`cell_width` = 100). For each text in the `text_summaries` list, we position the cell horizontally in the center of the page by calculating the margin necessary to center and we add a text cell with several lines, with an alignment centered inside the cell (`align="C"`), a width of 100, and a row height of 10.

```
#create a file pdf
pdf = FPDF()
pdf.add_page()
pdf.set_font("Arial", size=12)
cell_width = 100
for text in text_summaries:
    pdf.set_x((pdf.w - cell_width) / 2)
    pdf.multi_cell(cell_width, 10, text, align="C")
```

Writing the test code

Selection of the Q&A dataset

All questions and answers generated in the previous phase could not be used during the testing phase due to the high cost associated with using the ChatGPT API. We therefore decided to limit the dataset to 66 questions and answers. The imbalance observed in the questions selected for reporting is due to initial testing where running the code with the full set of questions revealed too high a cost. As we will see later, this reduced number of questions nevertheless proved sufficient to draw relevant conclusions. Below is the breakdown of the questions selected for each report.

Report name	Number of questions relating to this report
Detailed_Relatorio_1	35
Detailed_Relatorio_2	2
Detailed_Relatorio_3	2
Detailed_Relatorio_4	2
Detailed_Relatorio_5	2
Detailed_Relatorio_6	2
Detailed_Relatorio_7	2
Detailed_Relatorio_8	2
Detailed_Relatorio_9	15
Detailed_Relatorio_10	2
Total	66

Input and Output of the program

- **input**: .txt file containing the 66 questions ;

```
QUESTION_1 = "Qual foi o foco da inspeo realizada na Plataforma P-XX aps
    15 anos de operao  contnua ?"
QUESTION_2 = "Em que tipo de ambiente a Plataforma P-XX est operando?"
QUESTION_3 = "Quais aspectos da Plataforma P-XX foram avaliados durante a
    inspeo?"
QUESTION_4 = "Quanto tempo a Plataforma P-XX esteve em operao antes de
    ser inspecionada?"
QUESTION_5 = "Quais  condies  severas a Plataforma P-XX enfrenta em suas
    operaes?"
```

- **output**: "Respostas" folder completed with 66 files. 1 file contains the answer and the chunks selected by the Retriever.

```
Pergunta 1: Qual foi o foco da inspeo realizada na Plataforma P-XX aps 15
    anos de operao  contnua ?
Resposta: O foco da  inspeo  realizada na Plataforma P-XX aps 15 anos de
    operao  contnua  foi a  avaliao  abrangente da integridade estrutural,
    funcionalidade dos sistemas de ancoragem e a durabilidade dos
    revestimentos protetores.
\end{itemize}

Trechos selecionados pelo Retriever:

Trecho 1:
Relatorio de Inspecdo para a Plataforma P-XX

*Contexto Geral*

A inspecdo realizada na Plataforma P-XX, apds 15 anos de operacdo
    continua, teve como foco a avaliacdo abrangente
```

Building of the Retriever agent

- **Processing of the pdfs**
- **Creation of the prompt**: a prompt was created to summarize each chunk that the Retriever will receive ;

```
    prompt_text = """You are an assistant tasked with summarizing tables
        and text. \
    Give a concise summary of the table or text. Table or text chunk:
        {element} """
```

- **Definition of the model**: The gpt-3.5-turbo model is chosen and used via a chain to execute the summaries.
- **Applications of the summary string to text and table elements**
- **Vector configuration and storage**: Data is converted into dense vectors through embedding models so that similar elements are spatially close in the vector space. When a query is converted to vector, vectorstore searches for the closest vectors in space. Once a query finds a relevant vector in the `vectorstore`, It is often necessary to access the original document to provide a useful or complete answer, these are present in the `docstore`.

```
# The vectorstore to use to index the child chunks
vectorstore = Chroma(collection_name="summaries",
    embedding_function=OpenAIEmbeddings(api_key=API_KEY))

# The storage layer for the parent documents
store = InMemoryStore()
id_key = "doc_id"

# The retriever (empty to start)
retriever = MultiVectorRetriever(
    vectorstore=vectorstore,
    docstore=store,
    id_key=id_key,
)
```

Selection of the evaluation dataset

To test the performances of the rewriting algorithm, we created three report datasets:

- **dataset 1**: reports selected from those created in step 1 (10 reports)
- **dataset 2**: 10 rewrites of reports from dataset 1 (10 reports)
- **dataset 3**: dataset 1 + dataset 2 (20 reports)

The report dataset was limited to 10 reports to be able to execute the code without generating too much cost. This quantity was considered sufficient to test the performance of the tool created. Several reports were tested and the 10 reports were kept which allow the creativity and calculation capacity of the rewriting code to be tested.

Measuring tools

To measure the Retriever's certainty, we used the accuracy measure. This tool allows you to evaluate the number of correct answers provided by the Retriever, a correct answer adds a weight of 1 and a wrong answer 0:

$$\text{accuracy} = \frac{\text{number of questions with correct answers}}{\text{number of questions asked}}$$

For more precision in the measurements, these tools which have not been used at our level are also interesting and could be explored in a more in-depth analysis of performance:

For more precision in the measurements we also evaluated the MRR. This measurement takes into account the position of the relevant extract in the extracts selected by the Retriever. Thus, a correct answer provided by an extract appearing in the first position has greater weight than one based on an extract appearing in the second position:

$$\text{MRR} = \frac{1}{Q} \sum_{i=1}^{Q} \frac{1}{p_1^i}$$

Q: total number of questions

p1i: position of the correct result for question i

Results obtained

After rewriting, taking into account the 10 reports of dataset 1, we obtain the following statistics: 49% of the words are identical and 51% are different. 74% of the numbers have been written in words or replaced by their fractional equivalent.

After running the code for the three datasets, the following results were obtained:

	Dataset 1	Dataset 2	Dataset 3
Accuracy	0.68	0.61	0.70
MRR	0.58	0.54	0.62

In the tests carried out, 44% of the texts selected by the Retriever were the rewritten versions and 66% of the time the original versions. There is therefore a tendency to take more consideration of the original versions but this remains quite balanced. In 57% of the time, the first extracts selected are the originals and in 43% the rewrites, which shows a certain balance in the treatment of the two versions.

Observations

Several interesting behaviors were noticed during the rewrite.

The ChatGPT model often provides overly detailed responses, extracting excessive information from reports when a concise answer of a few words would suffice. For instance, answers that could be summarized in 5 words sometimes extend to 100. Dataset 3's performance is inconsistent, matching Dataset 2's responses 33% of the time and Dataset 1's 25% of the time, making it unpredictable whether the output will be rewritten or remain unchanged. While Dataset 2 consistently writes out numerical values (90% of cases), Dataset 3 alternates between fully written-out numbers and providing their numerical equivalents in parentheses. Occasionally, Dataset 3 selects the wrong excerpt initially but corrects itself later, or it fails to pick the correct excerpt altogether—cases where Dataset 1 performs better. Additionally, Dataset 3 sometimes introduces errors in numerical rewriting, such as transforming "6" into "one-sixth," resulting in inaccuracies despite correctly identifying the targeted text.

The results remain inherent to the chosen Q&A dataset and the analysis could be deepened by exploring other types of questions. However, this study has already highlighted some interesting behaviors during the rewriting process, revealing which practices to avoid and which to favor to improve performance.

Thus, the results obtained confirm the strong and weak points of Chat GPT reported during a business test carried out by the Harvard Business School and the MIT Sloan School of Management. (QUISQUATER, 2023) Indeed, Chat GPT would be good for innovation and creation but bad for problem solving. The use of Chat GPT in reformulation and in the creation of synthetic data is understandable and the results relevant. However, when it comes to numbers, their transformation does not require creativity but simply calculation. For example, 85% should just be replaced by 17/20 and 100% by "all". A simple replacement calculation algorithm can take care of this task, using regular expressions.

6.4 Phase 4: Text Reduction System

In this phase, we focus on transforming large, complex report passages into concise and contextually relevant summaries tailored to a user's query. Figure 14 illustrates how our system works:

- **Splits** large text passages into smaller, manageable segments, and questions into query splits, like key words of the question.
- **Embeds** these segments and queries into a semantic vector space to capture their underlying meaning.
- **Scores** the segments by comparing their embeddings with those of the queries, identifying the most relevant ones.
- **Filters** out less useful segments to ensure the final returned text is both relevant and succinct.

This layered approach ensures that users receive targeted, context-rich information drawn from extensive and detailed source documents.

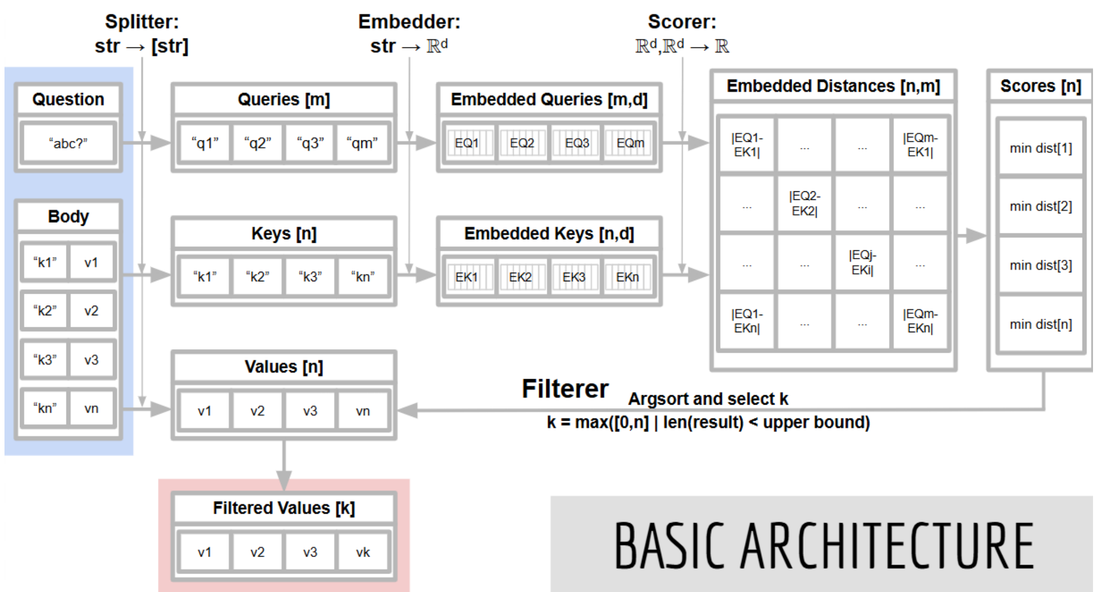

Figure 14 – Diagram explaining the text reduction basic architecture (Figure 14).

6.4.1 Text Splitting (Splitter)

The splitter preprocesses lengthy documents into smaller, structured segments (queries for questions and keys for document sections). By reviewing Figures 15, 16, 17, and 18, we can see the various techniques considered:

- **Non-Overlapping Splits**: The document is divided into equal-length chunks without overlap (Figure 15).
- **Overlapping Splits**: Adjacent segments share content to maintain continuity (Figure 16).
- **Abstract Splitting**: Dynamically identifies logical boundaries (Figure 17).
- **Table Splitting**: Treats each table column as a chunk, with headers as keys (Figure 18).

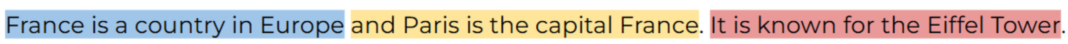

Figure 15 – Splitting a text into non-overlapping chunks.

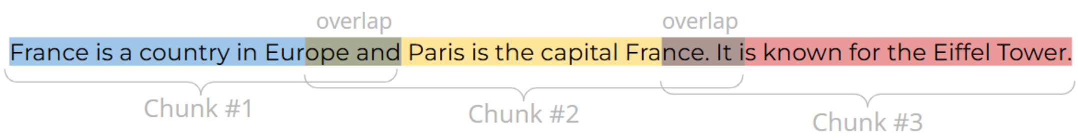

Figure 16 – Splitting a text into overlapping chunks.

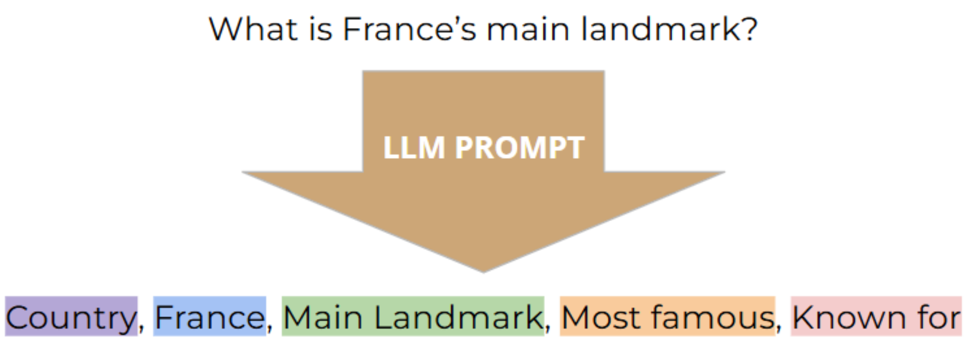

Figure 17 – Splitting a text into abstract key chunks.

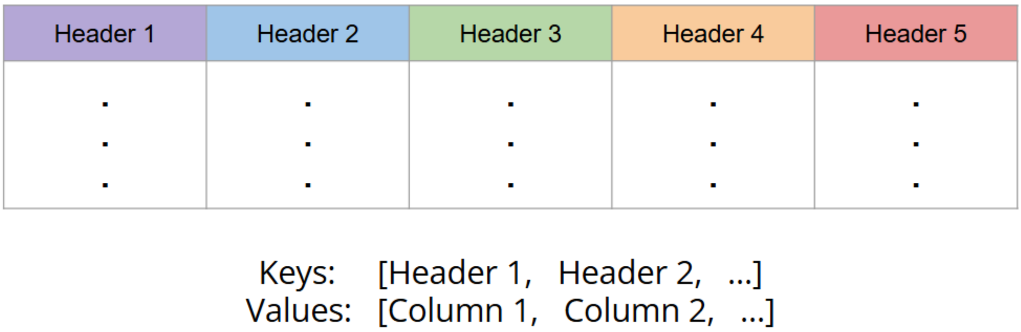

Figure 18 – Splitting a table into columns.

Our Approaches

- Abstract splitting for queries.
- Overlapping splits for lengthy text passages for the keys, and no overlap for the values.
- Table splitting for tabular data. The header name is the key, the whole column is the value.

Pseudocode for Splitting

```
FUNCTION SplitText(text, max_size, overlap_ratio):
    words ← Split text into words
    non_overlap_chunks ← partition words into size max_size
    overlap_chunks ← []

    FOR each chunk in non_overlap_chunks (except first and last):
        overlap_chunks ← combine partial segments
                         from previous and next chunks
                         to retain context

    RETURN non_overlap_chunks, overlap_chunks
```

6.4.2 Embedding Creation (Embedder)

Once we have queries and segments, we convert them into numerical embeddings. These embeddings represent the semantic meaning of the text, enabling effective similarity comparisons.

- Both segments and questions are formatted before embedding (e.g., "query: segment" for queries).

- We use pre-trained language models (e.g., OpenAI embeddings (INTRODUCING..., 2022)) to convert the text into high-dimensional vectors.

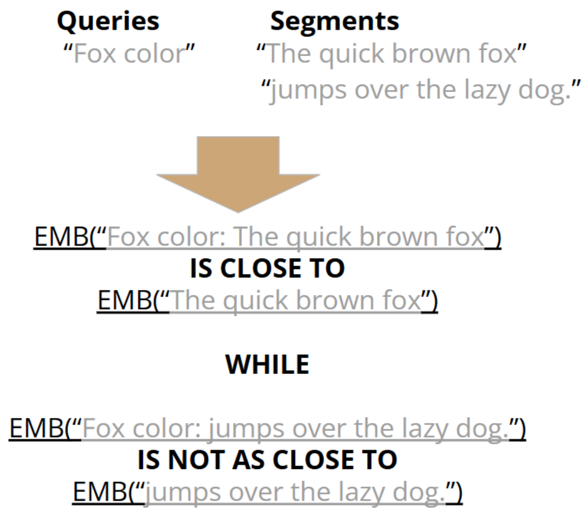

Figure 19 – Embedding comparing example (Figure 19).

Pseudocode for Embedding Preparation

```
FUNCTION GetEndpoints(queries, keys, formats):
    FOR each query in queries:
        FOR each key in keys:
            FOR each format in formats:
                (formatted_query, formatted_key) = format(query, key)
                store endpoints and endqueries
    RETURN endpoints, endqueries

FUNCTION EmbedText(text_list):
    RETURN embeddings from model
```

6.4.3 Similarity Scoring (Scorer)

We measure relevance by comparing query embeddings with segment embeddings. We use 1 - cosine similarity since it is common and effective in semantic search contexts.

Pseudocode for Scoring

```
FUNCTION ScoreKeys(queries, keys, formats, Embed, ScoreFunction):
    endpoints, endqueries ← GetEndpoints(queries, keys, formats)
    endpoints_emb ← Embed(endpoints)
    endqueries_emb ← Embed(endqueries)

    scores ← compute raw similarity (dot product)
    scores ← ScoreFunction(scores)  # e.g., 1 - cosine similarity

    best_scores_per_key ← find best match across queries and formats
    RETURN best_scores_per_key
```

6.4.4 Filtering (Filterer)

After scoring, we filter the segments to return only the most relevant ones. We do this by setting a maximum token size threshold and including only the top segments until we reach that limit.

Pseudocode for Filtering

```
FUNCTION FilterSegments(keys_scores, original_text, max_ratio):
    max_tokens = CountWords(original_text) * max_ratio
    ordered = Sort keys_scores by score

    FOR each threshold in ordered:
        candidate_text = segments with score <= threshold
        IF word_count(candidate_text) > max_tokens:
            candidate_text = segments with score < threshold
            BREAK
    RETURN candidate_text
```

6.4.5 Example Workflow

1. **Input**: User query: "A linha de ancoragem #10 da plataforma já apresentou defeitos?"
2. **Splitter**: Splits a passage into overlapping segments:
   - Segment A: "A linha #10 apresentou falhas de corrosão,..."
   - Segment B: "...porém a linha #1 não demonstrou nenhum tipo de defeito."
3. **Embedder**: Generates embeddings for each segment and the query.
4. **Scorer**: Computes similarity scores and retrieves Segment A, ranking it highest for its semantic relevance.

5. **Filterer**: Removes less useful segments, in this case B, reducing the text size.

6.4.6 Integration into Summarization

This summarization function brings all steps together, using only pseudocode at a high level.

Pseudocode for Summarization

```
FUNCTION SummarizeText(embedder, question, formats, text_extract):
    queries ← SplitText(question, small_max_size, no_overlap)
    keys, keys_overlap ← SplitText(text_extract, larger_max_size, some_overlap)

    keys_scores ← ScoreKeys(queries, keys_overlap, formats, embedder, CosineSim)
    summarized ← FilterSegments(keys_scores, text_extract, max_ratio=0.5)

    RETURN summarized
```

6.4.7 Evaluation

To assess the effectiveness of the summarization approach in preserving essential information for accurate question-answering, we conducted three distinct experiments using the previously generated Q&A dataset. Each experiment varied in the method of scoring and selecting text segments for summarization. The following subsections outline the experimental conditions and the corresponding evaluation metrics.

Experimental Conditions

1. **Random Scoring (Control):** In this baseline experiment, each text segment key was assigned a random score between 0 and 1. This random assignment simulates an uninformed selection process, reducing the text length by approximately 50%. The purpose of this control is to establish a performance benchmark against which more sophisticated methods can be compared.

2. **Direct Embedding (No Formatting):** This experiment involved scoring text segments based on the cosine similarity between the embeddings of the question and each key segment. Specifically, we computed `embed({query})` and `embed({key})` for each pair, without any additional formatting. This approach leverages semantic similarity to identify and retain the most relevant segments related to the question.

3. **Query-Enhanced Embedding (With Formatting):** Building upon the second experiment, this approach introduced a string formatting step. For each key segment, we compared `embed("query: key")` with `embed({key})`. The inclusion of the query

within the embedding input aims to provide contextual information, potentially enhancing the relevance of the selected segments by explicitly linking them to the question.

The pseudocode for the main evaluation routine is presented below:

```
PSEUDOCODE: MAIN EVALUATION ROUTINE

load Q&A dataset from file
initialize semantic embedder
initialize output structure

for each (text_segment, question) in dataset:
    summarized_segment = summarize(text_segment, question)

    answer_full = generate_answer(text_segment, question)
    answer_summary = generate_answer(summarized_segment, question)

    vec_full = embed(answer_full)
    vec_summary = embed(answer_summary)
    similarity = cosine_similarity(vec_full, vec_summary)

    store (
        original_text,
        question,
        summarized_text,
        answer_full,
        answer_summary,
        similarity
    )
```

Figure 20 – Pseudocode outlining the main evaluation routine for comparing answers derived from full and summarized texts.

Results

Threshold Justification

Table1

text	question	summarized_text	answer from full text	answer from summarized text	similarity
Relatório de Inspecdo para a Plataforma P-XX *Contexto Geral* A inspecdo realizada na Plataforma P-XX; apds 15 anos de operacdo continua; teve como foco a avaliacdo abrangente avaliacdo abrangente da integridade estrutural; funcionalidade dos sistemas de ancoragem e a durabilidade dos revestimentos protetores. Localizada em aguas ultraprofundas e operando sob condicées severas de alta pressdo; salinidade ciclos térmicos	Qual o foco principal da atividade de avaliação realizada na plataforma após um período prolongado de operação?	de operacdo continua; teve como foco a avaliacdo abrangente avaliacdo abrangente da integridade estrutural; funcionalidade dos sistemas de ancoragem e a durabilidade dos revestimentos	O foco principal da atividade de avaliação realizada na plataforma após um período prolongado de operação foi a avaliação abrangente da integridade estrutural; funcionalidade dos sistemas de ancoragem e a durabilidade dos revestimentos protetores.	O foco principal da atividade de avaliação realizada na plataforma após um período prolongado de operação foi a avaliação abrangente da integridade estrutural; funcionalidade dos sistemas de ancoragem e a durabilidade dos revestimentos.	0.99795
Relatório de Inspecdo para a Plataforma P-XX *Contexto Geral* A inspecdo realizada na Plataforma P-XX; apds 15 anos de operacdo continua; teve como foco a avaliacdo abrangente avaliacdo abrangente da integridade estrutural; funcionalidade dos sistemas de ancoragem e a durabilidade dos revestimentos protetores. Localizada em aguas ultraprofundas e operando sob condicées severas de alta pressdo; salinidade ciclos térmicos	Quantos anos de operação contínua teve a plataforma antes da inspeção?	de operacdo continua; teve como foco a avaliacdo abrangente avaliacdo abrangente da integridade estrutural; funcionalidade dos sistemas de ancoragem e a durabilidade dos revestimentos	A plataforma teve 15 anos de operação contínua antes da inspeção.	The text does not provide information on how many years of continuous operation the platform had before the inspection.	0.84267
Relatório de Inspecdo para a Plataforma P-XX *Contexto Geral* A inspecdo realizada na Plataforma P-XX; apds 15 anos de operacdo continua; teve como foco a avaliacdo abrangente avaliacdo abrangente da integridade estrutural; funcionalidade dos sistemas de ancoragem e a durabilidade dos revestimentos protetores. Localizada em aguas ultraprofundas e operando sob condicées severas de alta pressdo; salinidade ciclos térmicos	Em que tipo de ambiente a plataforma está localizada?	P-XX *Contexto Geral* A inspecdo realizada a avaliacdo abrangente avaliacdo abrangente da ancoragem e a durabilidade dos revestimentos operando sob condicées severas de alta	A plataforma está localizada em águas ultraprofundas e opera sob condições severas de alta pressão; salinidade e ciclos térmicos.	The text does not provide information on the type of environment the platform is located in.	0.76729
e extremos; a plataforma desempenha um papel essencial na producao de petrdleo em campos de dificil acesso. **O bjetivos Especificos* 1. Avaliar o estado das linhas de ancoragem para determinar sua capacidade de carga e resistencia a fadiga. 2. Identificar sinais de corrosdo e danos estruturais em superficies metalicas internas e externas. 3. Verificar a aderência e eficacia dos revestimentos anticorrosivos em ambientes de alta salinidade. 4. Propor acées corretivas para prolongar a vida Util dos sistemas criticos e prevenir riscos operacionais. 5. Garantir que a plataforma esteja em conformidade com as regulamentacdes de seguranca operacional. **Metodologia e Procedimentos Realizados**	Qual é o foco principal da avaliação realizada na Plataforma P-XX após um longo período de operação?	papel essencial na producao de petrdleo em campos de dificil acesso. **O das linhas de ancoragem para determinar sua capacidade de carga e resistencia corrosdo e danos estruturais em superficies metalicas internas e externas. 3. Verificar que a plataforma esteja em conformidade **Metodologia e Procedimentos Realizados**	O foco principal da avaliação realizada na Plataforma P-XX após um longo período de operação é avaliar o estado das linhas de ancoragem; identificar sinais de corrosão e danos estruturais; verificar a eficácia dos revestimentos anticorrosivos; propor ações corretivas para prolongar a vida útil dos sistemas críticos e garantir que a plataforma esteja em conformidade com as regulamentações de segurança operacional.	O foco principal da avaliação realizada na Plataforma P-XX após um longo período de operação é determinar a capacidade de carga e resistência à corrosão e danos estruturais nas linhas de ancoragem e em superfícies metálicas internas e externas.	0.96827

Figure 21 – Sample of the generated CSV results, including text, question, summarized text, answers, and similarity scores.

The similarity threshold of 0.9 was chosen based on manual inspection of the results presented in Figure 21. It was observed that pairs with similarity scores below 0.9 generally exhibited significant discrepancies in meaning between the answers derived from the full and summarized texts. While the threshold is not absolute and some exceptions exist, it serves as a practical benchmark to differentiate between effective and ineffective summarizations. This threshold allows for a clear and consistent measure to evaluate the retention of crucial information necessary for accurate question-answering.

Experiment 1: Random Scoring (Control)

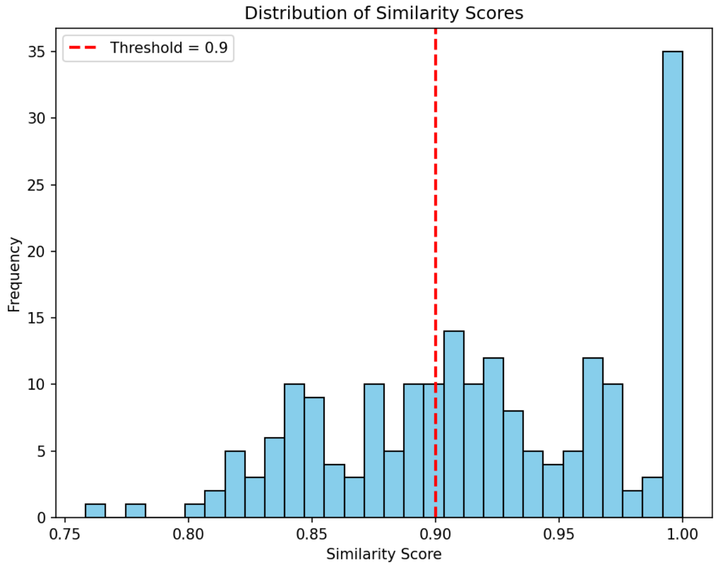

Figure 22 – Similarity score distribution for Random Scoring. The 0.9 threshold is indicated.

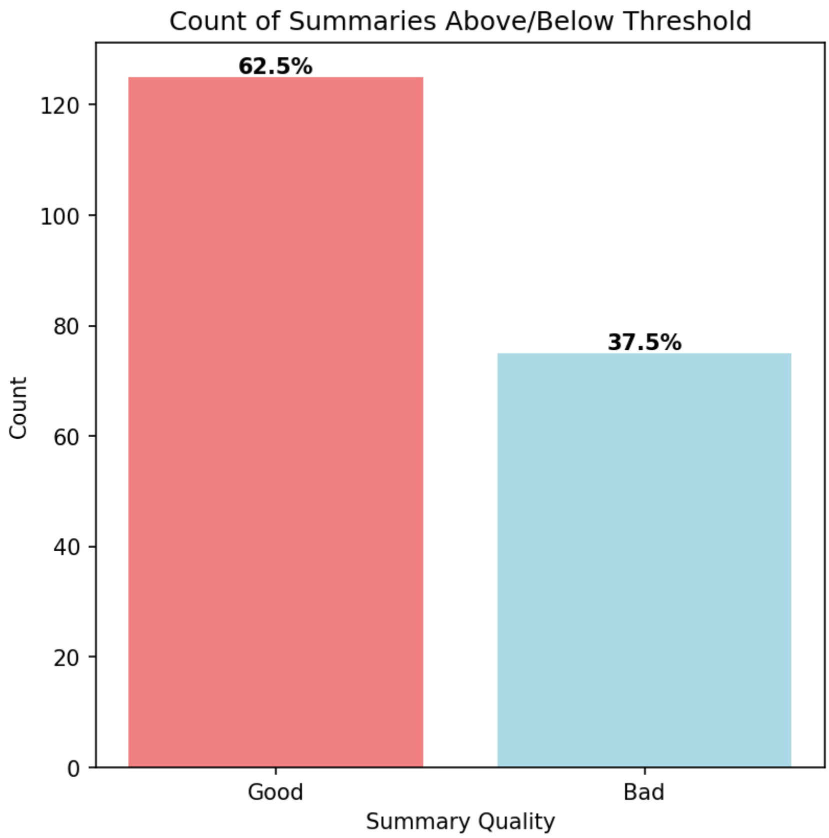

Figure 23 – Count of Q&A pairs above and below the 0.9 threshold for Random Scoring.

Experiment 2: Direct Embedding (No Formatting)

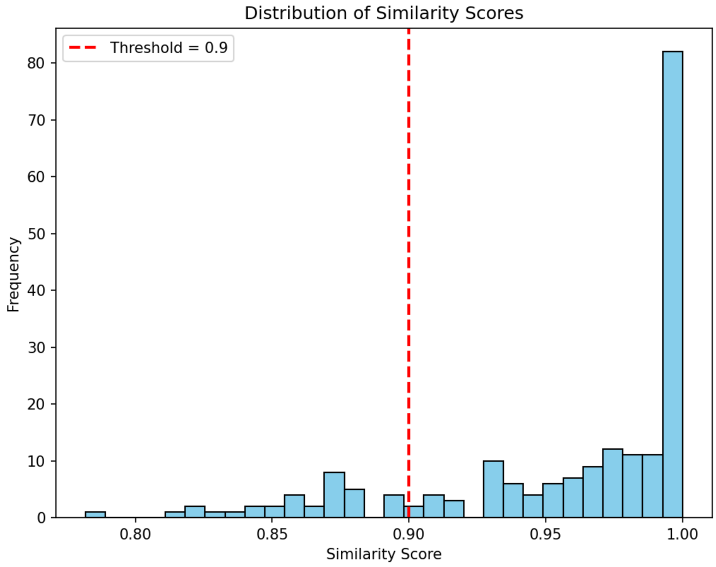

Figure 24 – Similarity score distribution for Direct Embedding (No Formatting). The 0.9 threshold is shown.

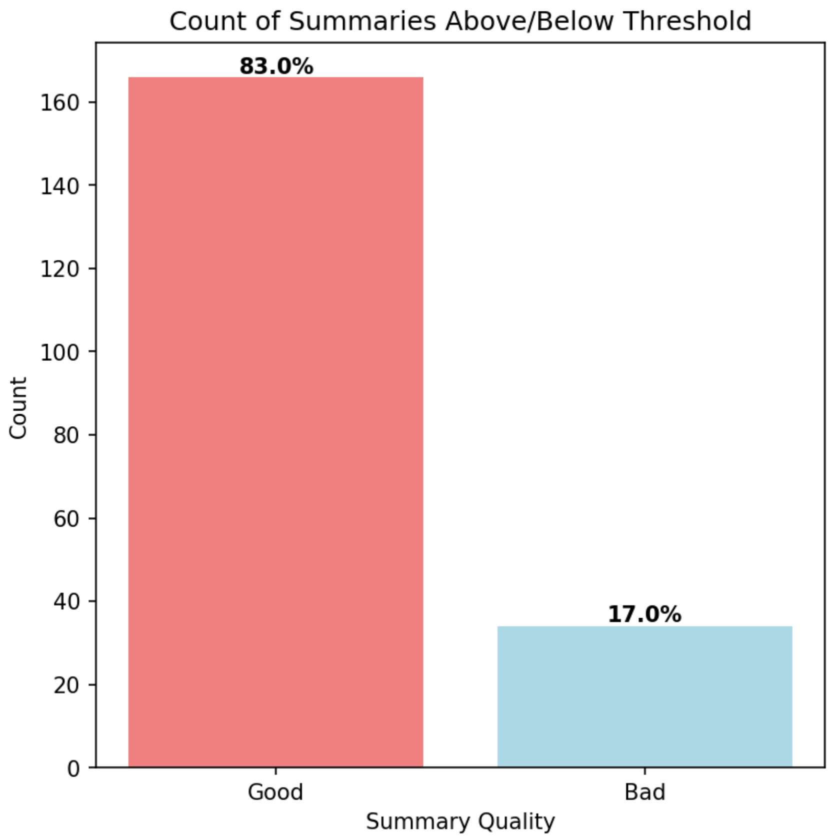

Figure 25 – Count of Q&A pairs above and below the 0.9 threshold for Direct Embedding.

Experiment 3: Query-Enhanced Embedding (With Formatting)

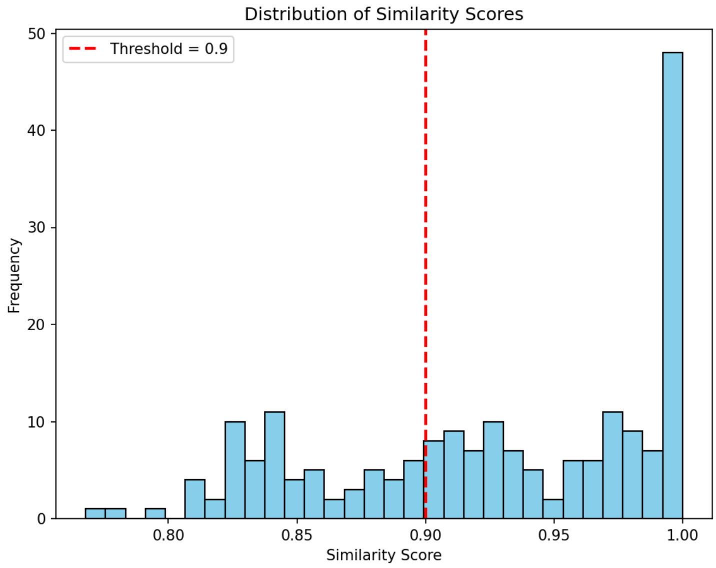

Figure 26 – Similarity score distribution for Query-Enhanced Embedding. The 0.9 threshold is marked.

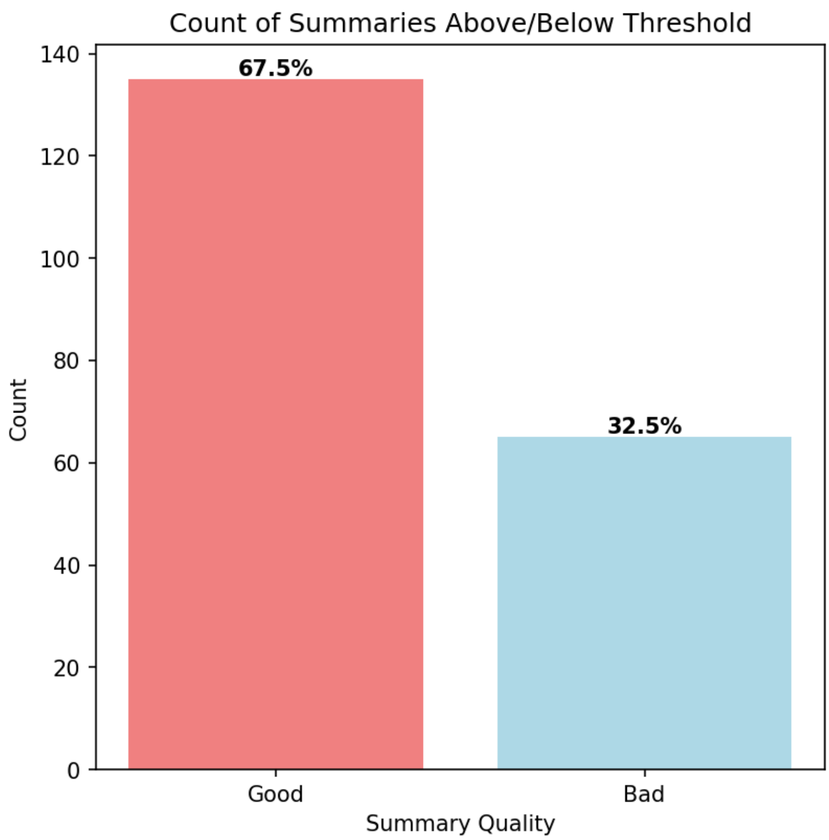

Figure 27 – Count of Q&A pairs above and below the 0.9 threshold for Query-Enhanced Embedding.

The results from the three experiments are summarized as follows:

- **Random Scoring (Control):** 62.5% of the Q&A pairs achieved similarity scores above the 0.9 threshold, while 37.5% fell below. This indicates that random segment selection slightly favors "good" summarizations, albeit without any informed selection criteria.
- **Direct Embedding (No Formatting):** A significant improvement is observed, with 83% of the Q&A pairs surpassing the 0.9 similarity threshold and only 17% falling below. This demonstrates that leveraging semantic embeddings for scoring substantially enhances the quality of summarization compared to random selection.
- **Query-Enhanced Embedding (With Formatting):** In this experiment, 67.5% of the Q&A pairs exceeded the 0.9 threshold, while 32.5% did not. Although this approach outperforms the random baseline, it did not achieve the expected improvement over the Direct Embedding method.

Statistical Analysis

The objective of the statistical analysis is to determine whether the improvements observed in the Direct Embedding (No Formatting) and Query-Enhanced Embedding (With Formatting) experiments are statistically significant compared to the Random Scoring (Control) experiment. To achieve this, we employed the chi-squared test, a suitable method for comparing categorical data to assess whether there is a significant association between the type of summarization method and the quality of the summarization outcome.

The chi-squared test (NEWBOLD; CARLSON; THORNE, 2013) is appropriate for this analysis because:

- **Categorical Variables:** Both the summarization method (Random Scoring, Direct Embedding, Query-Enhanced Embedding) and the outcome category (Good, Bad) are categorical.
- **Independence:** Each Q&A pair is an independent observation.
- **Sample Size:** With a sample size of 200 per experiment, the expected frequencies in each cell of the contingency tables are sufficient to satisfy the chi-squared test assumptions.

Alternative methods, such as Fisher's Exact Test (FISHER, 1934) or logistic regression (HOSMER; LEMESHOW; STURDIVANT, 2013), could be considered. However, Fisher's Exact Test is more suitable for smaller sample sizes, and logistic regression is more complex and generally used when modeling the probability of an outcome based on multiple predictors. Given the simplicity and adequacy of the chi-squared test for our data, it was the most appropriate choice.

Chi-Squared Test for Direct Embedding vs. Random Scoring

Hypotheses:

- **Null Hypothesis (H_0):** There is no association between the summarization method (Direct Embedding vs. Random Scoring) and the summarization quality. In other words, the proportion of "Good" summarizations is the same for both methods.
- **Alternative Hypothesis (H_1):** There is an association between the summarization method and the summarization quality. Specifically, the Direct Embedding method results in a higher proportion of "Good" summarizations compared to Random Scoring.

Observed Frequencies:

	Good	Bad	Total
Direct Embedding	166	34	200
Random Scoring	125	75	200
Total	291	109	400

Expected Frequencies:

$$E_{\text{Direct, Good}} = \frac{291}{400} \times 200 = 145.5$$
$$E_{\text{Direct, Bad}} = \frac{109}{400} \times 200 = 54.5$$
$$E_{\text{Random, Good}} = 145.5$$
$$E_{\text{Random, Bad}} = 54.5$$

Chi-Squared Calculation:

$$\chi^2 = \sum \frac{(O-E)^2}{E} = \frac{(166-145.5)^2}{145.5} + \frac{(34-54.5)^2}{54.5} + \frac{(125-145.5)^2}{145.5} + \frac{(75-54.5)^2}{54.5}$$

$$\chi^2 = \frac{(20.5)^2}{145.5} + \frac{(-20.5)^2}{54.5} + \frac{(-20.5)^2}{145.5} + \frac{(20.5)^2}{54.5}$$
$$\chi^2 = \frac{420.25}{145.5} + \frac{420.25}{54.5} + \frac{420.25}{145.5} + \frac{420.25}{54.5}$$
$$\chi^2 = 2.89 + 7.72 + 2.89 + 7.72 = 21.22$$

Degrees of Freedom:

$$df = (Rows - 1) \times (Columns - 1) = (2-1) \times (2-1) = 1$$

Conclusion: With $\chi^2 = 21.22$ and $p < 0.001$, we reject the null hypothesis at the 95% confidence level. This indicates that the Direct Embedding approach significantly outperforms Random Scoring in achieving "Good" summarizations.

Chi-Squared Test for Query-Enhanced Embedding vs. Random Scoring
Hypotheses:

- **Null Hypothesis (H_0):** There is no association between the summarization method (Query-Enhanced Embedding vs. Random Scoring) and the summarization quality. In other words, the proportion of "Good" summarizations is the same for both methods.

- **Alternative Hypothesis (H_1):** There is an association between the summarization method and the summarization quality. Specifically, the Query-Enhanced Embedding method results in a higher proportion of "Good" summarizations compared to Random Scoring.

Observed Frequencies:

	Good	Bad	Total
Query-Enhanced Embedding	135	65	200
Random Scoring	125	75	200
Total	260	140	400

Expected Frequencies:

$$E_{\text{Query, Good}} = \frac{260}{400} \times 200 = 130$$

$$E_{\text{Query, Bad}} = \frac{140}{400} \times 200 = 70$$

$$E_{\text{Random, Good}} = 130$$

$$E_{\text{Random, Bad}} = 70$$

Chi-Squared Calculation:

$$\chi^2 = \sum \frac{(O-E)^2}{E} = \frac{(135-130)^2}{130} + \frac{(65-70)^2}{70} + \frac{(125-130)^2}{130} + \frac{(75-70)^2}{70}$$

$$\chi^2 = \frac{(5)^2}{130} + \frac{(-5)^2}{70} + \frac{(-5)^2}{130} + \frac{(5)^2}{70}$$

$$\chi^2 = \frac{25}{130} + \frac{25}{70} + \frac{25}{130} + \frac{25}{70}$$

$$\chi^2 = 0.192 + 0.357 + 0.192 + 0.357 = 1.098$$

Degrees of Freedom:

$$df = (Rows - 1) \times (Columns - 1) = (2-1) \times (2-1) = 1$$

Conclusion: With $\chi^2 = 1.098$ and $p = 0.459$, we fail to reject the null hypothesis at the 95% confidence level. This indicates that the Query-Enhanced Embedding approach does not significantly differ from Random Scoring in achieving "Good" summarizations.

Summary of Findings

- **Direct Embedding (No Formatting) vs. Random Scoring:**
  - **Chi-Squared Statistic:** $\chi^2 = 21.22$
  - **P-Value:** $p < 0.001$
  - **Conclusion:** The Direct Embedding approach significantly outperforms Random Scoring in achieving "Good" summarizations at the 95% confidence level.
- **Query-Enhanced Embedding (With Formatting) vs. Random Scoring:**
  - **Chi-Squared Statistic:** $\chi^2 = 1.098$
  - **P-Value:** $p = 0.459$
  - **Conclusion:** The Query-Enhanced Embedding approach does not significantly differ from Random Scoring in achieving "Good" summarizations at the 95% confidence level.

These results confirm that while the Direct Embedding method provides a statistically significant improvement over the Random Scoring control, the Query-Enhanced Embedding method does not offer a meaningful enhancement in summarization quality compared to random selection.

Implications

Based on the control experiment, the Query-Enhanced Embedding was not as effective as anticipated, achieving only a modest improvement over random scoring (67.5% vs. 62.5%). In contrast, the Direct Embedding approach demonstrated a substantial enhancement in summarization quality, increasing the proportion of "good" summarizations from 62.5% to 83%. Although not perfect, this represents a significant improvement over the baseline method of simply cropping half of the text without any scoring.

It is interesting to see how our initial proposal of adding more context to the embedding by formatting the string beforehand was not only insufficient, but detrimental. The simpler non formatting case is better.

6.4.8 Conclusion

The evaluation of the summarization approach using the Q&A dataset revealed that embedding-based scoring methods, particularly the Direct Embedding approach, significantly improve the preservation of essential information necessary for accurate question-answering. While the Query-Enhanced Embedding did not perform as expected, the Direct Embedding method achieved an 83% success rate in maintaining high similarity between answers derived from full and summarized texts. This performance marks a considerable advancement over the baseline random selection method, which achieved a 62.5% success rate.

This substantial improvement highlights the effectiveness of semantic embedding-based scoring in enhancing summarization quality compared to both random selection and query-enhanced formatting. Future work should focus on refining embedding techniques and exploring more sophisticated formatting strategies to further enhance summarization quality, potentially increasing the proportion of "good" summarizations and approaching near-perfect information retention.

7 Final Remarks

7.1 General Conclusion

This project aimed to address two primary challenges in Petrobras's Semantic Search on Offshore Engineering (SeSO) system: the scarcity of real-world failure reports and the retrieval limitations imposed by large document passages. By generating synthetic failure reports, augmenting these datasets through rewriting techniques, and implementing summarization methods, significant progress was made in improving the system's ability to retrieve and answer questions effectively.

7.1.1 Results and Achievements

- **Synthetic Report Generation:** - A dataset of realistic synthetic reports was generated, effectively mitigating the lack of real-world failure reports. These reports maintained structural and contextual fidelity to actual operational documents. - The synthetic dataset allowed for an expanded training and evaluation environment, contributing to the refinement of SeSO's retrieval capabilities.

- **Augmented Report Generation:** - Rewriting techniques produced a 51% variation in word choices and a 74% accuracy in transforming numerical values into written fractions or equivalents. - Augmented reports introduced significant linguistic and contextual diversity, enriching the dataset.

- **QA Dataset Generation:** - A dataset of 66 question-answer pairs was generated, providing a baseline for evaluating the retrieval system's performance. Accuracy for retrieval ranged from 0.61 to 0.70 across different datasets.

- **Passage Summarization:** - Summarization methods achieved a high cosine similarity score of 0.9 or above for 83% of question-answer pairs using the Direct Embedding method. - Direct Embedding significantly outperformed both Random Scoring (62.5% success rate) and Query-Enhanced Embedding (67.5%), demonstrating its effectiveness in retaining critical information while reducing passage length.

- **Retrieval Performance:** - The retrieval system exhibited balanced treatment of original and rewritten datasets, with rewritten versions selected 43% of the time as the most relevant segment. - Statistical analysis confirmed the superiority of Direct Embedding, with a chi-squared value of 21.22 and $p < 0.001$, indicating significant improvement over the baseline.

7.1.2 Flaws and Challenges in the Methodology

Despite these achievements, several flaws and challenges emerged:

- **Numerical Rewriting Errors:** - While 74% of numerical values were correctly reformatted, occasional errors (e.g., rewriting "6" as "one-sixth") revealed the limitations of the LLM's numerical handling capabilities. These inaccuracies highlighted the need for deterministic algorithms for numerical transformations.
- **Over/Under-detailed Questions and Answers:** - Questions and Answers generated by the system were often excessively detailed or not specific enough compared to actual Petrobras employees expected Questions and Answers. This behavior is not perfectly reflective of the real world.
- **Query-Enhanced Embedding Underperformance:** - Contrary to expectations, Query-Enhanced Embedding did not outperform the simpler Direct Embedding approach. This indicates that the additional query formatting introduced noise rather than enhancing relevance.
- **High Costs of API Usage:** - The extensive reliance on external APIs for generating synthetic reports and running summarization tasks proved costly, limiting the scale of experimentation, particularly for larger datasets.
- **Dataset Imbalance:** - The Q&A dataset's limited size (66 pairs) and uneven question distribution across reports (e.g., 35 questions for one report vs. 2 for others) may have skewed the evaluation results.
- **Token Limit Constraints:** - The summarization system occasionally faced challenges in handling very large passages, requiring further optimization for real-world deployment.

7.2 Future Work

Building on the insights and limitations of this project, several directions for future work are proposed:

- **Improved Numerical Rewriting:** - Implement deterministic algorithms for numerical transformations to ensure 100% accuracy in converting numbers to fractions or written equivalents.
- **Scaling Q&A Dataset:** - Expand the Q&A dataset with more diverse and evenly distributed questions across all reports, enabling more robust testing and evaluation of retrieval and summarization methods.

- **Advanced Summarization Techniques:** - Explore more sophisticated summarization approaches, such as hierarchical models or hybrid methods combining semantic embeddings with rule-based techniques.
- **Cost Optimization:** - Investigate cost-efficient alternatives to external APIs, including open-source LLMs and on-premise deployment of summarization and rewriting tools.
- **Dynamic Passage Reduction:** - Develop adaptive summarization methods that dynamically adjust passage length based on the complexity of the user query and the document context.
- **Integration with SeSO:** - Conduct a pilot integration of the summarization and rewriting modules into SeSO to evaluate real-world performance and identify further optimization opportunities.
- **Broader Statistical Validation:** - Perform additional statistical analyses, such as logistic regression, to better understand the factors influencing retrieval success and summarization quality.
- **User Feedback and Usability Testing:** - Incorporate feedback from Petrobras employees to refine the system's interface, answer generation, and summarization outputs for improved user experience.
- **Embedding Optimization:** - Experiment with fine-tuned embedding models tailored to Petrobras's domain to enhance retrieval accuracy and summarization relevance.
- **Error Correction Mechanisms:** - Implement post-processing checks to correct errors in generated summaries, numerical transformations, and retrieved passages before presenting them to users.

By addressing these areas, the project's methodologies can be further refined, scaled, and integrated into Petrobras's operations. These advancements will enable SeSO to provide more reliable, efficient, and accurate question-and-answer capabilities, ensuring its long-term value in supporting offshore engineering challenges.

Bibliography

AWS. Qu'est-ce que la rag (génération augmentée de récupération) ? *AWS*, 2020. 8, 16

CHATGPT: Optimizing Language Models for Dialogue. 2022. <https://openai.com/blog/chatgpt/>. Accessed: 2024-11-25. 15

COHESITY. La génération augmentée de récupération (rag). *Cohesity*, 2021. 16

CORBASSON, A. 6 ingrédients pour faire un (bon) prompt chatgpt. *Digitad*, 2023. 49

ELASTICSEARCH: The Definitive Guide. [S.l.]: O'Reilly Media, 2015. <https://www.oreilly.com/library/view/elasticsearch-the-definitive/9781449358532/>. Accessed: 2024-11-25. 12

FISHER, R. A. *Statistical Methods for Research Workers*. [S.l.]: Oliver and Boyd, 1934. 67

GONCALVES, P. et al. Semantic search in offshore engineering with linguistics and neural processing pipelines. In: AMERICAN SOCIETY OF MECHANICAL ENGINEERS (ASME). *Proceedings of the ASME 2021 40th International Conference on Ocean, Offshore and Arctic Engineering (OMAE2021)*. Houston, Texas, USA, 2021. p. 9. Available from Internet: <https://bv.fapesp.br/en/publicacao/228263/>. 8

HOSMER, D. W.; LEMESHOW, S.; STURDIVANT, R. X. *Applied Logistic Regression*. 3rd. ed. [S.l.]: Wiley, 2013. 67

INTRODUCING OpenAI Embeddings. 2022. <https://openai.com/blog/introducing-text-and-code-embeddings/>. Accessed: 2024-11-25. 59

LANGCHAIN: Building Applications with Large Language Models. 2022. <https://github.com/hwchase17/langchain>. Accessed: 2024-11-25. 16

LANGUAGE Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, v. 33, 2020. Accessed: 2024-11-25. 15

LIU, Y.; LAPATA, M. Text summarization with pretrained encoders. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019. p. 3730–3740. Available from Internet: <https://arxiv.org/abs/1908.08345>. 24

MICHARD, T. Comparatif des meilleurs chatbots ia grand public en 2024. *Reglo.ia*, 2024. 15

NEWBOLD, P.; CARLSON, W.; THORNE, B. *Statistics for Business and Economics*. 8th. ed. [S.l.]: Pearson, 2013. 67

PETROBRAS Official Website. <https://petrobras.com.br/>. Accessed: 2024-11-25. 8

QUISQUATER, B. C. e. C. C. J.-J. Chatgpt, bon pour créer, mauvais pour solutionner. *La Tribune*, 2023. 55