AI-Powered Sustainability Data Extraction from Annual Reports: A Technical Report

Atef BOUZID & Hamdi BARKOUS

March 24, 2024

Abstract

This technical report presents the development and evaluation of an AI-powered solution for extracting sustainability data from annual reports. The solution addresses the challenges associated with structured & unstructured data and diverse document formats by employing a combination of natural language processing techniques, document parsing tools, and large language model (LLM)-based question-answering. The project focused on analyzing annual reports from South African companies, with the goal of enabling Unifi, a sustainability data and analytics company, to efficiently obtain specific sustainability information. The report details the data preprocessing pipeline, embedding and vector store creation, LLM selection and prompt engineering, Q&A interaction, and post-processing and evaluation procedures. The solution demonstrated promising accuracy in extracting predefined activity metrics, highlighting the potential of this approach for automating sustainability data collection. The report concludes with a discussion of the project's strengths and weaknesses, trade-offs, and constraints, as well as potential improvements and extensions for future research and applications.

1 Introduction

1.1 Problem Statement

Many companies are striving to integrate sustainability initiatives into their core business practices. This shift is driven not only by a desire to contribute positively to society and the environment but also by the recognition that sustainability can enhance business performance. Sustainable companies often operate with greater efficiency and effectiveness, optimizing resource utilization, minimizing waste and energy costs, and fostering a more engaged and productive workforce.

However, managing sustainability initiatives effectively requires the collection and analysis of specific business metrics that are not readily available from traditional enterprise

resource planning (ERP) systems. One valuable source of sustainability data is the integrated reports that companies publish annually. These reports offer a comprehensive overview of a company's performance, financial health, and progress towards sustainability goals. They also provide transparency to stakeholders, including shareholders, investors, regulators, employees, and the public.

Despite the public availability of annual reports, extracting and analyzing relevant data from these documents presents several challenges. Annual reports often contain unstructured data, lack standardized formats, and vary significantly in structure and presentation across different companies. This makes it difficult to efficiently gather and compare sustainability information.

1.2 Objectives

The primary objective of this project was to develop a solution that parses annual reports in PDF format and automatically extracts information about predefined sustainability activity metrics. This solution aimed to enable Unifi, a sustainability data and analytics company, to efficiently obtain specific sustainability data from large corporations.

1.3 Scope and Methodology

This project focused on analyzing annual reports from companies in South Africa. The solution was developed using open-source and free tools, platforms, and large language models (LLMs), adhering to the competition's budget constraints. A key requirement was to ensure the solution's generalizability for future use with updated data and potentially different activity metrics.

The methodology employed involved a combination of natural language processing (NLP) techniques, document parsing tools, and LLM-based question-answering. The project included data exploration and preprocessing, embedding and vector store creation, LLM selection and prompt engineering, Q&A interaction, and post-processing and evaluation.

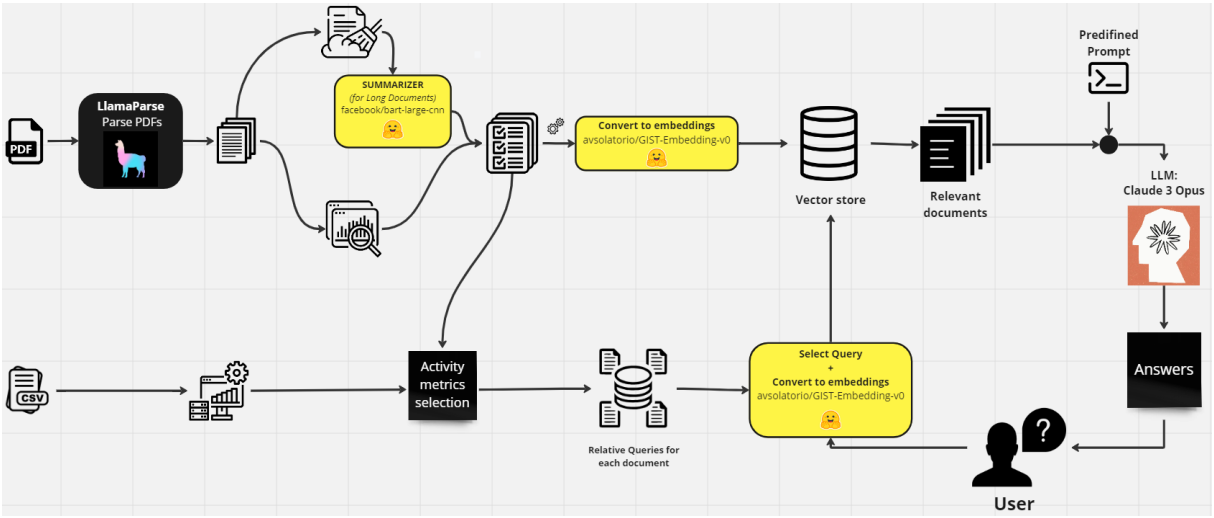

Figure 1: Solution Overview

2 Data Understanding and Exploration

2.1 Data Loading and Overview

The project utilized several datasets provided for the AI competition:

- **PDFnames.csv**: This file contained the names and corresponding file paths of the annual reports in PDF format.
- **AMKEY_GoldenStandard.csv**: This dataset listed the predefined activity metrics (AMKEYs) and their associated queries used to extract information from the reports.
- **ActivityMetricsSynonyms.csv**: This file provided synonyms and alternative names used by different companies to refer to the activity metrics.
- **Train.csv**: This dataset contained training data with values for the activity metrics from the years 2019 to 2021.
- **SampleSubmission.csv**: This file served as a template for the expected output format.

2.2 Data Visualization and Exploration

Several data visualization techniques were employed to gain insights into the datasets:

1. **Null vs non-Null Values Distribution**

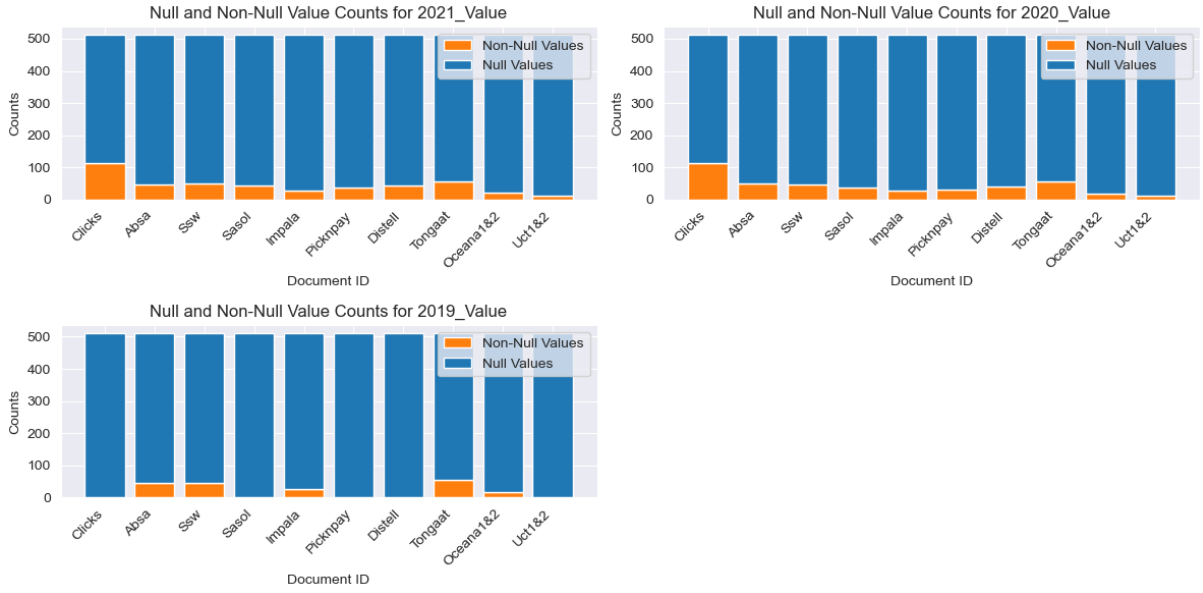

Figure 2: distribution of null and non-null values for each year (2019, 2020, 2021)

2. **Text Statistics**: The number of numerical tokens and total word count for each document, annotated with the total number of pages.

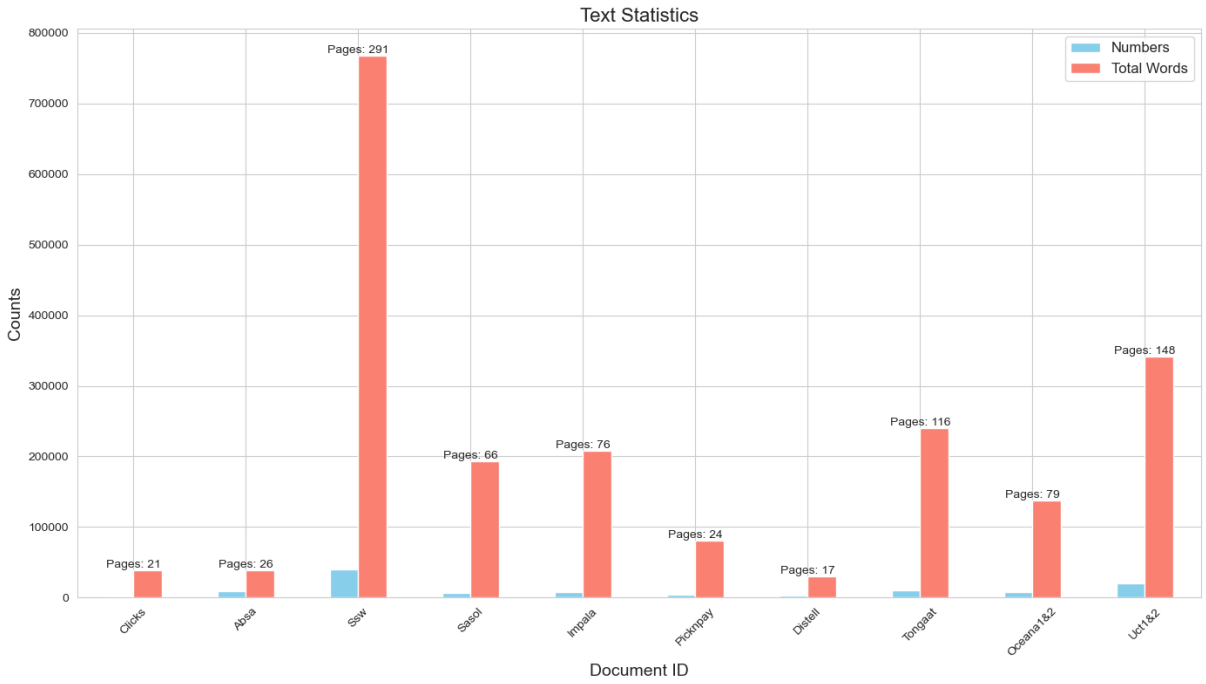

Figure 3: Text statistics for each document before preprocessing

These visualizations revealed valuable insights, such as the prevalence of null values in certain years and documents, the varying lengths and numerical content of the reports, and the distribution of synonyms used by different companies.

2.3 Challenges and Insights

The data exploration phase highlighted several challenges:

1. **Missing Data**: The presence of null values in the training data indicated that some activity metrics might not be reported by all companies or in all years.
2. **Synonym Handling**: The use of synonyms and alternative names for activity metrics required careful consideration to ensure accurate data extraction.
3. **Document Structure Variability**: The diverse structures and formats of the annual reports posed challenges for parsing and extracting information consistently.

These challenges informed the subsequent data preprocessing and LLM interaction strategies. For instance, the synonym handling challenge was addressed by incorporating synonym mapping into the document parsing process. Additionally, the document structure variability was mitigated by employing a combination of text cleaning techniques and LLM-based question-answering that could handle diverse text formats.

Overall, the data understanding and exploration phase provided crucial insights into the characteristics and challenges of the datasets, paving the way for effective data preprocessing and model development.

3 Data Preprocessing

3.1 Documents Parsing

The **annual reports**, provided in **PDF format**, necessitated a robust **PDF parsing library** for text extraction. Though **PyPDF2** was an option, the choice fell on **LlamaParse**, a cutting-edge tool launched on **February 20, 2024**. Unlike PyPDF2, which solely extracts text, making it challenging for language models to discern columns and rows in complex tables, LlamaParse demonstrates **superior performance** in handling intricate PDF structures. Its standout feature is the ability to extract documents into **markdown format**. This capability is particularly advantageous for extracting tables in an organized format, preserving the clarity of multi-index tables.

LlamaParse was employed to load each PDF document, efficiently extracting the text content in markdown format. The extracted text was then segmented into individual pages, using the delimiter `\n---` for page-level segmentation. This method proved especially beneficial in maintaining the structure of tables and other complex elements, facilitating further processing and analysis. The enhanced clarity in data extraction offered by LlamaParse's markdown output significantly improved the handling of the reports' structured content.

3.2 Document Cleaning Techniques

Several techniques were implemented to clean and prepare the annual report documents for efficient data extraction:

3.2.1 Addressing Fiscal Year Abbreviations

The documents often used abbreviations like "FY22" to represent fiscal years. To ensure consistency and facilitate year-specific data extraction, these abbreviations were replaced with their full year equivalents (for instance, "2022"). This process also included the flexibility to handle different year formats based on the specified YEAR variable.

3.2.2 Removing Unnecessary Pages

Analysis revealed that the first and last pages of most documents typically contained irrelevant information like cover pages and table of contents. Therefore, these pages were removed to reduce the amount of data to be processed. However, for the 'Absa' document, only the first page was removed as the last page contained relevant information.

Additionally, any pages that did not contain information about the specified YEAR (2022 in our case) were also removed. This helped to further focus the data extraction process on the relevant year. The number of pages removed from each document was reported to track the impact of this cleaning step.

These initial cleaning steps significantly reduced the data size and complexity, improving the efficiency and accuracy of subsequent data extraction processes.

3.2.3 Eliminating Unwanted Text Patterns

Further analysis identified specific text patterns within certain documents that were irrelevant for data extraction, such as headers, footers, and website URLs. A predefined mapping was created to associate document IDs with the corresponding text patterns to be removed. A helper function streamlined the process of cleaning these patterns from the document content.

3.2.4 Summarizing Long Pages

Some documents contained pages with excessive text that could hinder efficient data extraction. To address this, a summarization model was used to condense the content of pages exceeding a specified token length. This process targeted specific documents identified as having particularly long pages. The number of pages summarized for each document was reported to track the impact of this step.

3.3 Preprocessing Pipeline and Impact

The document cleaning techniques were combined into a preprocessing pipeline that was applied to all documents. This pipeline consisted of the following steps:

1. Replacing fiscal year abbreviations with full year representations.
2. Removing unnecessary pages (first, last, and non-relevant year pages).
3. Eliminating unwanted text patterns based on document-specific mappings.
4. Summarizing long pages in selected documents.

The preprocessing pipeline significantly reduced the data size and complexity, leading to several benefits:

- Improved Efficiency: By removing irrelevant information, the pipeline enabled faster processing and reduced computational costs.
- Enhanced Accuracy: By focusing on relevant content and standardizing year formats, the pipeline improved the accuracy of data extraction.
- Increased Focus: By summarizing long pages, the pipeline allowed the LLM to focus on the most important information.

By incorporating LlamaParse into the preprocessing pipeline, the project ensured efficient and accurate extraction of textual content from the PDF documents, preparing the data for subsequent analysis and LLM interaction.

The impact of the preprocessing pipeline was evident in the reduced data size and improved text statistics. This prepared the documents for efficient and accurate data extraction using embeddings, vector stores, and LLM-based question-answering.

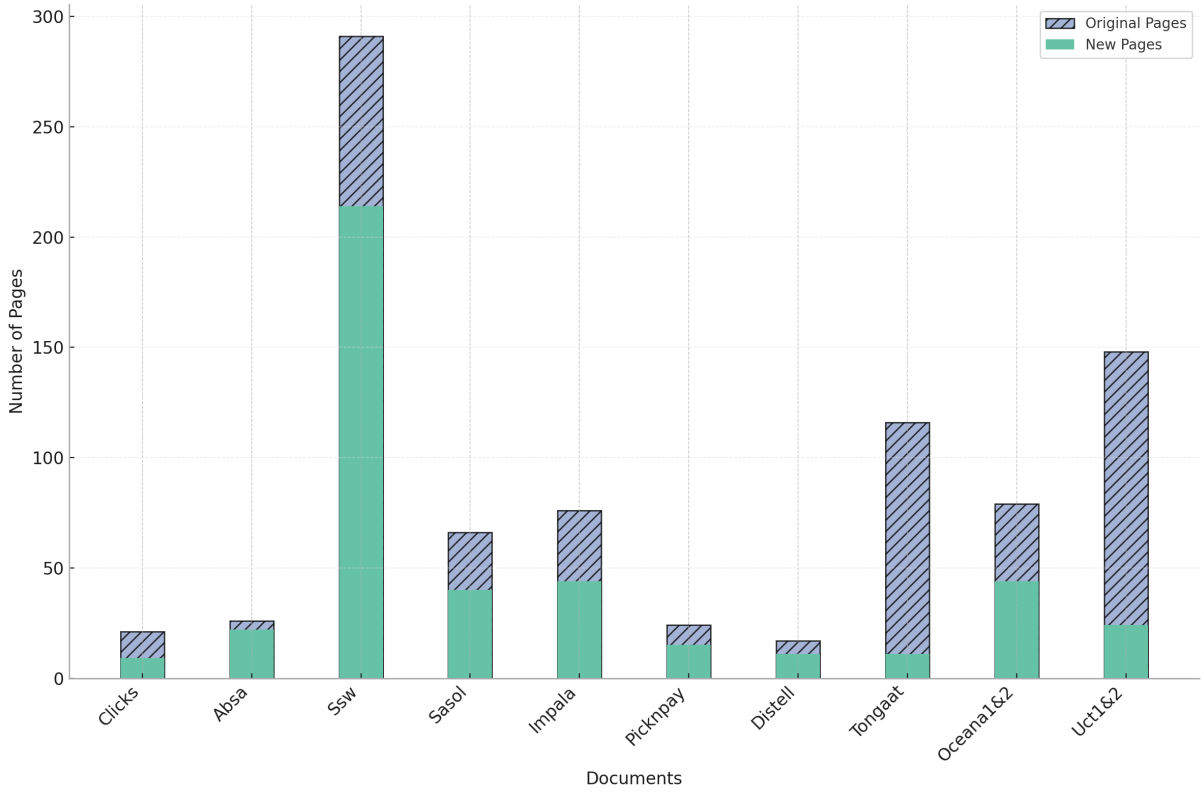

Figure 4: Number of pages before and after preprocessing

4 Embeddings and VectorStore

4.1 Selecting an Embedding Model

Embeddings play a crucial role in converting textual data into a numerical representation that can be understood by machine learning models. Various embedding models exist, each with its own strengths and weaknesses. For this project, the focus was on finding a model that balanced speed and performance effectively.

Two embedding models were considered:

- all-MiniLM-L6-v2: This model is known for its speed and efficiency, making it suitable for large datasets.
- GIST (avsolatorio/GIST-Embedding-v0): This model offers a good balance between speed and performance, providing accurate representations of sentence meanings.

After experimentation and evaluation, GIST was selected as the primary embedding model due to its superior performance in capturing semantic relationships within the documents.

4.2 Creating a FAISS Vector Database

Once the embedding model was chosen, it was used to convert the preprocessed document content into numerical vectors. These vectors were then stored in a FAISS vector database.

FAISS is a library for efficient similarity search and clustering of high-dimensional vectors.

The FAISS vector database allowed for fast and accurate retrieval of documents relevant to specific queries. This capability was essential for the subsequent LLM-based question-answering process, where the model needed to access relevant context from the documents to generate accurate answers.

4.3 Justification for Chosen Approach

The selection of GIST as the embedding model and FAISS as the vector database was based on several factors:

- Performance: GIST embeddings provided a good balance between speed and accuracy, effectively capturing the semantic meaning of sentences within the documents.
- Efficiency: FAISS enabled fast and efficient similarity search, allowing the LLM to quickly access relevant context from the vector database compared to ChromaDB.
- Scalability: Both GIST and FAISS are scalable solutions, capable of handling large datasets effectively.

This approach ensured that the LLM had access to accurate and relevant document representations, facilitating the extraction of precise information about sustainability activity metrics.

5 LLM Selection and Prompt Engineering

5.1 Experimentation with Different LLMs

Large language models (LLMs) have demonstrated remarkable capabilities in understanding and generating human-like text. To leverage this potential for sustainability data extraction, various LLMs were explored, including both local implementations and models accessible through API calls.

Initially, local LLM implementations were tested. However, these models did not produce satisfactory results, particularly in terms of output quality and accuracy. Consequently, the focus shifted to exploring LLMs available via APIs, such as GPT-3.5 and Claude 3 (released on March 4, 2024).

5.2 Rationale for Choosing Opus

Among the tested models, Claude 3 exhibited impressive performance. Notably, even the sonnet version of Claude 3, which is more cost-effective than the Opus version, showed promising results. However, after careful consideration and comprehensive evaluation, Opus was ultimately selected as the LLM of choice for this project.

The decision to use Opus was based on several factors:

- Superior Performance: Opus consistently outperformed other tested models in accurately extracting sustainability data from the documents.

- Advanced Capabilities: Opus demonstrated a strong ability to understand complex prompts, handle diverse text formats, and generate well-structured and informative responses.
- Cost-Effectiveness: While Opus is more expensive than the sonnet version of Claude 3, its superior performance justified the additional cost for this project.

5.3 Prompt Template Design and Features

Prompt engineering is crucial for guiding LLMs towards generating desired outputs. A carefully designed prompt template was developed to instruct the LLM on how to extract statistical variables from the documents, focusing specifically on data from the years 2021 and 2022.

The prompt template incorporated several key features:

- Contextual Relevance and Year Specification: The template emphasized the importance of extracting information only from the relevant years, ensuring the model disregarded data from other periods.
- Conversion and Standardization of Units: It provided specific instructions for converting various units (e.g., percentages, currencies, volumes) to a standardized form.
- Formatting Rules for Numbers: The template included rules for formatting numbers (e.g., removing commas, spaces) to ensure consistency and facilitate data analysis.
- Handling Missing Data: The template instructed the LLM to assign 'N/A' when a variable was not present in the context, providing a systematic way to deal with missing information.
- Output Structuring: The template defined a clear output format, aligning variable names with their corresponding values from 2021 and 2022.

This structured and informative prompt template guided the LLM towards generating accurate, consistent, and well-formatted responses, facilitating the extraction and analysis of sustainability data from the annual reports.

6 Q&A and Data Extraction

6.1 Activity Metric/Query Checker

Before interacting with the LLM, the activity metrics and their associated queries were carefully reviewed. This involved checking for consistency and ensuring that the queries were well-formulated to extract the desired information from the documents.

The approach to Q&A and data extraction was tailored specifically for each document. I leveraged The identified synonyms for query optimization, ensuring the questions were as precise and relevant as possible This analysis revealed that there were a total of 460 activity metrics to be extracted from the documents. Additionally, the distribution of queries across different document groups was visualized to understand the workload distribution.

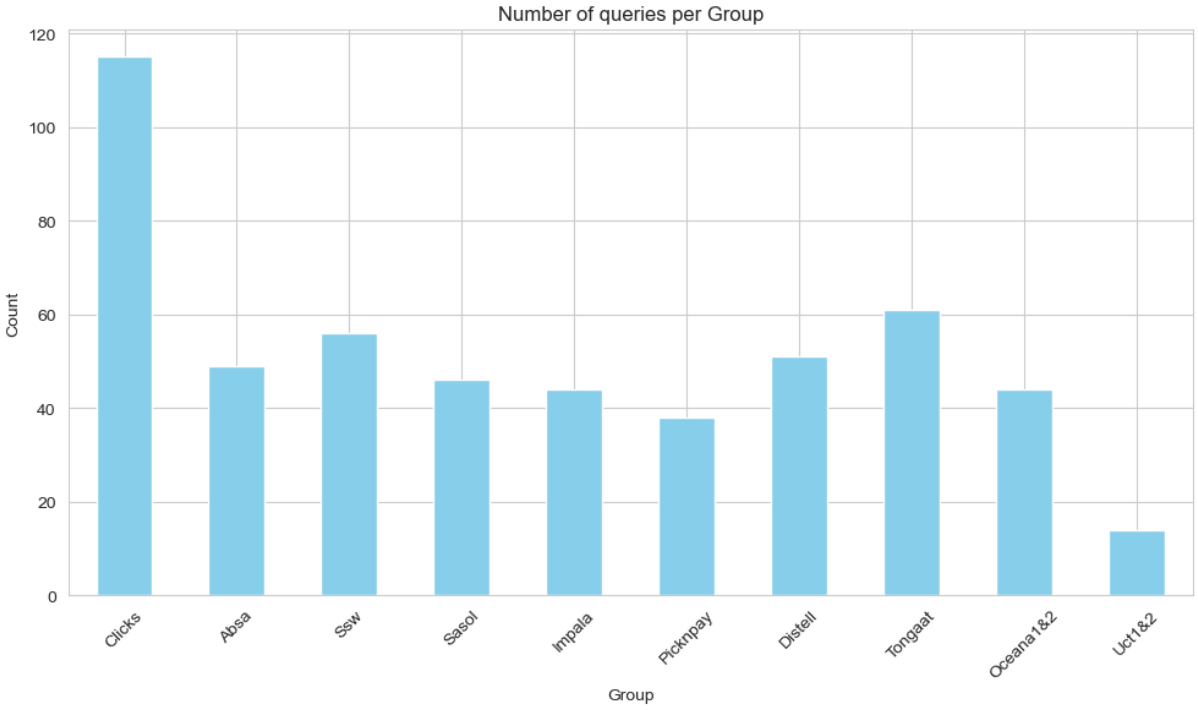

Figure 5: number of queries associated with each document group

6.2 Optimizing Query Processing for LLMs

To optimize the LLM interaction process and minimize costs, the queries were tailored for each document. This involved identifying activity metrics that were relevant to specific documents based on synonym mapping and historical data availability.

For example, for documents like 'Clicks' and 'Picknpay', only activity metrics that had recorded values in 2021 were included in the queries. This helped to reduce the number of unnecessary queries and focus the LLM's attention on relevant information.

6.3 LLM Interaction & Information Retrieval Workflow

Utilizing the **200,000 context length capacity** of **Claude 3**, the process's efficiency in interacting with the Language Learning Model (LLM) was significantly enhanced. This capacity of **Claude 3** enabled the simultaneous processing of all 20 pertinent pages in a *single pass*, streamlining the retrieval of query responses while augmenting the *accuracy* and *relevance* of the extracted information.

With optimized queries and a prompt template, the LLM was tasked to extract data from these documents. Relevant content was provided to the model, which then analyzed the context and generated responses in a specified format. To ensure cost-efficiency and adherence to API request limits, pauses were implemented between processing each document. This approach fostered controlled and efficient data extraction within the competition's budget constraints.

The LLM interaction process yielded promising results. The model successfully extracted data for a significant portion of the activity metrics. The output was rendered in

a *user-friendly* and *analytically convenient* format:

```
Variable Name: [2022: Value, 2021: Value]
```

This format offers a clear and concise representation, facilitating easy comparison and analysis of data trends across different time periods, particularly useful for tracking variations or trends in values associated with variable names.

The extracted data, with its clear format and accuracy, was then passed on to the post-processing stage for further refinement and analysis.

7 Post-Processing and Evaluation

7.1 Data Extraction and Formatting

The responses generated by the LLM required further processing to extract the data and format it into a structured and usable form. This involved:

- Cleaning the LLM Output: Removing unnecessary text and formatting elements from the responses.
- Extracting Variable Names and Values: Identifying the variable names and their corresponding year-value pairs from the cleaned output.
- Constructing a DataFrame: Organizing the extracted data into a Pandas DataFrame with columns for variable name, document name, year, page number, confidence score, and year-specific value.

This is a portion of the constructed data frame:

ID	Variable Name	2022	page_number	confidence_score	2022_value
35_X_Clicks	Audit committee meeting attendance rate	N/A	16	0.479788	0.0
46_X_Clicks	BBBEE procurement spend from Exempt Micro Enterprises	N/A	18	0.631564	0.0
49_X_Clicks	B-BBEE Scorecard Level	4	16	0.518139	4.0
50_X_Clicks	Percentage of black board members	60	4	0.541434	60.0
52_X_Clicks	Board meeting attendance rate	N/A	16	0.468244	0.0

Table 1: Created DataFrame

7.2 Analysis of Findings

The extracted data was then analyzed to assess the performance of the overall solution. This included:

- Evaluating Predictive Analytics for 2021: Comparing the extracted values for 2021 with the actual values in the training data to calculate performance metrics such as accuracy, precision, recall, and F1 score.
- Null Value Analysis: Analyzing the distribution of null values in the extracted data to identify missing information and potential areas for improvement.

7.3 Performance Metrics (Accuracy, Precision, Recall, F1 Score)

The evaluation of predictive analytics for 2021 revealed promising results:

- Accuracy: 0.942
- Precision: 0.962
- Recall: 0.978
- F1 Score: 0.97

These metrics indicated that the solution achieved a good level of accuracy in extracting sustainability data from the annual reports.

Overall, the post-processing and evaluation phase provided valuable insights into the performance of the solution and identified opportunities for further refinement and improvement.

8 Results & Discussion

8.1 Calude 3 Evaluation

Model	Embedding	Runtime	Cost	LB	PB
Sonnet	GIST	336.10 sec	0.24$	0.9445	0.9382
Opus	GIST	703.91 sec	3.23$	0.9471	0.9429

Table 2: Model performance metrics

8.2 Ablation Study

Techniques	Round 1	Round 2	Round 3
PyPDF	✓	✓	×
LlamaParser	×	×	✓
With Preprocessing	×	✓	✓
Model Score:	0.9197	0.9321	0.9471

Table 3: Ablation study results across three rounds

The study involved an ablation analysis conducted over three testing phases to evaluate the efficacy of different text extraction techniques on model performance. The analysis was structured to establish a baseline using a combination of preprocessing and the LlamaParser tool. During the initial phase (Round 1), the model operated without the aid of LlamaParser and preprocessing, resulting in a model score of **0.919**. The subsequent phase (Round 2) integrated preprocessing, which led to a marginal enhancement in model performance, elevating the score to **0.932**. The final phase (Round 3) excluded PyPDF while retaining the use of LlamaParser and preprocessing, culminating in a further improved model score of **0.947**. This incremental increase in model scores across the rounds suggests that the synergistic application of preprocessing and LlamaParser is most conducive to optimizing performance.

8.3 What Worked and What Didn't

The developed solution demonstrated promising results in automatically extracting sustainability data from annual reports. Several aspects of the approach contributed to its success:

- Effective Preprocessing: The document cleaning techniques significantly reduced data complexity and improved the accuracy of data extraction.
- Appropriate Embeddings and VectorStore: The GIST embedding model and FAISS vector database enabled efficient retrieval of relevant document context for the LLM.
- Powerful LLM and Prompt Engineering: The Opus LLM, guided by a carefully designed prompt template, effectively extracted data from diverse text formats.

However, some challenges remained:

- Missing Data: Despite the efforts to optimize queries and improve LLM performance, some activity metrics remained unextracted due to non-reporting or extraction failures.
- Synonym Handling: While synonym mapping was implemented, it might not have captured all possible variations in how companies refer to activity metrics.
- Cost Considerations: Using powerful LLMs like Opus can be expensive, especially for large-scale data extraction tasks.

8.4 Trade-offs and Considerations

Several trade-offs were considered during the project:

- Speed vs. Accuracy: While faster embedding models like all-MiniLM-L6-v2 were available, GIST was chosen for its superior accuracy in capturing semantic relationships.
- Local vs. API-based LLMs: Local LLM implementations were initially explored but ultimately abandoned due to performance limitations. API-based LLMs offered better performance but incurred additional costs.

- Number of Queries vs. Cost: Optimizing queries for each document reduced the number of API calls but required additional analysis and development effort.

8.5 Constraints and Implications

The project was subject to certain constraints:

- Budget: The competition limited the use of proprietary services and tools to a maximum cost of $20 per month. This influenced the choice of LLMs and the need for query optimization.
- Data Availability: The training data only included values for certain activity metrics in specific years, which limited the evaluation and potentially the performance of the solution.

These constraints highlight the importance of carefully considering cost-effectiveness and data availability when developing AI-powered solutions.

Overall, the discussion section provides a critical analysis of the project's strengths and weaknesses, trade-offs, and constraints, offering valuable insights for future improvements and applications.

9 Model Deployment

While this project focused primarily on model development and evaluation, a realistic and simple deployment strategy can be outlined to operationalize the solution for real-world use.

Deployment Strategy:

- *Cloud-Based API Integration:* The LLM (Opus) is already accessible through an API. This eliminates the need for local model hosting and maintenance.
- *Web Application Interface:* Develop a user-friendly web application that allows users to upload annual reports in PDF format.
- *Preprocessing and Data Extraction:* Integrate the preprocessing pipeline and LLM interaction code into the web application backend.
- *Data Visualization and Download:* Display the extracted sustainability data in a clear and interactive format within the web application. Provide options for users to download the data in various formats (e.g., CSV, Excel).
- *Containerization and Scalability:* Containerize the web application using technologies like Docker to ensure portability and facilitate deployment on cloud platforms. This also enables horizontal scaling to handle increased user demand.

Benefits of this Strategy:

- Simplicity: Leverages existing API access for the LLM, minimizing deployment complexity.

- Accessibility: A web application provides a user-friendly interface for a wide range of users.
- Scalability: Containerization allows for easy scaling to accommodate growing data extraction needs.
- Cost-Effectiveness: Utilizing cloud platforms for deployment can optimize resource utilization and minimize infrastructure costs.

Additional Considerations:

- Security: Implement robust security measures to protect user data and ensure the integrity of the extracted information.
- Monitoring and Maintenance: Continuously monitor the performance of the deployed solution and perform regular maintenance to ensure optimal functionality.
- User Feedback and Improvement: Integrate mechanisms for collecting user feedback to identify potential issues and drive ongoing improvements to the model and deployment strategy.

By adopting this realistic and simple deployment strategy, the AI-powered sustainability data extraction solution can be effectively operationalized, providing valuable insights to companies and stakeholders committed to advancing sustainability goals.

10 Conclusion and Future Work

10.1 Summary of Achievements

This project successfully developed an AI-powered solution for extracting sustainability data from annual reports. The solution employed a combination of NLP techniques, document parsing tools, and LLM-based question-answering to overcome the challenges associated with unstructured data and diverse document formats. The evaluation results demonstrated promising accuracy in extracting predefined activity metrics, highlighting the potential of this approach for automating sustainability data collection.

10.2 Potential Improvements and Extensions

Several avenues exist for further improving and extending the solution:

- Addressing Missing Data: Exploring techniques like data imputation or incorporating additional data sources could help fill in missing values.
- Enhancing Synonym Handling: Implementing more sophisticated synonym detection and mapping strategies could improve the solution's ability to handle variations in activity metric names.
- Exploring Alternative LLMs: Investigating the use of other LLMs, potentially fine-tuned for specific data extraction tasks, could further enhance performance.

- Expanding Scope: The solution could be extended to handle annual reports from other countries or industries, requiring adjustments to the synonym mapping and potentially the prompt template.
- Incorporating User Feedback: Integrating a feedback mechanism could allow users to identify and correct extraction errors, leading to continuous improvement.

By addressing these potential improvements and extensions, the solution can become even more robust and versatile, contributing significantly to the efficient and accurate collection of sustainability data from annual reports. This, in turn, can empower companies and stakeholders to make informed decisions and drive meaningful progress towards sustainability goals.

11 REFERENCES

Zindi - Unifi Value Frameworks PDF Lifting Competition

LlamaIndex Blog - Introducing LlamaCloud and LlamaParse

Hugging Face - BART Large CNN

Anthropic Docs - Claude Models Overview

Anthropic News - Claude-3 Family