GHG Emissions Insights & Analysis Agent: Overview

1. Proposed Solution & Core Capabilities

This document details my AI-powered agent designed to serve as an assistant for carbon emissions analysis. It addresses the challenge of having critical data spread across multiple complex formats i.e. CSVs, a dense PDF knowledge base and peer reports with embedded tables. The agent is a conversational interface to access data and tools to get instant, accurate answers without needing to be data scientists. Core capabilities:

Knowledge Base- Answers general questions about emissions accounting by referencing the GHG Protocol.

Data Validation- Audits the company's emissions data against the protocol's guidelines, checking both classification and the plausibility of emission factors.

Comparative Analysis: Benchmarks the company's emissions (down to sub-scope level) against pre-processed peer data.

Text-to-SQL Analysis: Converts natural language questions into SQL queries to perform specific calculations and data retrieval from the company's CSVs.

Automated Reporting: Generates dynamic summary reports with key insights for any combination of emission.

2. Technical Approach & Architecture

The system is created as an "agent" that uses a tool-based design to ensure reliability and prevent hallucination. The core of the agent is an intent router running on Google's Gemini 1.5 Flash model, which directs user queries to the appropriate specialized tool.

Technology Stack: Core Logic: Python ; AI Model: Google Gemini 1.5 Flash ;UI: Streamlit *;* Frameworks: LangChain (for RAG), Pandas (for data manipulation), pandasql (for Text-to-SQL)

Agent Tools:

Router (get_intent): This is my core idea to get accurate solutions to the core questions without hallucination. It analyses the user's query and selects one or more tools to execute. It is capable of sequencing multiple tools for complex, multi-part questions. I've created tools which have embedded prompts which can run for various type of queries to get precisely what is needed.

Protocol QA Tool: A standard RAG (Retrieval-Augmented Generation) pipeline. The GHG Protocol PDF is loaded, chunked, and indexed into a FAISS vector store using GoogleGenerativeAIEmbeddings. The tool performs a semantic search to find relevant context before generating an answer.

SQL Query Tool: Converts natural language into SQL. A Gemini prompt, guided by few-shot examples, generates a SQL query which is then safely executed against the company's dataframes using pandasql.

Summary Report Tool: An automated tool that runs multiple, predefined SQL queries to gather key metrics (e.g., scope totals, top contributors) and then uses Gemini to synthesize these findings into a qualitative report.

Validation & Comparison Tools: These are hybrid tools that combine vector search (to get definitions from the protocol) and direct data analysis on the pandas DataFrames to perform their tasks.

3. Data Handling & Integration

CSVs (Company Data): Loaded directly into pandas DataFrames. These are treated as structured tables, accessible for direct querying via the SQL Tool or for targeted analysis by other tools.

PDF (GHG Protocol): Handled via the classic RAG pipeline described above. This converts the unstructured text into a searchable knowledge base.

PDFs (Peer Reports): This was the most complex data source, with tables embedded as images. To handle this, a separate, one-time pre-processing script (extract_peer_data.py) was created. It uses Gemini 1.5 Flash's multi-modal (VLM) capabilities to visually parse the relevant pages, extract the data from the tables, and save it into clean, analysis-ready CSVs. This decouples the slow, complex extraction from the agent.

4. Evaluation & Production Scaling

Evaluation: A robust, automated evaluation framework was developed to ensure agent quality.

1. Golden Dataset (evaluation_dataset.csv): A curated list of 10+ test questions covering all agent capabilities, each with an expected tool and a precise expected outcome.
2. Automated Script (evaluate_agent.py): A Python script that runs every query from the dataset through the agent.
3. LLM-as-a-Judge: The script then uses Gemini to programmatically score the agent's actual response against the expected outcome on two key metrics: Factual Accuracy and Relevance/Completeness.Scaling to Production:

- For Bigger Datasets:

Tabular (CSVs): The pandasql approach would become a bottleneck. The solution is to migrate the data to a scalable database (e.g., DuckDB, PostgreSQL).

PDFs: The FAISS vector store scales well, but for enterprise use, it would be moved to a dedicated vector database (e.g., Pinecone, Weaviate) for better management and performance

The Streamlit application can be containerized (e.g., with Docker) and deployed on a cloud service (like Google Cloud Run or Streamlit Community Cloud) that can auto-scale to handle concurrent user sessions. The existing caching (@st.cache_resource) already helps optimize performance.

5. Challenges, Improvements & Key Learnings

- Challenge: Data Extraction from Peer PDFs.

Using Gemini's VLM capability in a pre-processing step proved to be a highly effective and robust solution for handling tables embedded in images.

- Challenge: Multi-Part Questions.

The initial router logic was simplistic. It was upgraded to detect and create a sequence of tool calls, allowing the agent to break down a complex query into smaller, manageable steps.

- Future Improvements:

Interactive Visualizations: Generate charts and graphs for reports instead of just text and tables.

Proactive Insights: Develop a feature where the agent can analyze the data on a schedule and alert users to significant trends or anomalies.

More adaptive tooling can be implemented by using few shot training and which could effectively help build more accomplishing tools or skills into the framework.

Key Observations About the Data:

The company's scope1.csv and scope2.csv files are providing consumption data by facility, fuel/energy type, and date. This enables precise analysis, such as identifying that coal combustion at the "Manufacturing Plant" is the primary driver of Scope 1 emissions.

A major challenge is that peer reports contain highly aggregated data and, critically, present it in non-machine-readable formats (e.g., tables embedded within images in a PDF). This makes direct, granular comparison impossible without a pre-processing step and justifies the use of a multi-modal model for data extraction.