Project Report: Retrieval Augmented Generation Project (Email AI)

[**1. Introduction 3**](#)

[**2. Project Goals 3**](#)

[**3. Data Sources 3**](#)

[**4. System Design 3**](#)

[4.1 Overall System Design 3](#)

[4.2 Embedding Layer 3](#)

[4.3 Search Layer 3](#)

[4.4 Generative Layer 4](#)

[**5. Implementation Details 4**](#)

[**6. Performance Evaluation 4**](#)

[6.1 Query Search 4](#)

[Query 1: "What does Wall Street believe about enron?" 4](#)

[Query 2: "What is the service P&L for demand charges?" 5](#)

[Query 3 : "What is the impact of federal price limits on power companies?" 5](#_6imygb4t0op6)

[**7. Conclusion 6**](#)

# 1. Introduction

The Retrieval Augmented Generation Project, Email AI, aims to enhance email response generation by integrating advanced search mechanisms and AI-based text generation. This report outlines the design, implementation, and evaluation of the project.

# 2. Project Goals

- Develop an innovative and efficient system for generating responses to email queries.  
- Utilize state-of-the-art embedding and search techniques to enhance information retrieval.  
- Ensure high-quality and contextually relevant AI-generated responses.

# 3. Data Sources

The project leverages email data as the primary data source. The data can be found here -> https://www.kaggle.com/datasets/marawanxmamdouh/email-thread-summary-dataset

# 4. System Design

## 4.1 Overall System Design

The system architecture is designed to maximize efficiency and effectiveness in handling email queries. It consists of three main layers: Embedding Layer, Search Layer, and Generative Layer.  
  
The system follows a multi-layered approach:  
1. **Embedding Laye**r: Processes text data to create embeddings.  
2. **Search Layer**: Retrieves relevant information based on the query.  
3. **Generative Layer**: Produces AI-generated responses.

## 4.2 Embedding Layer

The embedding layer uses OpenAI text-embedding-ada-002 model to generate embeddings and load into chroma db

## 4.3 Search Layer

The **search layer** implements a robust search mechanism to fetch the most relevant chunks of text data.

A **caching mechanism** is integrated to store frequent queries and results, improving the system's response time.  
  
A **re-ranker** is employed to enhance the relevance of search results, using models like Cross-Encoder to re-evaluate and rank the results.

## 4.4 Generative Layer

:The generative layer uses GPT-based models to generate high-quality, contextually relevant responses. The prompt design ensures that the AI understands the context and requirements of the query.

# 5. Implementation Details

The implementation details include the integration of OpenAI Embedding models, the development of the caching and re-ranking mechanisms, and the creation of effective prompts for the generative model.

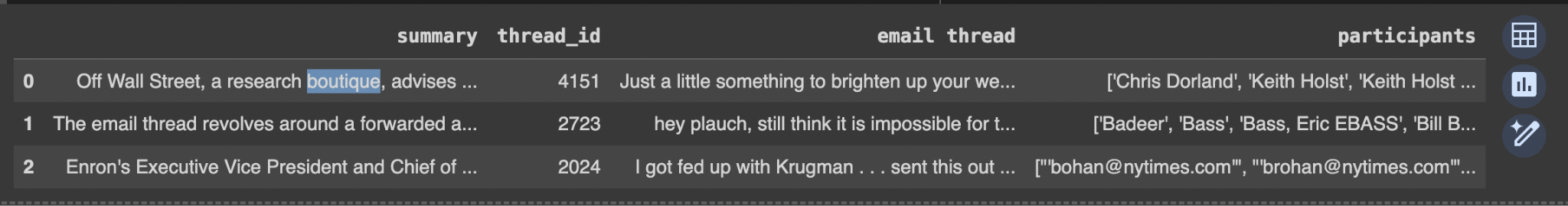
# 6. Performance Evaluation

## 6.1 Query Search

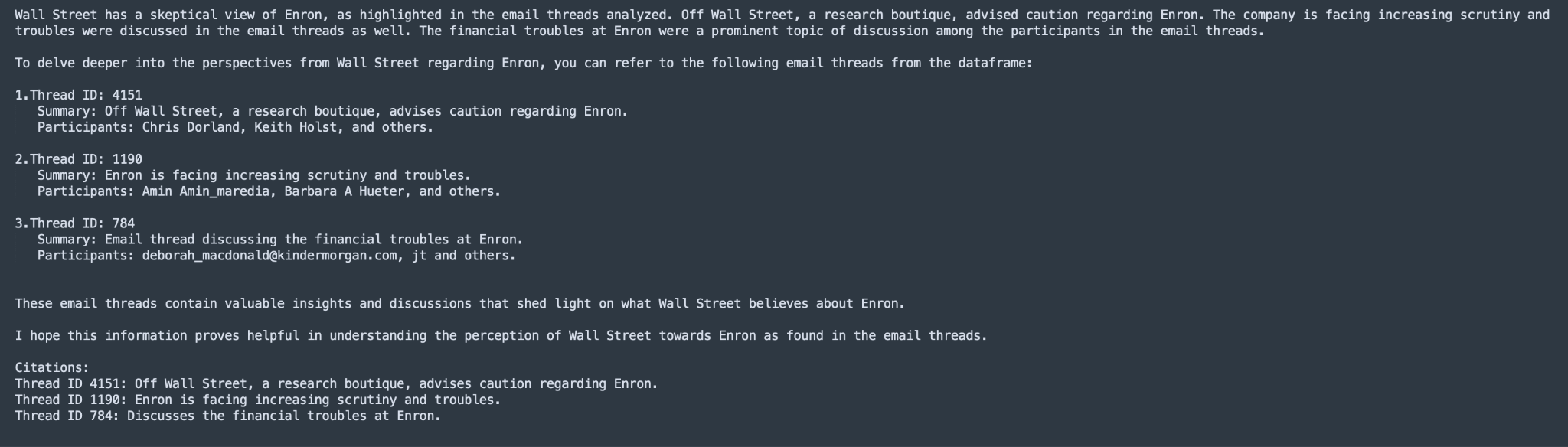
The system was tested against three self-designed queries to evaluate its performance. Below are the details:

### Query 1: "What does Wall Street believe about enron?"

**Search Layer Output**

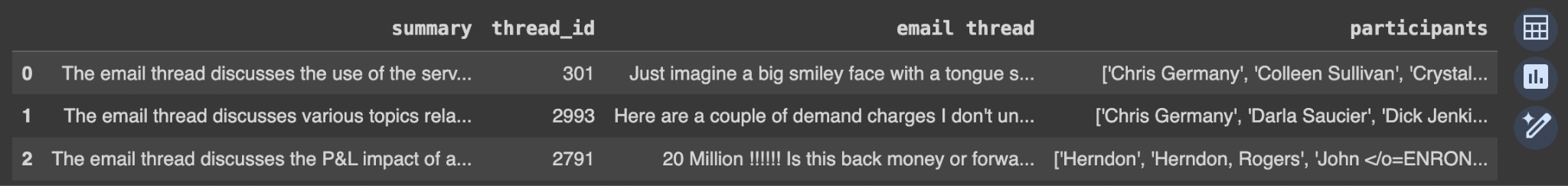


**Generative Layer Output**

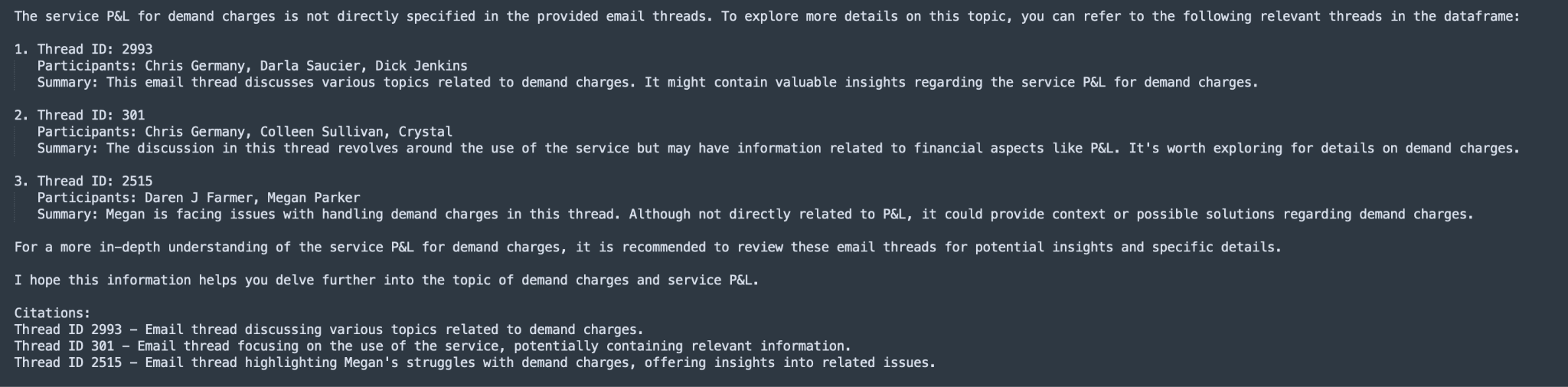
****

### Query 2: "What is the service P&L for demand charges?"

**Search Layer Output**



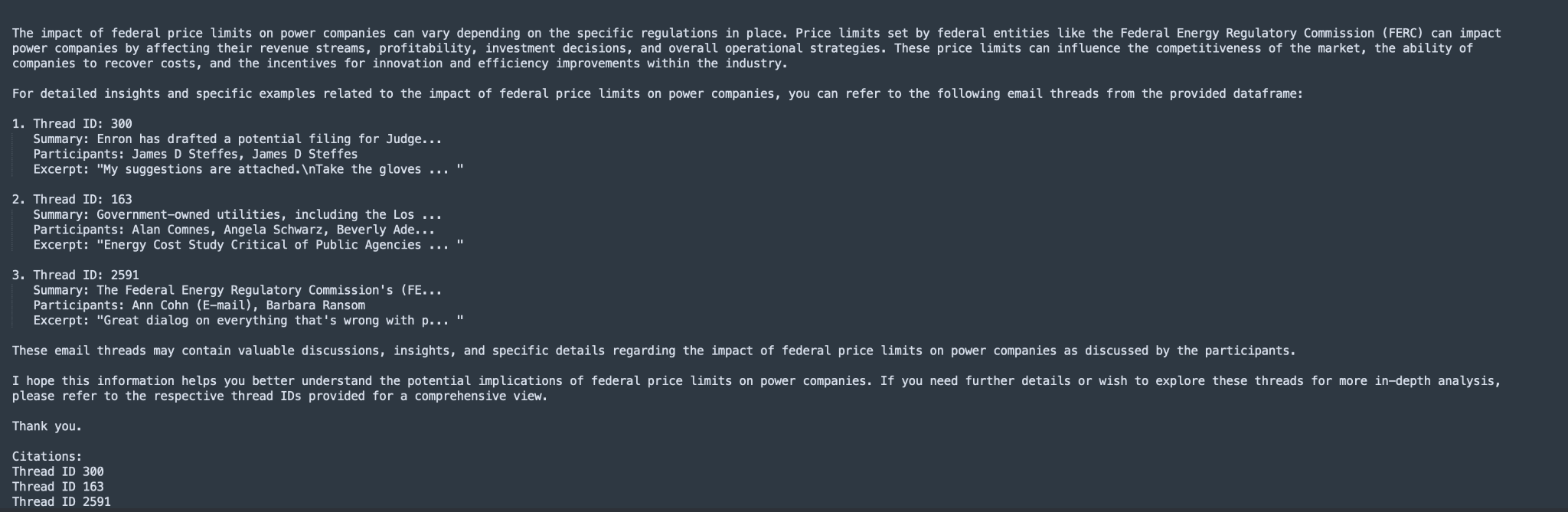
**Generative Layer Output**



### Query 3 : "What is the impact of federal price limits on power companies?"

**Search Layer Output**

  
**Generative Layer Output**

****

.

# 7. Conclusion

Email AI project successfully integrates retrieval and generation mechanisms to provide high-quality responses to email queries. The innovative design and effective implementation demonstrate significant improvements in handling and responding to email data.