PROJECT GOAL:

To create a email search system using RAG based on data of more than 20000 emails, using chromadb and chat completions API.

DATA SOURCE:

Data set available on kaggle having more than 20000 emails and their summaries in 2 separate csv files.

Link: <https://www.kaggle.com/datasets/marawanxmamdouh/email-thread-summary-dataset>

DESIGN CHOICES:

1. Data understanding and pre-processing-

Data contained subject, timestamp, from, to, body and summary column which we achieved after merging the 2 CSV files.

We removed noise from email body and subject column using regular expression and made it ready to add in data base as embedding for semantic search.

We didn’t required to do chunking as we already had csv with separate mails.

1. Adding data in chromaDB-

We created a persistent client and added data In it in batched as dataset was huge and had high number of characters. We used text-embedding-ada-002 embedding model for creating embeddings and also incorporated exception handling in it.

1. Creating a cache search system

We Created a separate collection as cache to check that collection first if we can find the relevant response there.

In case we were unable to find anything there we can move to search in main collection.

We created this so that we can save time of going through whole database for repetitive queries.

All the new queries will be added to cache for future use.

1. Cross Encoding-

We used re-ranking using cross encoding to get more relevant results and reduce effort of going through all the top 10 results.

1. Generative search-

We used chat\_completions API to address user query based on top 3 results we got from re-ranking.

We used prompt engineering to get the required output from these emails.

1. Evaluation-

We used some custom queries to test the efficiency and working of our model.

CHALLENGES FACED:

1. The email dataset was huge I face multiple failures and crashes while adding it to database, for this we tried tick token to count token size, chunking in case emails are large. At the end we added mails in batches and that solved the issue.
2. I tried creating a function search\_with\_cache in order to create a complete pipeline at the end, While testing it we were getting error as initially cache search results were empty, I resolved it with a little change in code.
3. The prompt for generative search I created was unable to give me required or well structured result, It was give too much or very less information. After certain hit and try attempts I was able to give clear instructions as prompt.