Research Report

by Shing Chung Z. Chiu

Initial Difficulities

No Domain Knowledge
No Question and Answer Dataset
ETL of the dataset is difficult
Study what Knowledge Does and how to apply it to the RAG flow I usually do

Initial Approches

Quick and Dirty Dify + Vector DB

I do this mainly to understand the pain points better.
This method will be AB test base line for my following research

ETL

I tried to use regex to transform the data to json format but it is not working

Chunking

I choose to chunk the text into different block that have similar approximate tokens size (500)

Generate Q&A for VectorRAG (Not Necessary)

I use LLM to generate Q&A based on the chunked documents

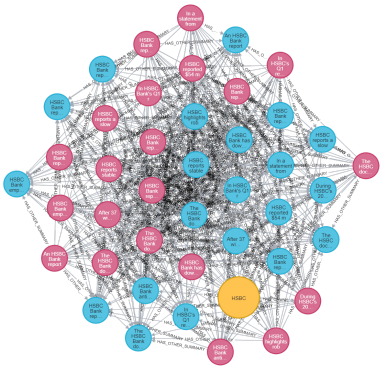
Create Summaries for different chunks

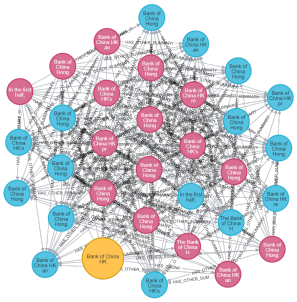
I use LLM to create Summaries for different chunks

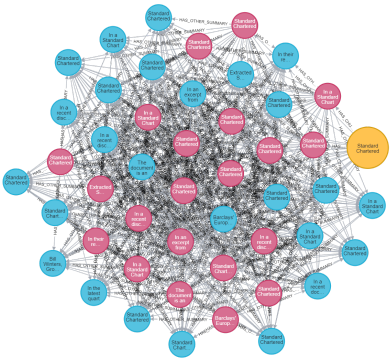
Add all the data to Milvus (VectorDB) and Neo4j (Knowledge Graph)

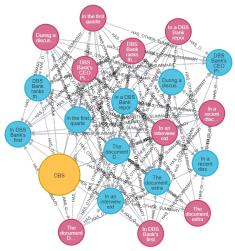
Insert most of the data to both Milvus and Neo4j
Milvus mostly focus on Generated Q&A and Neo4j focus on the summaries

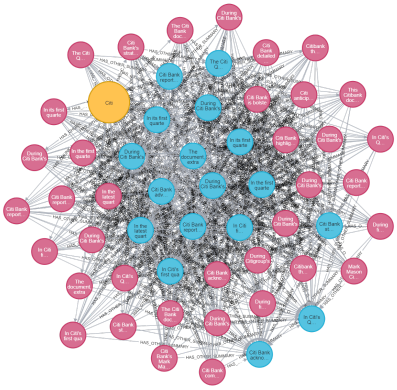
Difficulites for Neo4j first trial







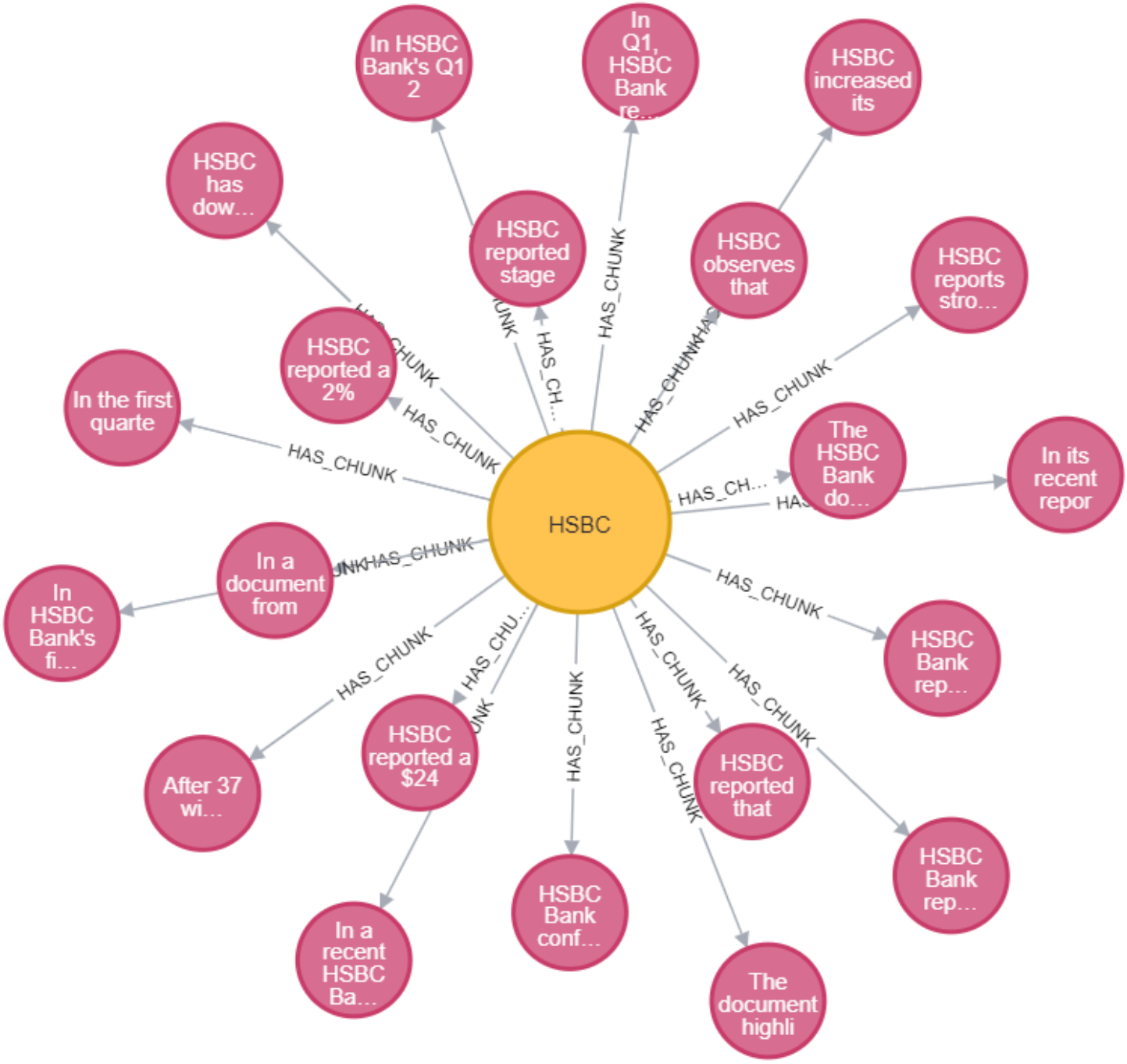




As you can see from the above graph, the summaries are too complex, and many nodes were duplicated.

Improvment

I removed the summaries nodes and combine all of them to one single article. Then replace the original chunk with the to the summary.



This is the relationship between the Vector-DB chunks and the bank.

Each chunks contains the summary of all chunks and the original paragraph of its own in order.

This will help LLM to better understand the context before and after itself and prevent the "**neglects the hierarchical nature**" problem suggested in the paper:

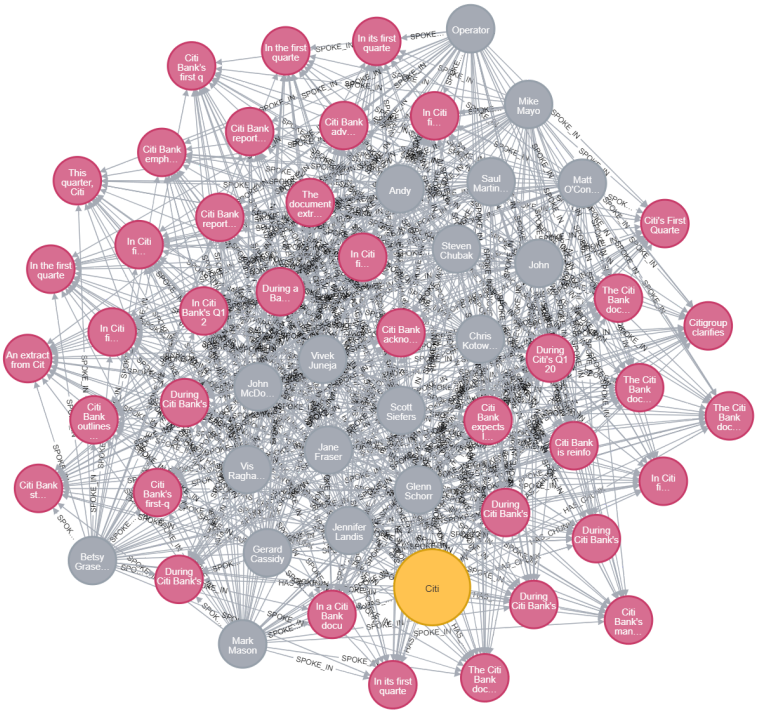
HybridRAG: Integrating Knowledge Graphs and Vector Retrieval

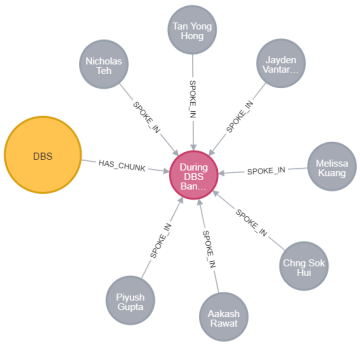
Augmented Generation for Efficient Information Extraction

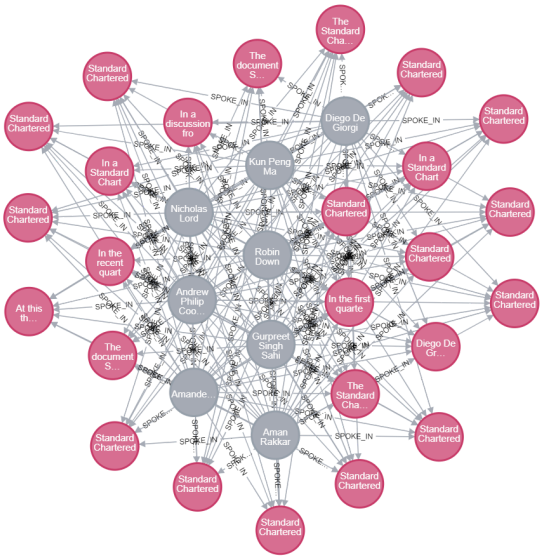
Based on that improvement, I started to create the speakers, questioners and bank relationships.

Difficulties for Neo4j Second approche

After extracting the speaker relationship from the Earning Calls, I put the speaker's information into the Knowledge as well.



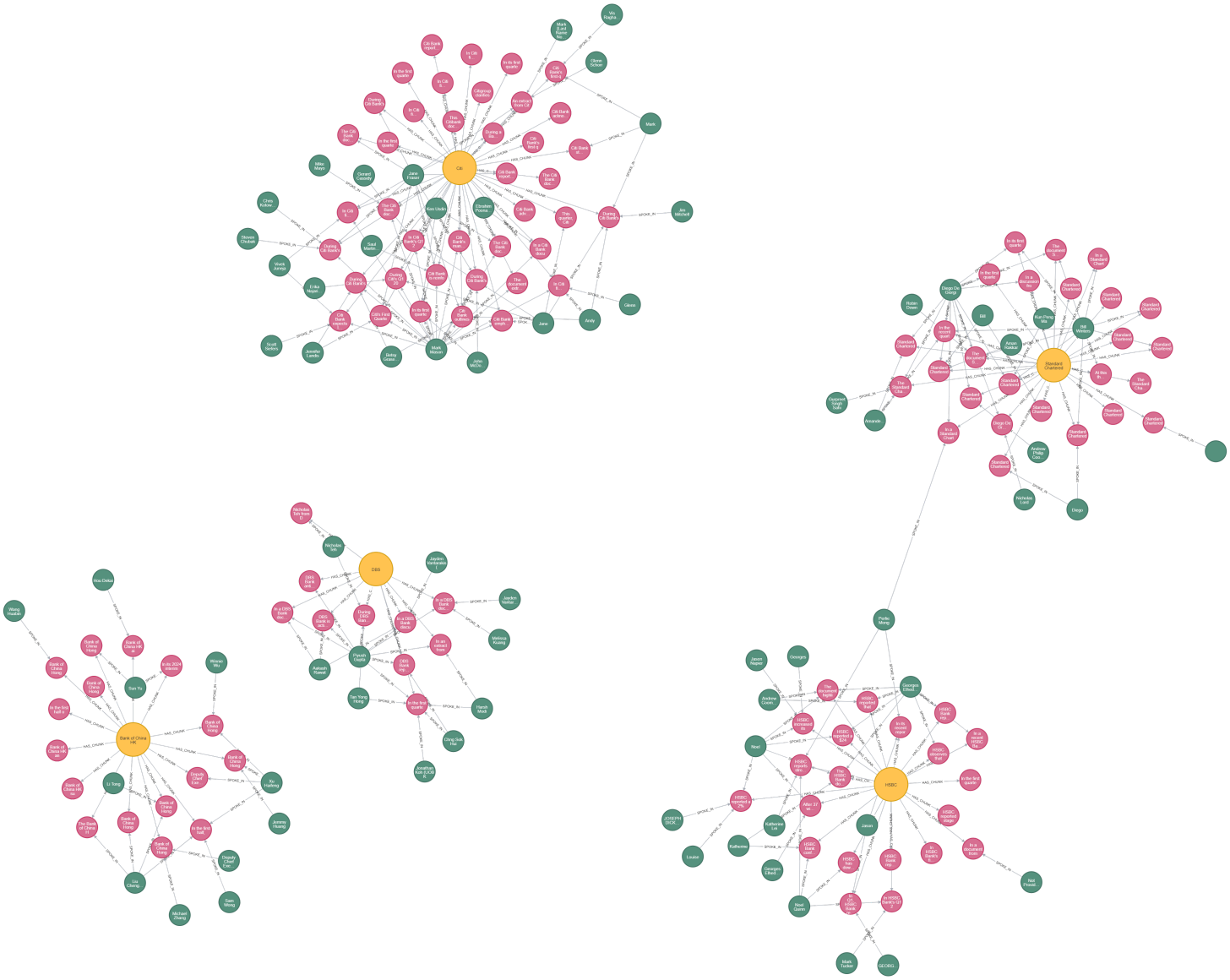




Neo4j is good on handling CSV and jsonl with pandas, but when massing with a more complex relationship, I need to be more careful. The above graph is the first attempt on insert the jsonl data of the earning call together with the extracted speaker's information. The resulting graphs seems "overconnected".

Improvement

To fix that problem, I read the extracted documents and adjected the prompts for extracting the data. I also modified the inserting code to be more effective.



This the resulting graph after the adjustment I made. It has become much more interesting compare than the pervious graph.
For example, Perlie Mong (At least with the same name and same company - KBW) asked questions in 2 different Earning Calls event.

Difficulities for the system on Third approch

Although the RAG quality improved on the first sight, but both methods still failed to answer some questions when taking look of a full picture is needed. So I decided to move on to a new approch.

Vector DB RAG	My version of Hybrid RAG
Approximate tokens size: 1844	Approximate tokens size: 1710
HSBC has reported a strong performance in various areas, including revenue growth, banking net interest income, and wealth management division. The bank has also achieved mid-single-digit loan growth over the medium-to-long term, despite subdued demand in Hong Kong and economic uncertainties.	HSBC has reported strong financial results for Q1 2024, with improved outlooks in Wealth and Global Payments Solutions, stable term deposits in Hong Kong, and robust growth in its Wholesale Transaction Banking and Wealth sectors. The bank maintains its 2024 guidance of at least $41 billion Banking Net Interest Income and mid-teens return on tangible equity.

I think Hybrid RAG follows the rules better and I think one of the reason is because after I added the pre and post information to the retrived chunks, the content is far easier for llama3.1 8B LLM to read.

The failed question was:

Who went to multiple banks' earnings calls event? Give me a short and simple answer.

Improvment

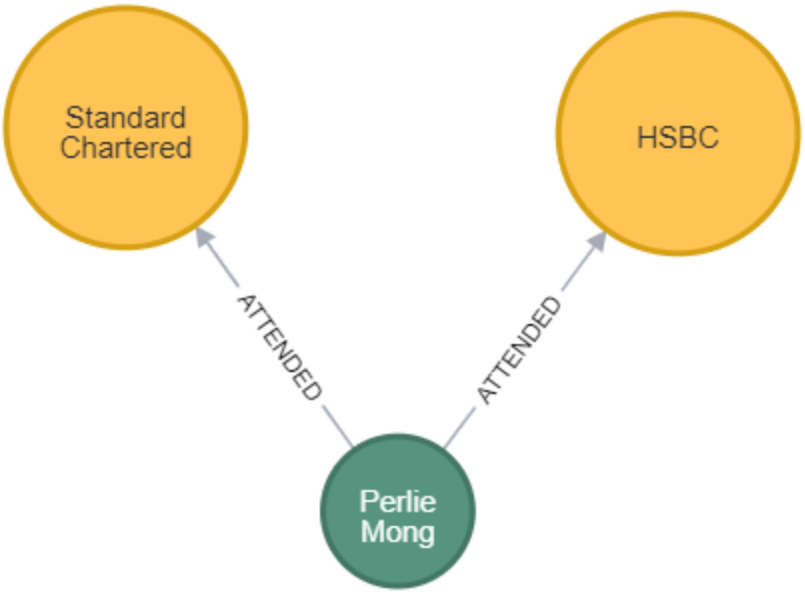
So I tried a more labor intensive approch, which is to prepare a separated Cypher query to answer that specific type of questions. It turns out llama3.1 8B LLM still not able to answer the question. So I used GPT-o1-mini at the end, and here is its answer.

Vector DB RAG	My version of Hybrid RAG
Approximate tokens size: 4006	Approximate tokens size: 300
No one is identified as attending multiple banks' earnings calls in the document.	*Perlie Mong spoke at both Standard Chartered and HSBC earnings calls.*

But I also come with a even more labor intensive approch, which is to prepare a separated Cypher query to answer one specific questions. This time, even llama3.1 8B LLM can answer the question correctly.

Vector DB RAG	My version of Hybrid RAG
Approximate tokens size: 4006	Approximate tokens size: 44

Vector DB RAG	My version of Hybrid RAG
No one is identified as attending multiple banks' earnings calls in the document.	*Perlie Mong attended both HSBC and Standard Chartered's earnings calls events.*



It is a unrealistic method for sure, and I need we need to find a more effective method to solve this task.
It works with or without the "Attended" relationship.

Decisive Differences between VectorDB RAG alone and My version of Hybird RAG (Without separated Cypher query)

When asking questions about analysis of earnings calls when the right chunk of data wasn't selected, Vector DB seems having much more difficulties on the task while My version of Hybrid RAG is doing a much better job.

(The chunk selecting issues can be improve via Reranker training.)

Vector DB RAG	My version of Hybrid RAG
Approximate tokens size: 2456	Approximate tokens size: 1734
Unfortunately, you didn't provide any specific earnings call information or data for HSBC's recent performance. However, I can give you a general	HSBC reported a 12.7 billion profit before tax in Q1 2023 and is maintaining its 2024 financial guidance,

Vector DB RAG	My version of Hybrid RAG
update based on publicly available information. As of my knowledge cutoff (which may not reflect real-time updates), HSBC reported: * Net profit attributable to shareholders: 13.9 billion in 2022, down from 17.1 billion in 2021 due to higher impairments and expenses. * Revenue: 54.7 billion in 2022, a decrease of 4% from 57.1 billion in 2021. Please note that these figures are subject to change and may not reflect the bank's current performance. For more up-to-date information, I recommend checking HSBC's official investor relations website or recent news articles for the latest earnings updates.	with strong first quarter results, including revenue growth to 20.8 billion.

My thought on this difference

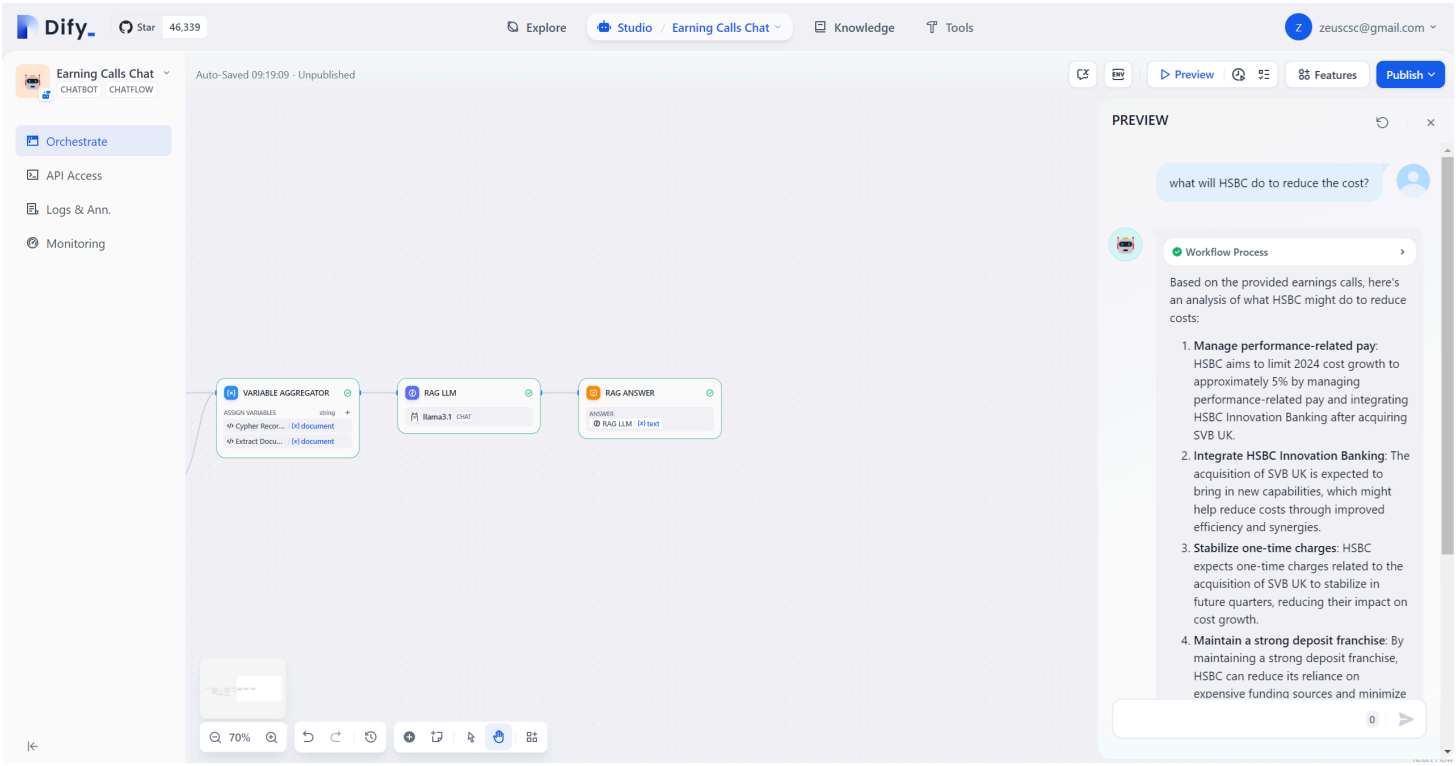
I think this bring out other issues of Vector DB RAG, which is when the query require doing multiple documents analysis, the selected chunks may not be evenly distributed and causing the LLM not able to retrieve required information effective.

For My version of Hybrid RAG, it will be more even since even if the Vector Search is not even, but after pre and post information added to the chunks, the LLM will be able to grasp the whole idea better and do a better job.

POC Development

Although I tried to add the hardcode question into the Vector DB and add its Cypher query results as a RAG document during inferencing, I switch to another approch after some test and thinking. The speed for this approch is faster indeed, but considering future development, adding these kind of hardcode questions into the VectorDB will pollute data, making the whole RAG system less accruate. I think this kind of specific questions need to use at least LLM to analysis the answering decision. That is why I am planning to develop using a LLM flow with an editor.

After the path is set, I created a POC that end-user can interact with.



During the development process, I improved the VectorDB indexing so that it pick the better chunks and use that for pre and post information. Since the previous chunking method with less accruacy will cause multiple pre and post data inserted to the chunks and increased the total tokens usage. The new chunking indexing method will improve the Vector DB querying ability too.
I have also adjusted the Vector Search Documents and Rerank Documents ratio to improvement of the POC.
Despise the above effort, using llama3.1 8B for the POC still pose some challenges.

Here is the video link for the POC:
https://www.youtube.com/watch?v=z5-TGtyC0sw