

## 2025 Intern Self-Review

<b>Intern name</b>	Amit Roy
<b>Intern alias</b>	aamitroy
<b>Last day of internship</b>	08-22-2025
<b>Project title</b>	Natural Language Question Answering on Multiple Spreadsheets via RAG and code-based LLM Agents
<b>Manager name, alias</b>	Piyush Paritosh, ppparito
<b>Mentor name, alias</b>	Zishan Muzeeb, zismuz

### Q1. Project work.

Provide a high-level overview of your project, including goals and key deliverables. Summarize progress toward deliverables and results you've achieved. Include comments on work you have planned for the future when applicable. (200 words maximum)

Q1:

In this project, a set of spreadsheets and a natural language query from these spreadsheets is provided and the task is to answer the query from these spreadsheets. A viable approach to deal with tabular data with LLMs, is to make LLM write a python code and execute it to fetch necessary information to answer the query. Also, it is important to provide the correct spreadsheets/column names from a large number of spreadsheets/columns which will be used as metadata to write the python code by LLM agents. Therefore, similarity matching between the query-spreadsheet or subquery-column name combination can extract the relevant column names necessary for the natural language query. From this intuition, we performed similarity matching between the LLM generated description vectors of query-spreadsheet or subquery-column combination to extract relevant spreadsheets necessary for the python code to answer the query. With experimental analysis, we have shown that for five different datasets in databench benchmark, our proposed cosine similarity based relevant spreadsheet extraction with code writing LLM Agent based approach achieves 86-90% accuracy for natural language query on spreadsheets. In future, updating a given spreadsheet via natural language instruction can be an extension of this project.

### Q2. Skills development.

What skills have you demonstrated proficiency in related to this role (e.g., Excel, writing, new coding languages, data structures, algorithms, code quality, etc.)? What skills are you working to improve? (160 words maximum)

Q2:

To work on Large Language Models (LLMs) for answering natural language query from a set of spreadsheets, I have to demonstrate proficiency in LLMs, LLM Agents, RAGs, Matrix/Vector operations with numpy and pandas in python programming language.

To make LLMs write python code for a natural language query, I utilized the code writing and execution tools with LLMs via *strands* sdk for development of LLM Agents with prompting and workflow design.

Later, to perform cosine similarity between query-spreadsheet or subquery-column description, I obtained the spreadsheet summary from *Anthropic* LLMs via *Bedrock* Client and utilized embeddings from *Titan* model for embedding generation.

Lastly, used pandas for interacting with spreadsheet datasets and matplotlib to visualize the performance of different methods.

### Q3. Actioning feedback.

What feedback did you receive and how did you apply it? (160 words maximum)

Q3:

To deal with structured data like spreadsheets with LLMs, I was provided the feedback to write and execute python code to answer natural language query. Therefore, I explored and learned about strands which allow integration of code writing and execution tools with LLMs and applied them to develop the code writing and execution based LLM Agent pipeline for natural language query in spreadsheets.

To correctly retrieve column names, I was asked to explore RAG and I designed query-spreadsheet/subquery-column similarity matching for relevant spreadsheet retrieval, reducing unnecessary metadata in the prompt of LLM Agent, leading to 80-94% accuracy.

I got the instruction to cover both single spreadsheet and multiple spreadsheet based questions from a set of spreadsheets. Therefore, prepared a dataset of 25 questions from two dataset of databench and performed accuracy and running time analysis.

#### Q4. Dealing with ambiguity.

What ambiguous situations did you run into and how did you adapt? (160 words maximum)

Q4:

Finetuned the performance for different number of spreadsheet to be retrieved, to determine the appropriate number of spreadsheet to be retrieved via query-spreadsheet/subquery-column matching.

For large number of spreadsheets and columns, faced delay in spreadsheet description and embedding generation. To resolve this issue, preprocessed spreadsheet description and embedding generation and load them from cache every time.

For multiple spreadsheet based question answering experiment, designed a mini dataset where questions are from two to three datasets of databench benchmark.

Also, for a large number of LLM calls, incorporated LLMs with cross account access to avoid request time out error.

#### Q5. Leadership Principles.

Select at least 3 Leadership Principles and rate them either as a superpower (strength) or growth area. Provide context and evidence for the rating.

Q5:

Leadership Principle	Strength or Growth Opportunity	Context and Evidence (50 words maximum)
Learn and be curious	Strength	Learned about how code writing and execution based LLM Agents work and curiously explored strands sdk to develop natural language question answering pipeline from multiple spreadsheets.
Invent and Simplify	Strength	Instead of providing the metadata of entire set of spreadsheet as metadata for code generation and execution-based LLM Agents, proposed the approach of query-spreadsheet or subquery-column matching to reduce the number of relevant spreadsheet in metadata.
Think Big	Strength	For large case scenario, preprocessed the spreadsheet description embeddings as cache and used LLMs with cross account access which might enable answering natural language questions at large scale.
Frugality	Strength	Provided a smaller number of spreadsheet as metadata to the LLM Agent prompt for correct column name extraction while writing code for the natural language query and frugal in number of LLM calls by caching the spreadsheet description vectors reducing latency of answering the query.
Dive Deep	Strength	Prepared own dataset for multiple spreadsheets based natural language question answering for performance analysis as well as finetuned with different number of retrieved spreadsheets to pinpoint the number of relevant spreadsheet for best performance in terms of accuracy and running time.
Deliver Results	Strength	Delivered implementation of proposed pipeline for databench dataset, delivered results for five single spreadsheets and multiple spreadsheets of databench with accuracy and running time analysis.
Earn Trust	Strength	To earn the trust of the user on the retrieved answer of the query from given spreadsheets, presented the extracted spreadsheet via the query-spreadsheet or subquery-column matching as well as provide the python code and executed output to show how we attempt to avoid hallucination of LLMs while searching in structured documents like spreadsheets.