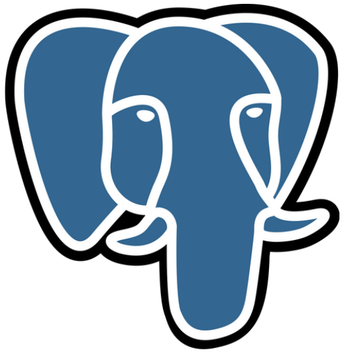
Summer 2023 Internship



CHATTING WITH DATABASES USING LLMS

Henry Zhang

July 2023

Contents

Overview of topics discussed in this presentation

Project Scope

What is the task?
What problem are we trying to solve?

Technologies

Which technologies are being used?

Solutions

Process to solve these problems

Issues and Obstacles

Roadblocks along the process
What are the shortcomings of solutions?

Further improvement

How can we move forward with development?

PROJECT OBJECTIVES

NATURAL LANGUAGE

Chat with a digital assistant about any questions

UNDERSTAND DOCUMENTS

Must read and understand a large corpus of information

ACCURACY

Ensure all information provided is from the documents and not fabricated

IMPROVE EFFICIENCY

Help employees improve efficiency of tasks by having AI parse loads of information

Example Use Cases

WRITE REPORTS

Give me a 200 word summarization of [project] by [company]

COMPARE ACROSS DOCUMENTS

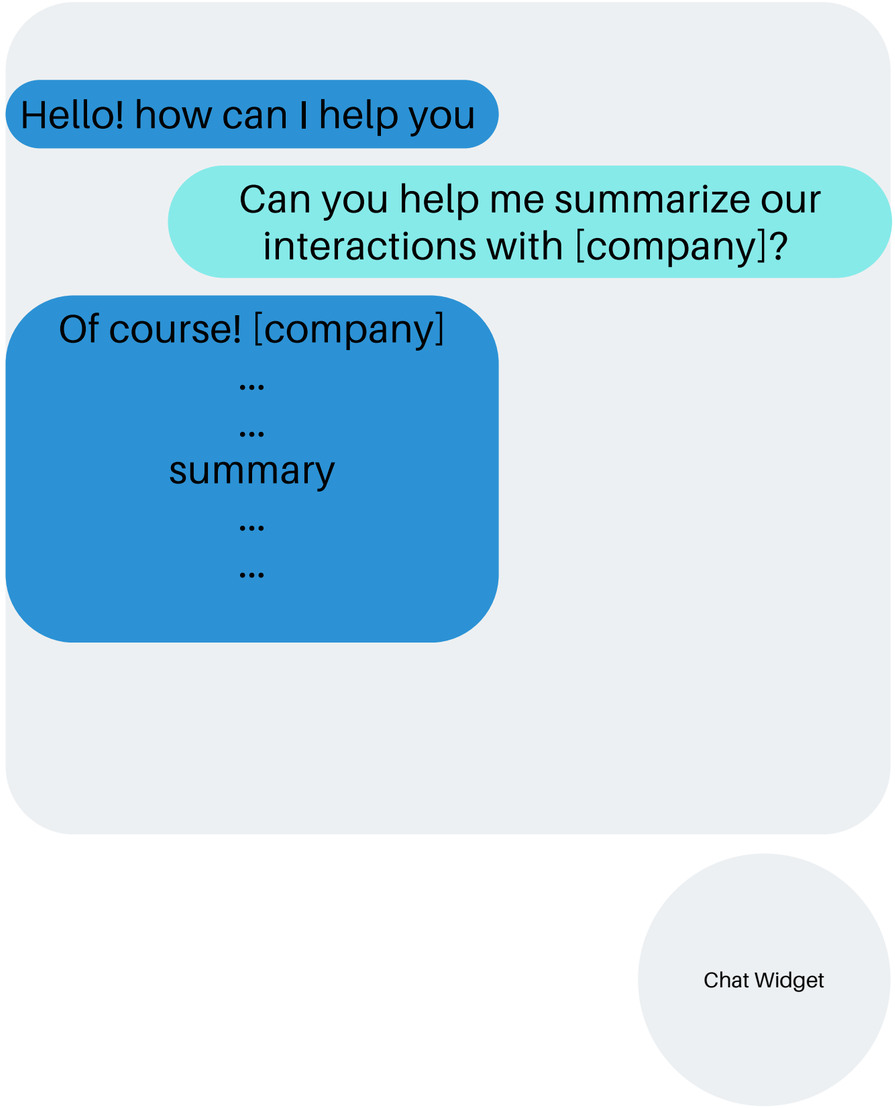
Which products are used most frequently among all clients

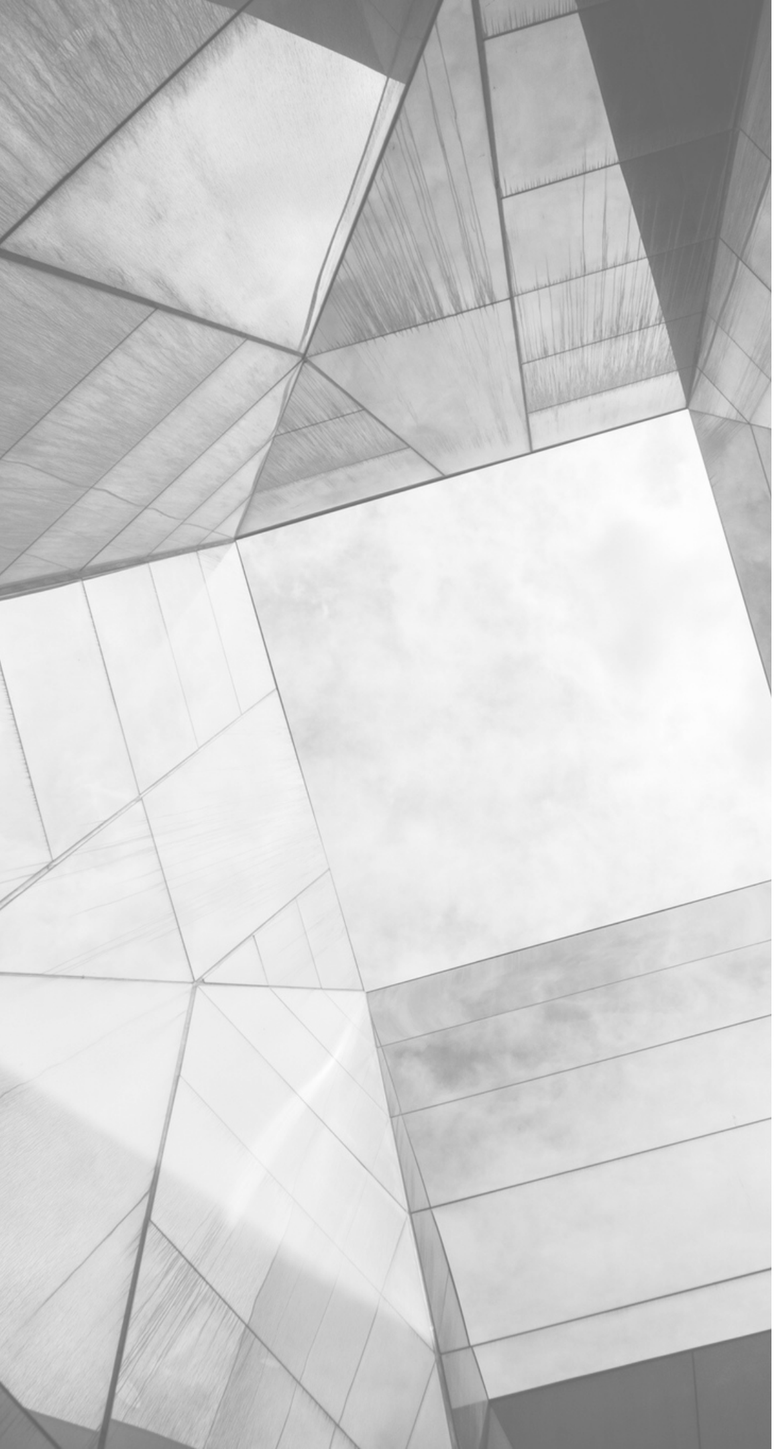
ASK SPECIFIC INFORMATION

What is the exact amount of products to be produced in 2022

Goal

Integrated into company portal





Relevant Technologies

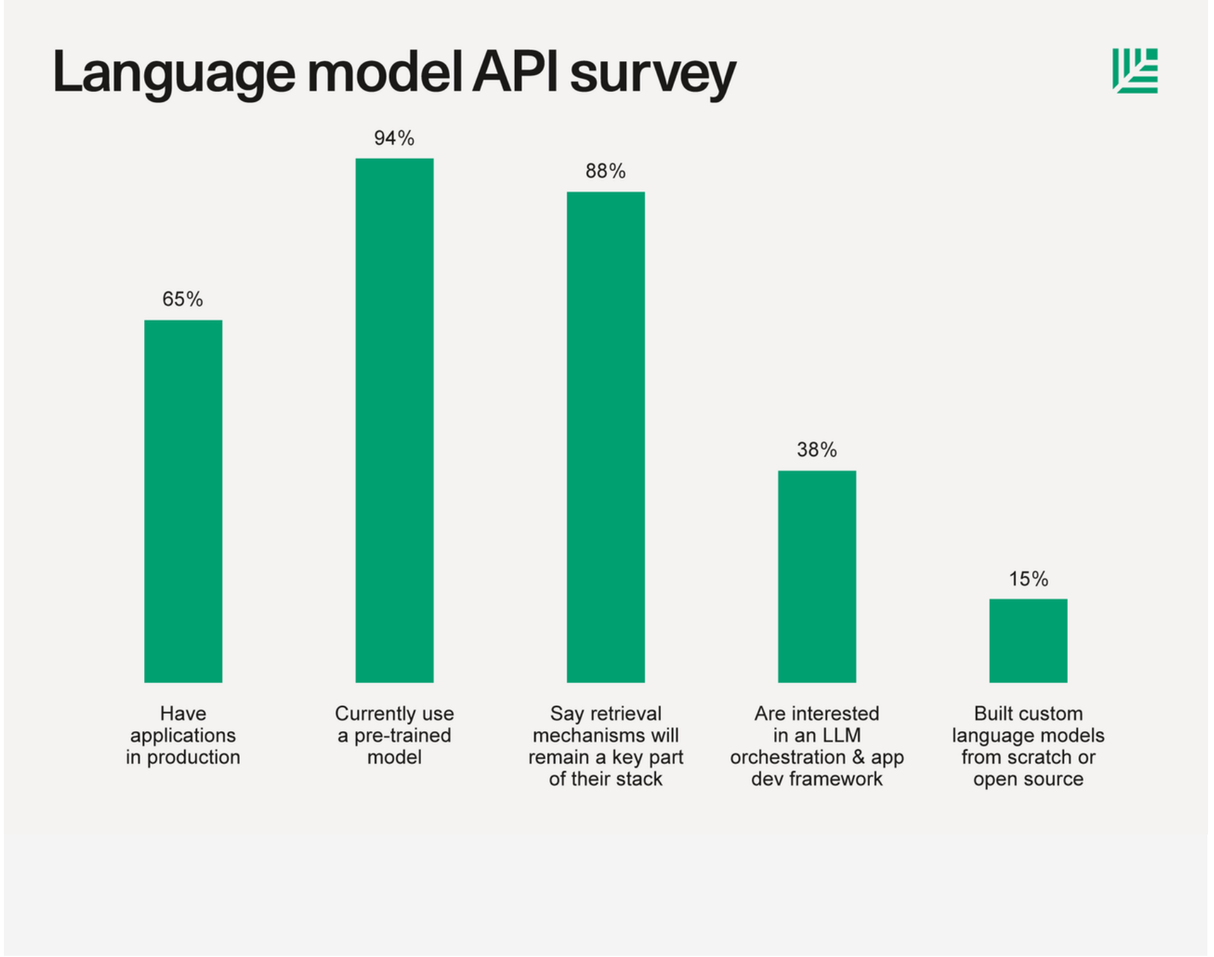
Overview of required tech and background

LLMS AND OPENAI

What is a Large Language Model?

Trained on large amount of parameters

Natural Language, processing, translation, etc



(Fradin, June 2023)

USING LLMS

Enabling Natural Language on our own data:

Training a custom model

Requires ML scientists, data, and infrastructure.

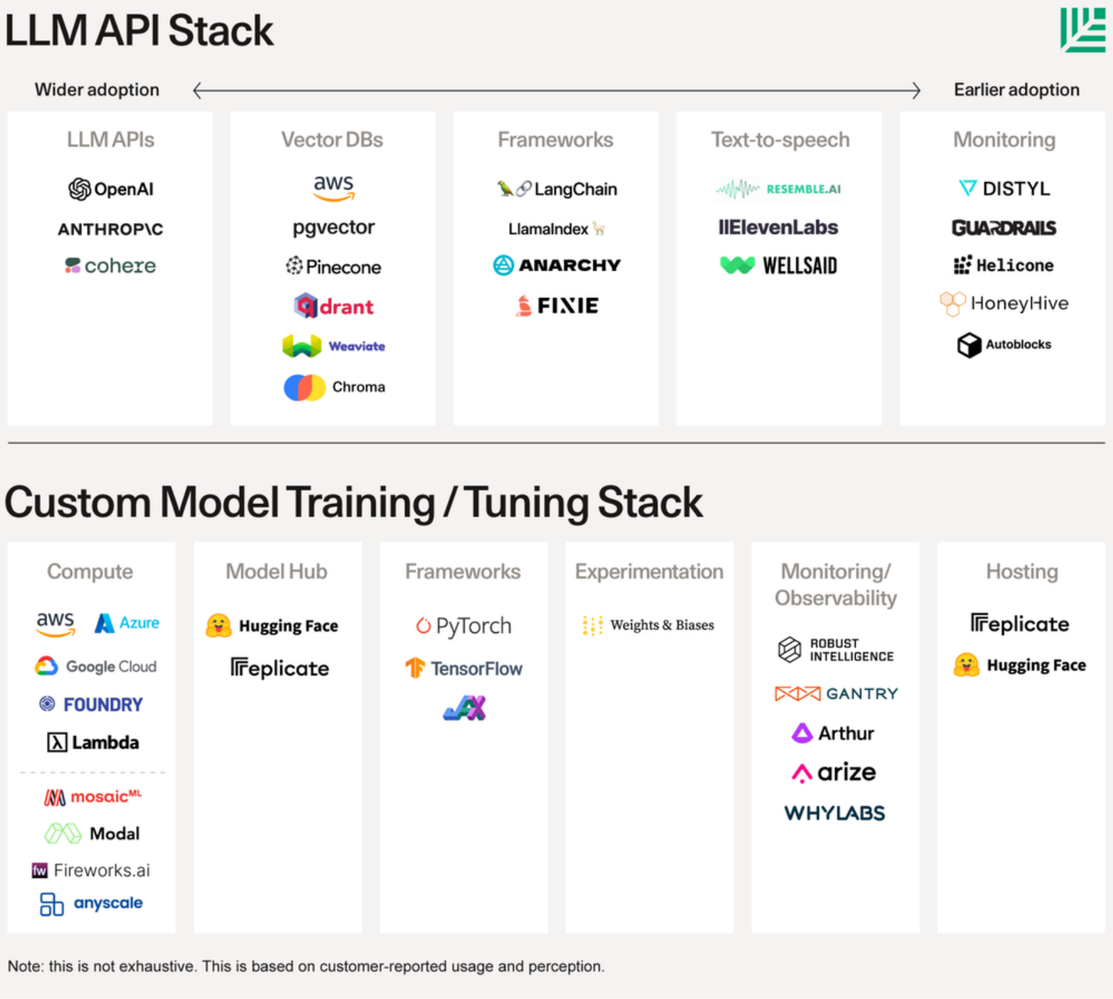
Most difficult way

Fine Tuning a base model

Additional training and updating weights on existing models.

Pre-trained model

Read and return information at the right time



Andrej State of GPT Talk

(Fradin, June 2023)

Langchain

An opensource tool to use LLMS

1

Integration with systems

Connection to data sources such as cloud storage, SQL, Google search provide context

2

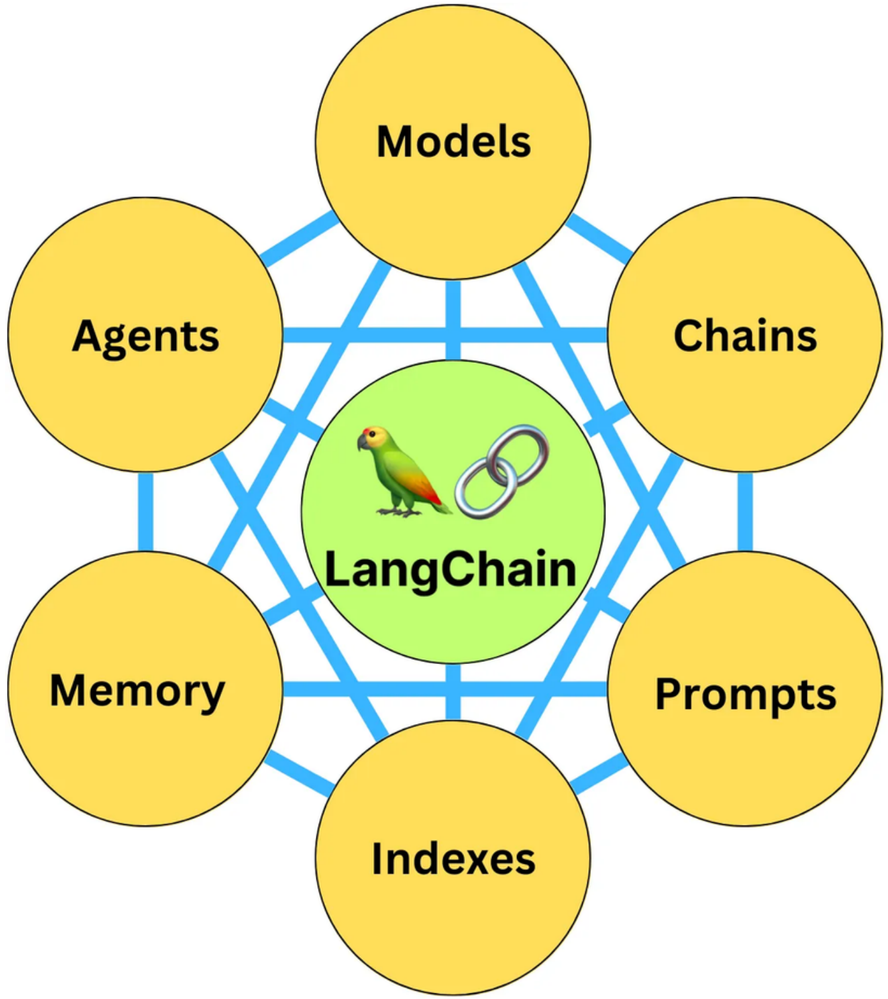
Agents

Can decide what tools to use to increase the compelxity of tasks it can solve

3

Modular components

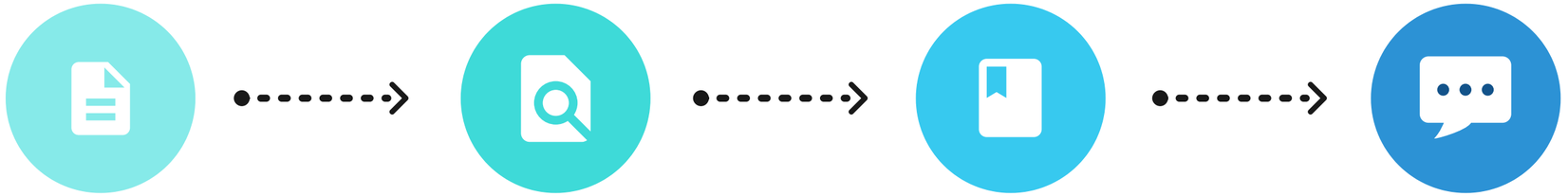
Chains and implementations such as memory, prompts, make it easy to optimize your LLM



(Xu, June 2023).

DIGITAL ASSISTANT

Design Process



1 -Information in Storage

Postgres Database of forms and Company info

2 - Parse or understand Data

Manipulate Data or inform AI to be able to fetch and use Data

AI

LLMs understand natural language input from the user to return the correct queries

4 -UI

Chat interface intergrated into the portal

Solution #1

Using Text to Answer Questions

Embeddings and prewritten Queries

What are Embeddings

Simplifies word representation

Numerical Representation of words that capture the semantic meaning

Measures relatedness

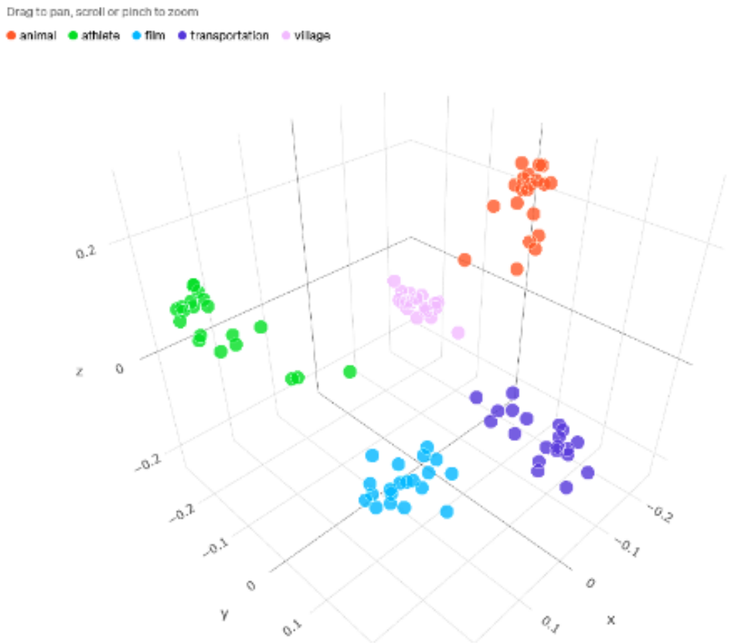
Calulates the relatedness of strings by the distance between vectors

Visualize and draw conclusions

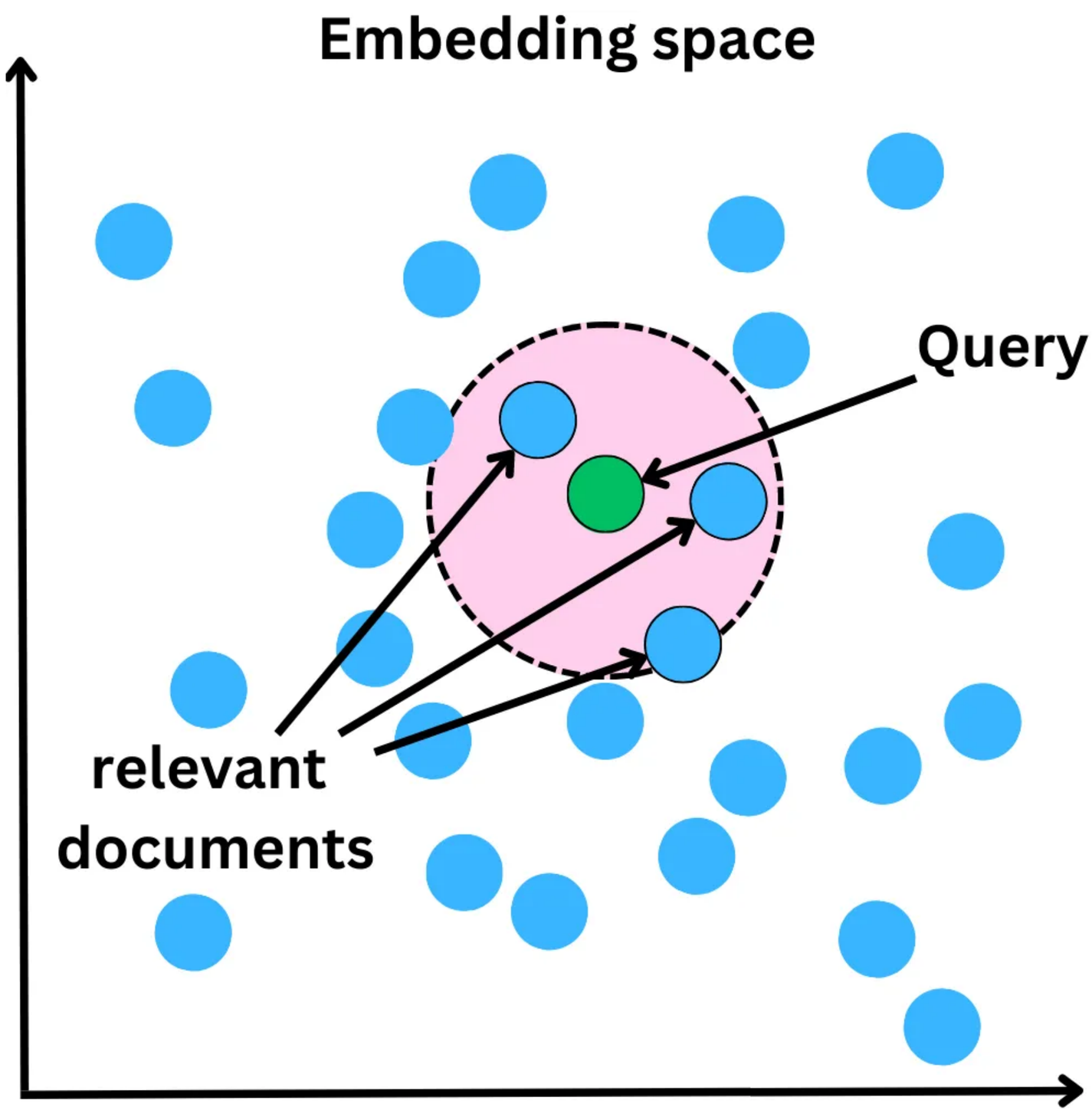
User input can be compared to embeddings to find the most relevent texts

Stored in a vector database

In this case, we are using Redis Vector Store



Semantic and Word Embeddings


Embedding space
Query
relevant
documents

USING EMBEDDINGS TO ANSWER QUESTIONS

Design Process

1 - Convert Database information into documents

Prewritten queries fetch info from the database and convert them into a .txt file to be stored in a blob storage

2 - Add Embeddings

Using the text files , chunk them and add embeddings to them in our vector database

3 - Ask Question

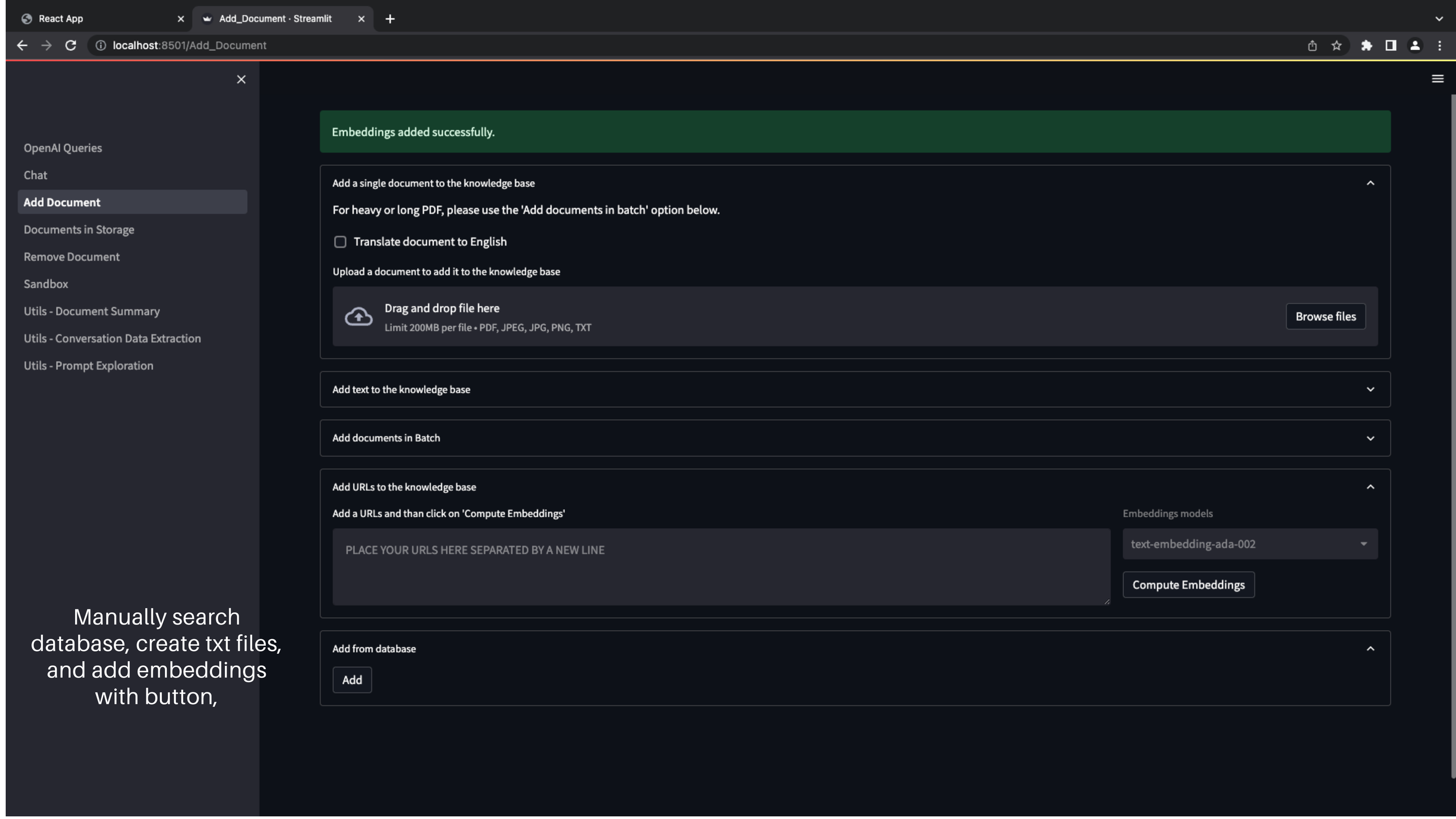
After a question, it searches for relevant answers based on embeddings

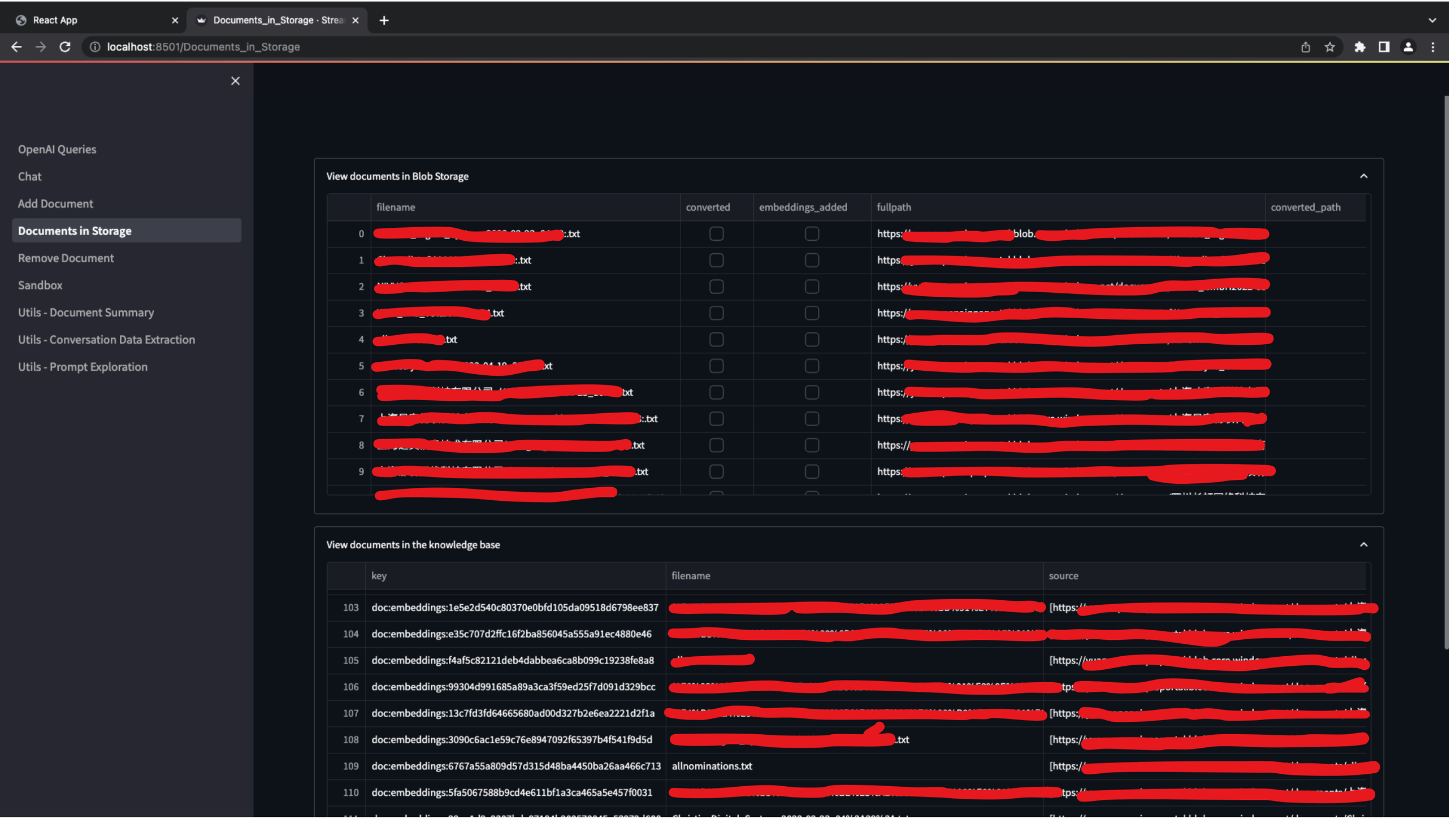
4 - Return answer

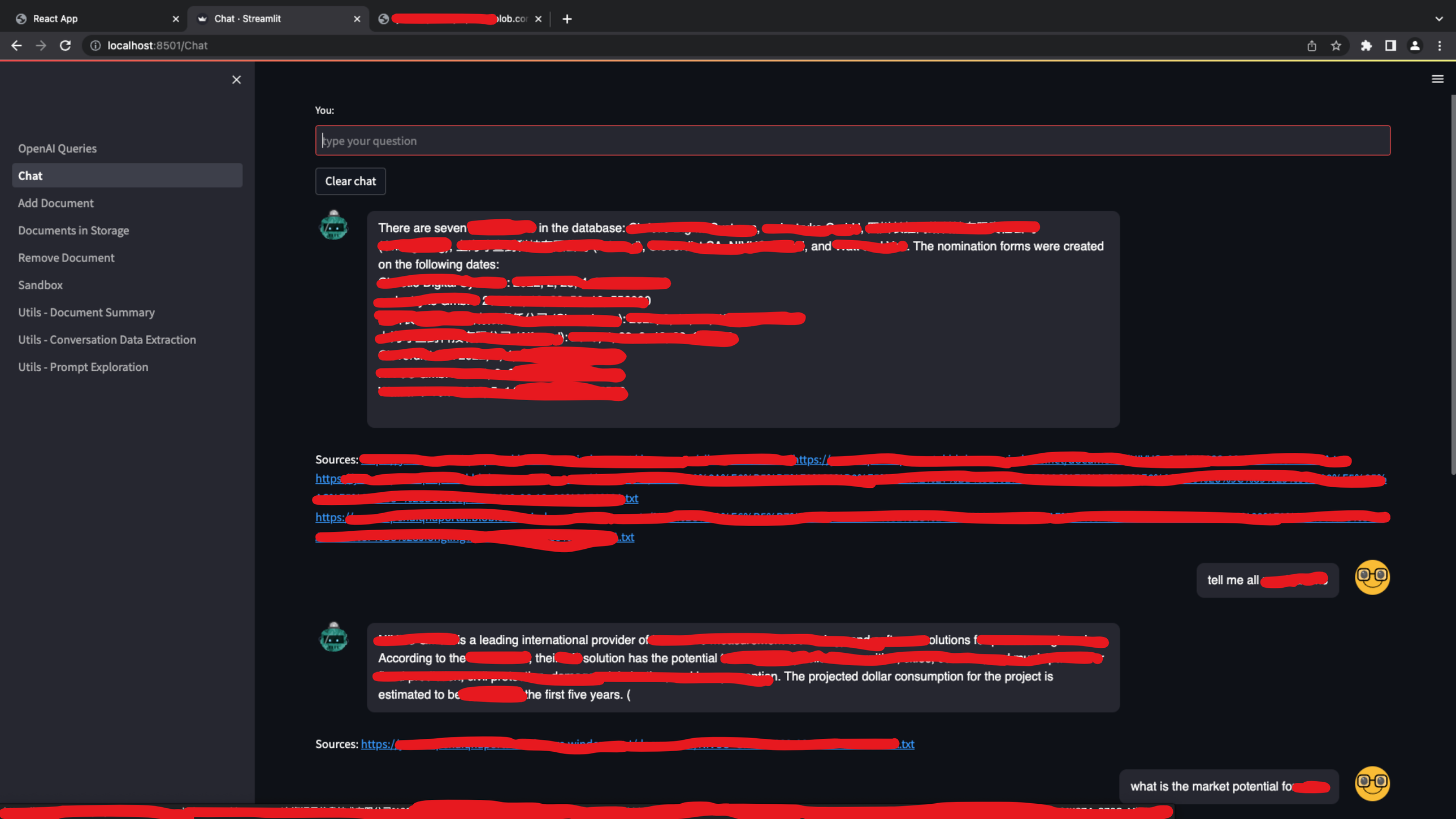
Returns the answer based on vector similarity, and cites the .txt files used

5 - Delivery

Use Stream-lit to return return our answer to the front end.

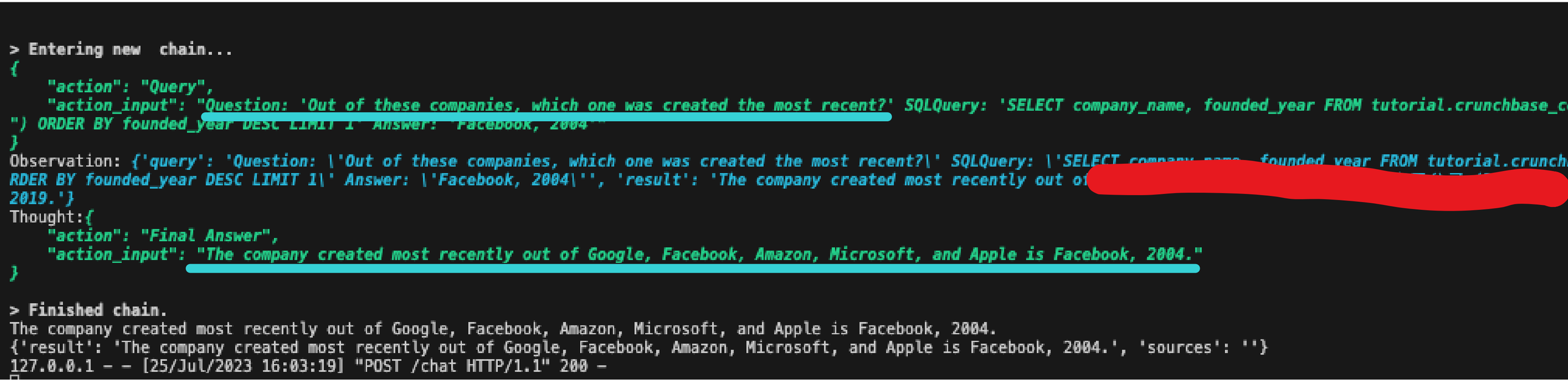

React App
Add_Document · Streamlit
localhost:8501/Add_Document
OpenAI Queries
Chat
Add Document
Documents in Storage
Remove Document
Sandbox
Utils - Document Summary
Utils - Conversation Data Extraction
Utils - Prompt Exploration
Embeddings added successfully.
Add a single document to the knowledge base
For heavy or long PDF, please use the 'Add documents in batch' option below.
Translate document to English
Upload a document to add it to the knowledge base
Drag and drop file here
Limit 200MB per file • PDF, JPEG, JPG, PNG, TXT
Browse files
Add text to the knowledge base
Add documents in Batch
Add URLs to the knowledge base
Add a URLs and than click on 'Compute Embeddings'
PLACE YOUR URLS HERE SEPARATED BY A NEW LINE
Embeddings models
text-embedding-ada-002
Compute Embeddings
Add from database
Add
Manually search database, create txt files, and add embeddings with button,


React App
Documents_in_Storage · Strea
localhost:8501/Documents_in_Storage
OpenAI Queries
Chat
Add Document
Documents in Storage
Remove Document
Sandbox
Utils - Document Summary
Utils - Conversation Data Extraction
Utils - Prompt Exploration
View documents in Blob Storage
filename
converted
embeddings_added
fullpath
converted_path
View documents in the knowledge base
key
filename
source
103 doc:embeddings:1e5e2d540c80370e0bfd105da09518d6798ee837
104 doc:embeddings:e35c707d2ffc16f2ba856045a555a91ec4880e46
105 doc:embeddings:f4af5c82121deb4dabbea6ca8b099c19238fe8a8
106 doc:embeddings:99304d991685a89a3ca3f59ed25f7d091d329bcc
107 doc:embeddings:13c7fd3fd64665680ad00d327b2e6ea2221d2f1a
108 doc:embeddings:3090c6ac1e59c76e8947092f65397b4f541f9d5d
109 doc:embeddings:6767a55a809d57d315d48ba4450ba26aa466c713 allnominations.txt
110 doc:embeddings:5fa5067588b9cd4e611bf1a3ca465a5e457f0031


React App
Chat · Streamlit
localhost:8501/Chat
OpenAI Queries
Chat
Add Document
Documents in Storage
Remove Document
Sandbox
Utils - Document Summary
Utils - Conversation Data Extraction
Utils - Prompt Exploration
You:
type your question
Clear chat
There are seven [redacted] in the database: [redacted]. The nomination forms were created on the following dates:
[redacted]
Sources: [redacted]
tell me all [redacted]
[redacted] is a leading international provider of [redacted] solutions f[redacted]. According to the [redacted], thei[redacted] solution has the potential [redacted]. The projected dollar consumption for the project is estimated to be [redacted] the first five years. (
Sources: [redacted]
what is the market potential fo[redacted]

Prompt Engineering

A common issue with GPT is its imaginative answers



Prompt Engineering

```
template = """{summaries}
Please reply to the question in as much detail using only the information present in the text above.
[Specific description of database and what exactly it is used for ]
Include references to the sources you used to create the answer if those are relevant ("SOURCES").
If you can't find it, reply politely that the information is not in the knowledge base and respond with information in the knowledge base that could be relevant to the question. If asking a clarifying question to the user would help, ask the question.
The question will also contain the corresponding SQL query and result. If applicable, Give an answer based on the query result
Question: {question}
Answer:"""

PROMPT = PromptTemplate(template=template, input_variables=["summaries", "question"])
```

OpenAI Queries
Chat
Add Document
Documents in Storage
Remove Document
Sandbox
Utils - Document Summary
Utils - Conversation Data Extraction
Utils - Prompt Exploration

You:

type your question

Clear chat

[redacted] is the most commonly used Microsoft product by companies, as it allows them to leverage Microsoft AI capabilities such as [redacted] and more. It also allows companies to use [redacted] Additionally, companies can utilize [redacted] in various ways, such as [redacted] connected to [redacted] and [redacted] and providing APIs for the connection to 3rd party applications [redacted]

Sources: [redacted] A.txt

which microsoft products are used the most by all companies

Limitations

- CANNOT ANSWER COMPLEX QUERIES
- STORAGE SPACE
- NOT UPDATED WITH THE DATABASE
- MUST MANUALLY CREATE .TXT QUERIES

Solution #2

Querying the database Directly

Connect to database using langchain utilities

QUERYING THE DATABASE FOR ANSWERS

Design Process

1 - Connect to database

Connect to the database directly and provide specific tables for Langchain to use

2 -Langchain

Langchain provides the capability to connect the LLM to the tables provided with SQLDatabase Chain

3 - Ask Question

After a question, the LLM generates a query to answer the question and runs the database

4 - Return answer

Returns the answer based on query result, and cites the SQL query used

5 - Delivery*

Use Stream-lit to return return our answer to the front end.

*front end implemented with React Later

How SQLDatabaseChain works

Still in its early stages of development

Chains Prompts together

SQL Generation

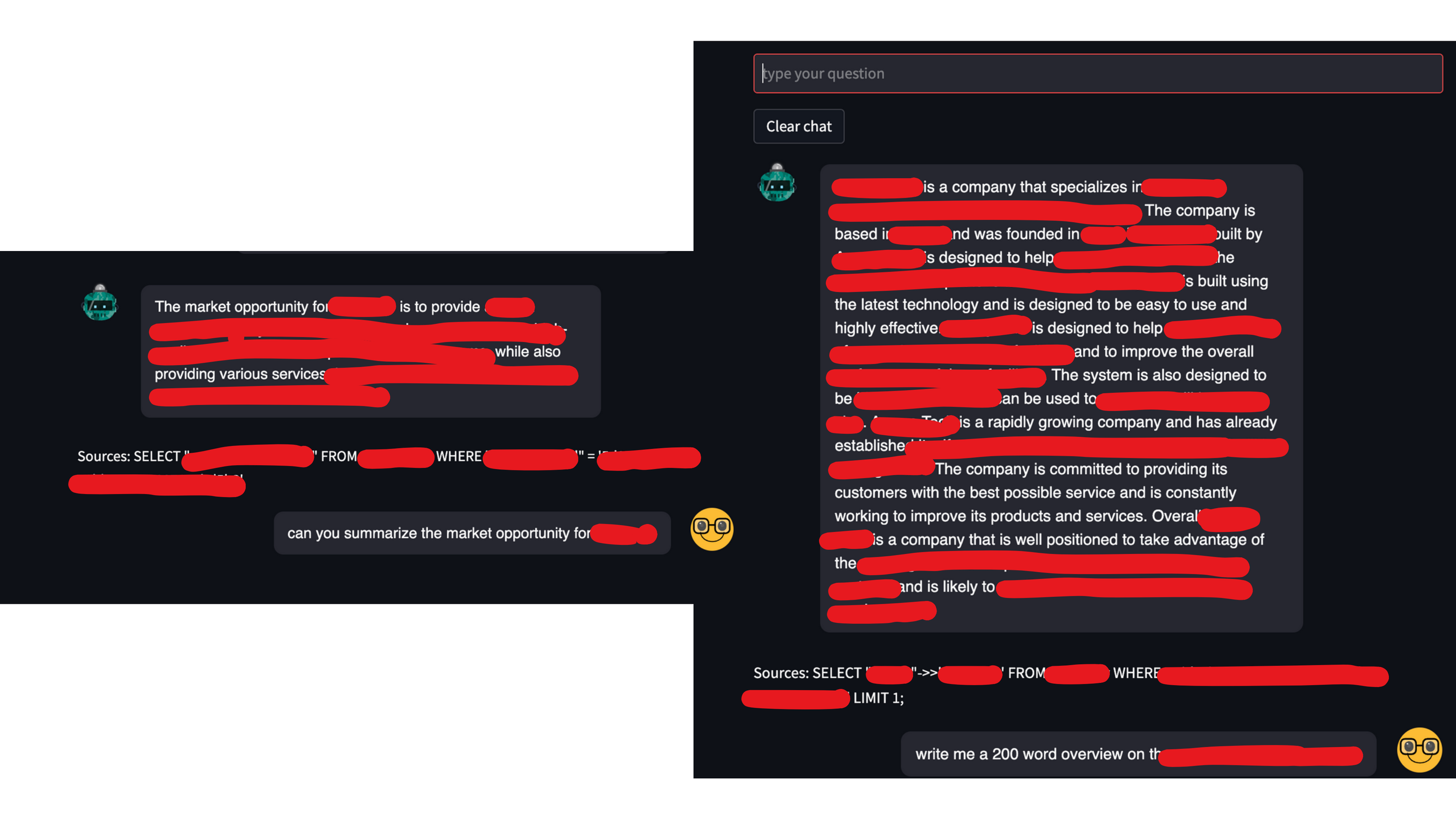
```
llm_inputs = {
    "input": input_text,
    "top_k": str(self.top_k),
    "dialect": self.database.dialect,
    "table_info": table_info,
    "stop": ["\nSQLResult:"],
}
```

Runs SQL statement

```
result = self.database.run(sql_cmd)
```

Converts into NL answer

```
db = SQLDatabase.from_uri(
        con,
        include_tables=['[example_table1]', '[example_table2]', '[example_table3]']
        )
    db_chain = SQLDatabaseChain.from_llm(llm, db, verbose= False, use_query_checker = False, return_intermediate_steps= True, top_k=20)
    result = db_chain(question)
```


type your question
Clear chat
is a company that specializes in
The company is
based in
and was founded in
built by
is designed to help
the
is built using
the latest technology and is designed to be easy to use and
highly effective
is designed to help
and to improve the overall
The system is also designed to
be
can be used to
is a rapidly growing company and has already
establishe
The company is committed to providing its
customers with the best possible service and is constantly
working to improve its products and services. Overall
is a company that is well positioned to take advantage of
the
and is likely to
Sources: SELECT
FROM
WHERE
LIMIT 1;
write me a 200 word overview on th
The market opportunity for
is to provide
while also
providing various services
Sources: SELECT
FROM
WHERE
can you summarize the market opportunity for

OpenAI Queries
Chat
Add Document
Documents in Storage
Remove Document
Sandbox
Utils - Document Summary
Utils - Conversation Data Extraction
Utils - Prompt Exploration

You:

type your question

Clear chat

The companies that created [redacted] in 2019 are:
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted])
- [redacted]

Sources: SELECT [redacted] FROM [redacted] JOIN [redacted] ON [redacted] WHERE [redacted] >= '2019-01-01' AND [redacted] < '2020-01-01' ORDER BY [redacted] DESC LIMIT 20;

which companies created [redacted] in 2019

There were [redacted] created in 2019.

Sources: SELECT [redacted] FROM [redacted] WHERE [redacted] >= '2019-01-01' AND [redacted] < '2020-01-01'

which [redacted] were created in 2019

OpenAI Queries
Chat
Add Document
Documents in Storage
Remove Document
Sandbox
Utils - Document Summary
Utils - Conversation Data Extraction
Utils - Prompt Exploration

You:

type your question

Clear chat

The Microsoft products used across all nominations are: [redacted]

Sources: SELECT DISTINCT UNNEST([redacted]) AS "Microsoft Products" FROM [redacted]

tell me all Microsoft products used across all [redacted]

Limitations

- SQLDATABASECHAIN DOES NOT HAVE MEMORY YET
- ONLY KNOWS HOW TO TAKE QUERY RELATED QUESTIONS
- NARROW SCOPE FOR ANSWERS
- INCORRECT QUERIES

ONLY KNOWS HOW TO TAKE QUERIES

Add Document

Documents in Storage

Remove Document

Sandbox

Utils - Document Summary

Utils - Conversation Data Extraction

Utils - Prompt Exploration

Clear chat

The names and URLs of the first 5 companies in the database are:

- [redacted]

Sources: SELECT "[redacted]" FROM [redacted] LIMIT 5;

Hello chat bot!

The most recent company was created on [redacted]

Sources: SELECT [redacted] FROM [redacted] ORDER BY [redacted] DESC LIMIT 1;

Hello agent

NARROW SCOPE FOR ANSWERS

- Add Document
- Documents in Storage
- Remove Document
- Sandbox
- Utils - Document Summary
- Utils - Conversation Data Extraction
- Utils - Prompt Exploration

stion

Clear chat

There are [redacted] that use the [redacted], and [redacted]

Sources: SELECT "company".name FROM "company" INNER JOIN "nomination" ON "company".uuid = "nomination"."companyUuid" WHERE 'iot-hub' = ANY("nomination"."msftProducts") LIMIT 20;

can you tell me al[redacted] that use the [redacted]

There are n[redacted] that use[redacted] in their solution according to the given tables.

Sources: SELECT [redacted]

can you tell me al[redacted] that use[redacted] in their solution

Incorrect Queries

ALWAYS A POSSIBLILTY

OPENAI is never 100 Percent accurate

POSTGRESQL SYNTAX

In our case, Postgres case sensitivity with quotations is a recurring issue, despite prompt modifications

INCONSISTENCIES AND TIMEOUTS

Query checkers make the same mistake
Inconsistent sucess rates make it hard to reproduce

The same query, run minutes apart: What companies created [sample] in 2020
s."sampleID" vs s.sampleID

There were no companies that competed [sample] on 2020.
{'result': 'There were no companies that competed [sample] on 2020.', 'sources': 'SELECT DISTINCT c.name\nFROM company c\nJOIN [sample] s ON c.id = s.***"sampleID"***\nWHERE [dateColumn] >= \'2020-01-01\' AND [dateColumn] <= \'2020-12-31\'\nORDER BY c.name\nLIMIT 20;'}

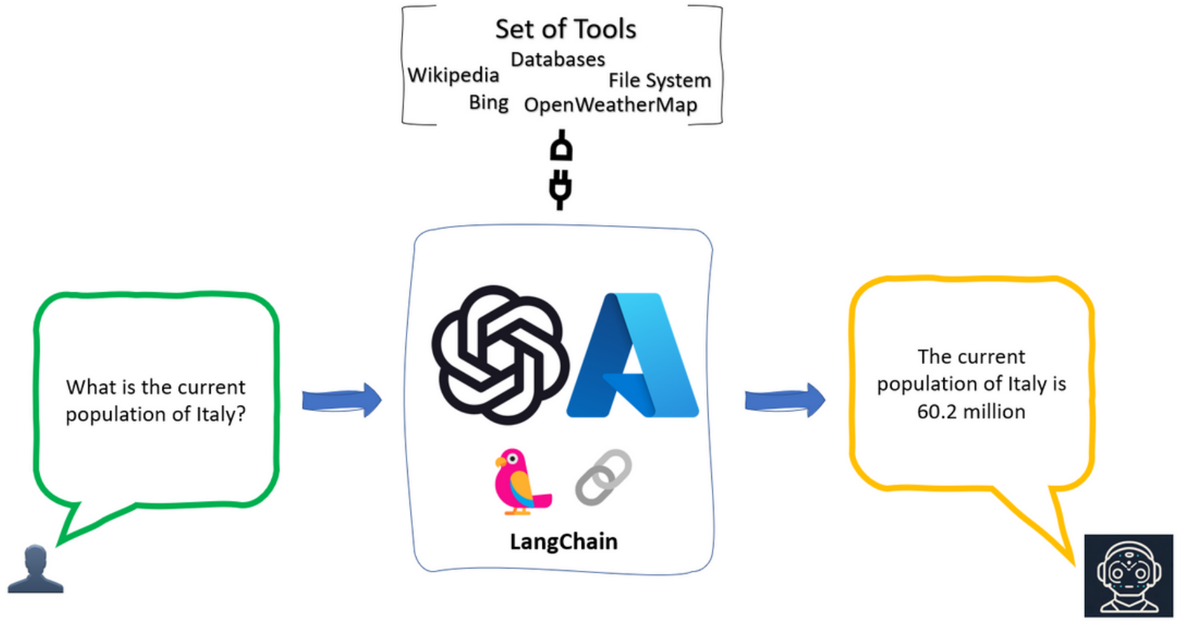
sqlalchemy.exc.ProgrammingError: (psycopg2.errors.UndefinedColumn) column s.sampleID does not exist
LINE 3: INNER JOIN sample n ON n ON c.uuid = c.uuid = s.sampleID
^
HINT: Perhaps you meant to reference the column "s.sampleID".
[SQL: SELECT DISTINCT c.name
FROM company c
INNER JOIN sample s ON c.id =s.sampleID
WHERE EXTRACT(YEAR FROM [dateColumn]) = 2020
ORDER BY c.name
LIMIT 20;]

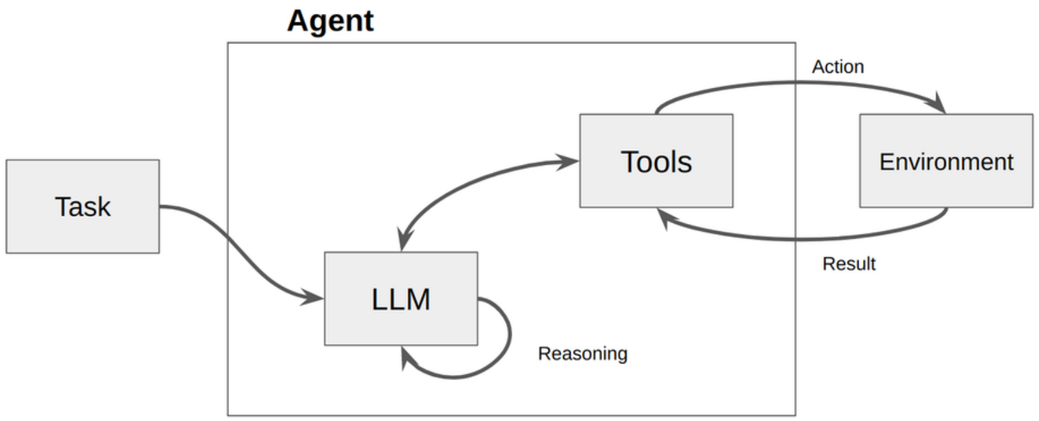
Solution #2.1

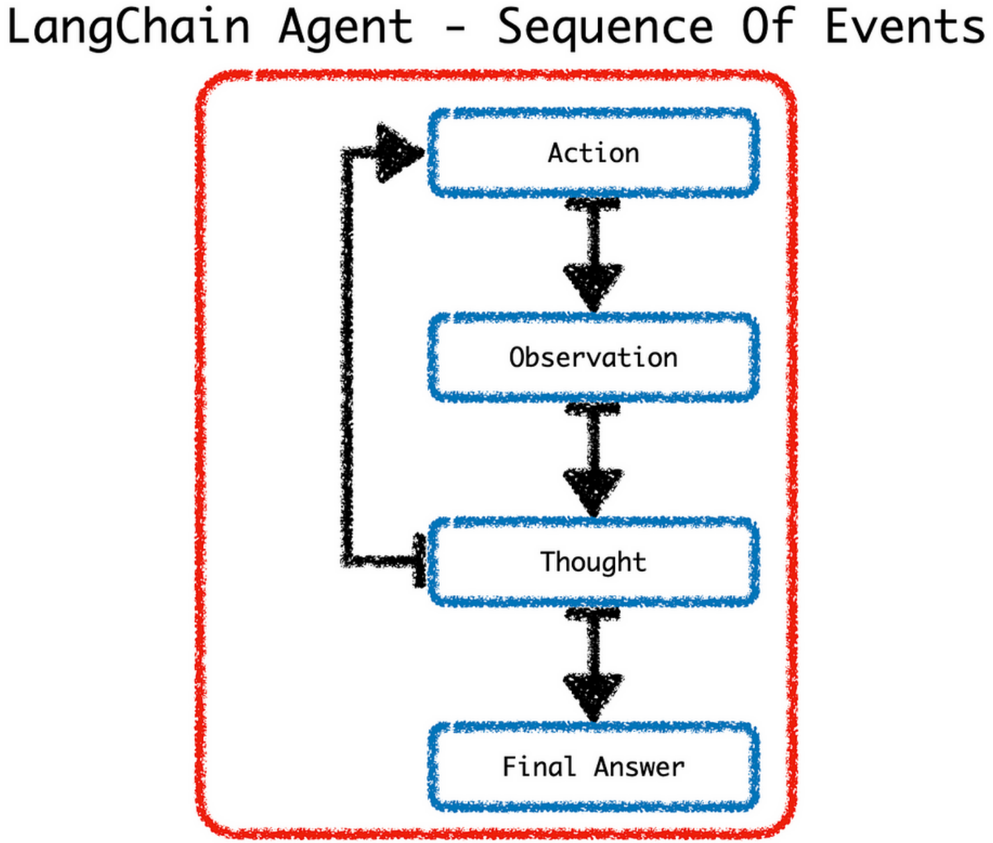
Building on and fixes to Solution #2

Using Agents, ReactJS

What are Agents?







IMPROVEMENTS

Small improvements

AGENTS ADD FLEXIBILITY

Now with an Agent, it can process and respond to non Queries

ADDED MEMORY

While adding memory to SQLDatabaseChain is not possible yet, adding memory to an Agent is doable

MORE AREA FOR IMPROVEMENT

Custom agents allow us to add more tools in the future

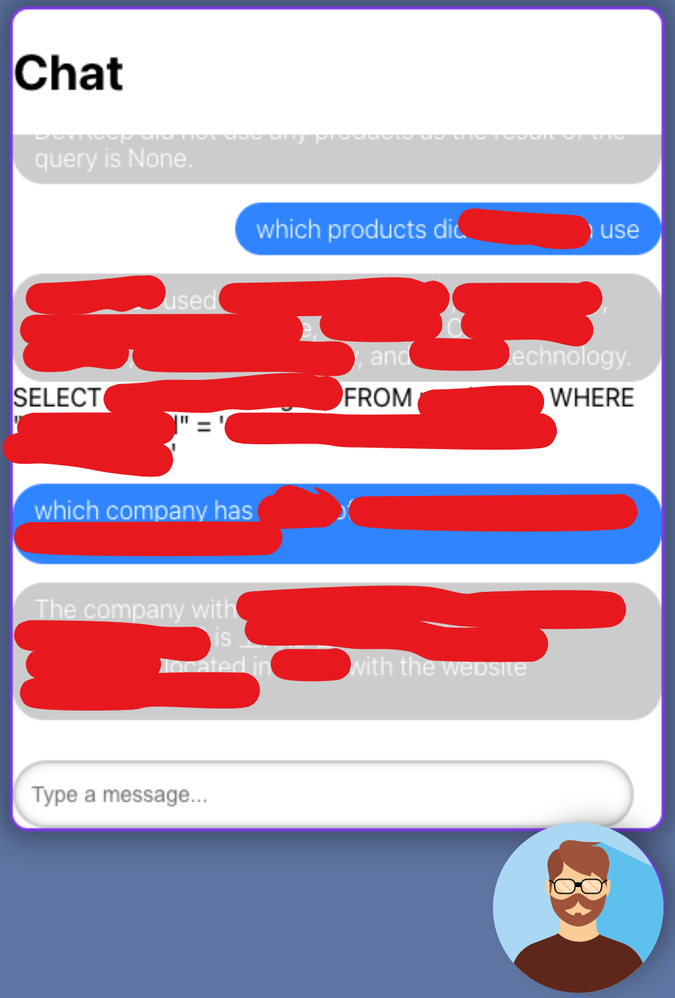
REACTJS FRONTEND

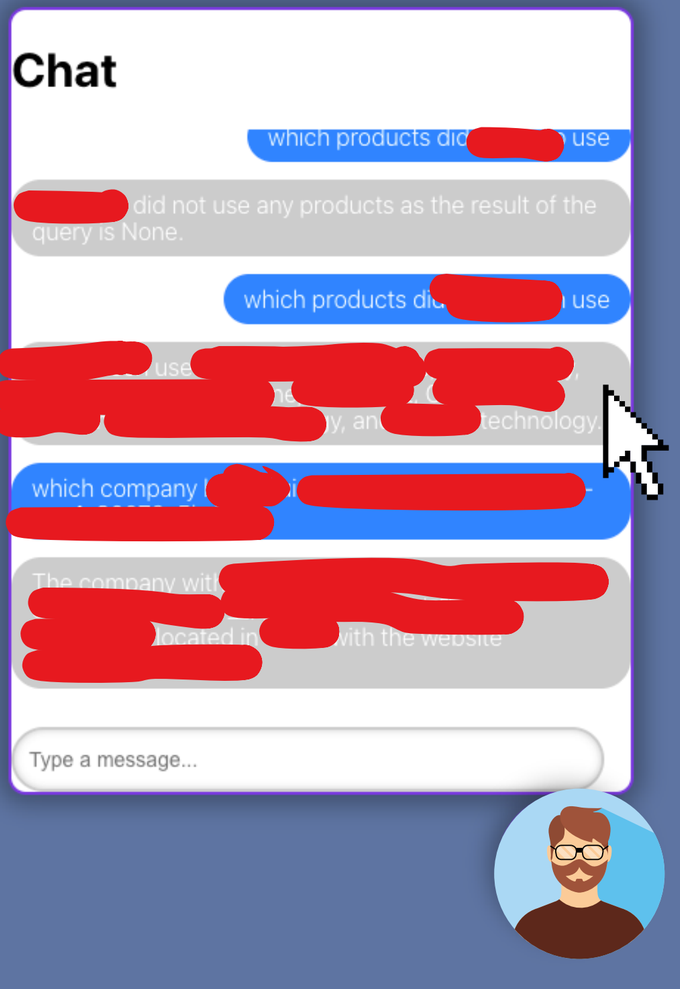
Using Flask in Python as a REST API, it connects to a simple widget component in React

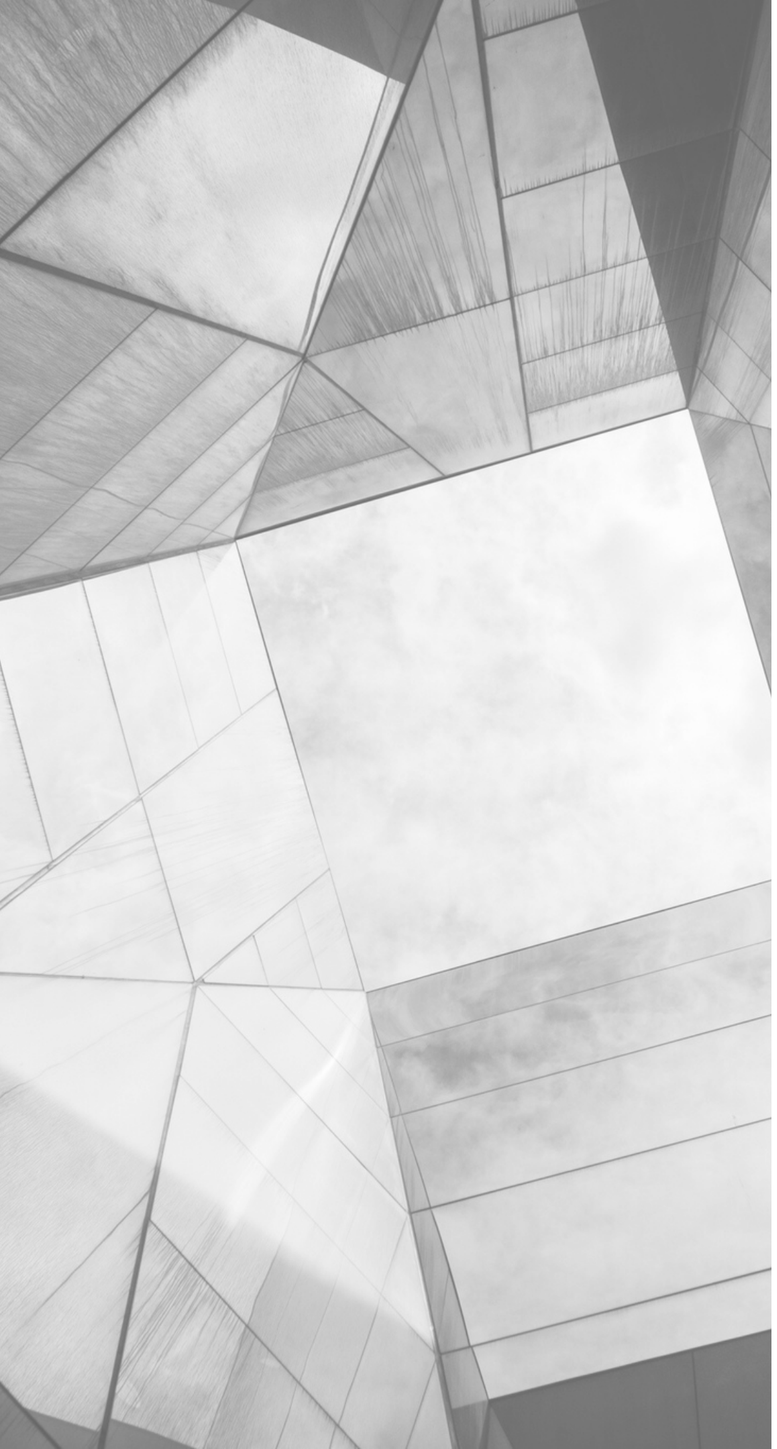
EXAMPLE

Enter Task Add

Toggleable Widget Component







Further Improvements

The next steps to take in improving and implementing this application

FURTHER IMPROVMENTS

Additional things unable to be implemented due to time constraints

IMPROVE QUERY RELIABLITY

Add chain of llm prompts to check queries to return less errors.

CUSTOMIZE PROMPTS

Edit prompts to eliminate imagination

MORE FUNCTIONALITIES

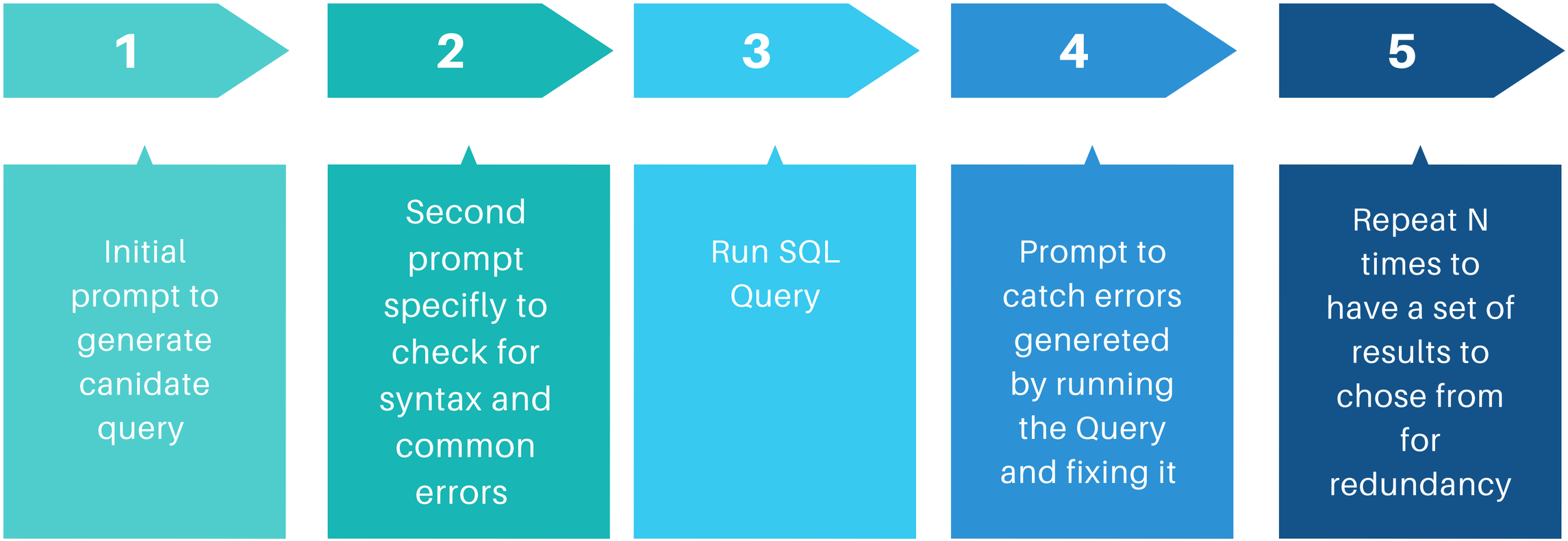
Add more tools to the Agent, such as another chat bot, formatting

REFINE PRODUCT

Improve UI, response time, and overall code

MAKING THE SQL GENERATION ERROR PROOF

Chaining specific prompts



Haren, January 2023

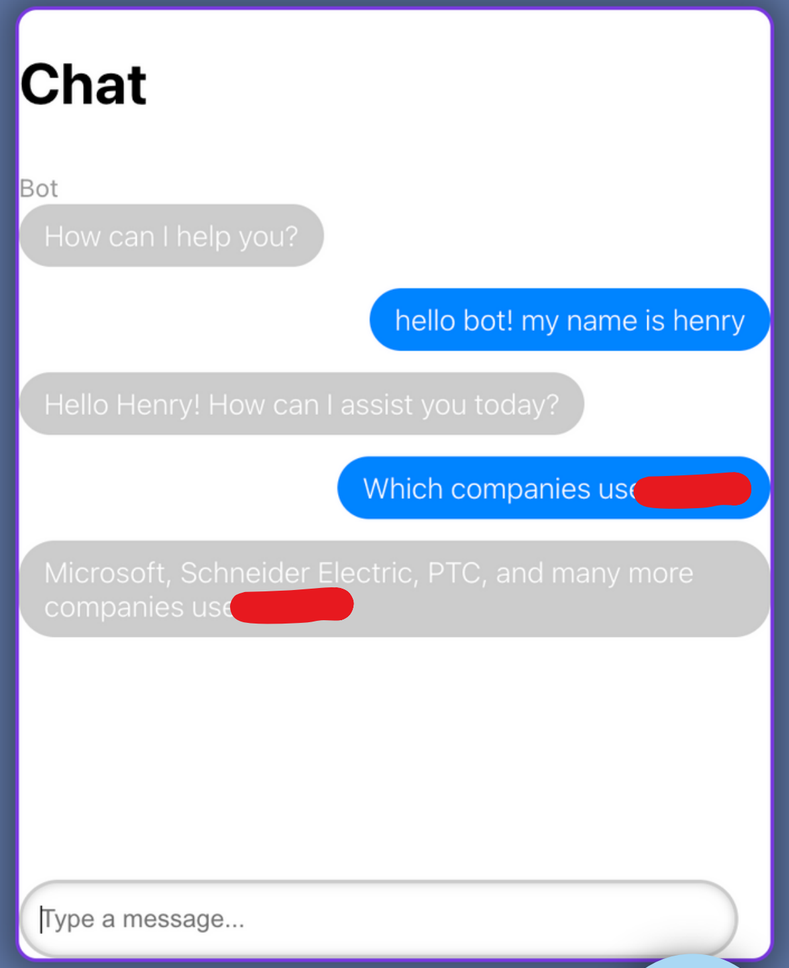
EXAMPLE

Enter Task

Add

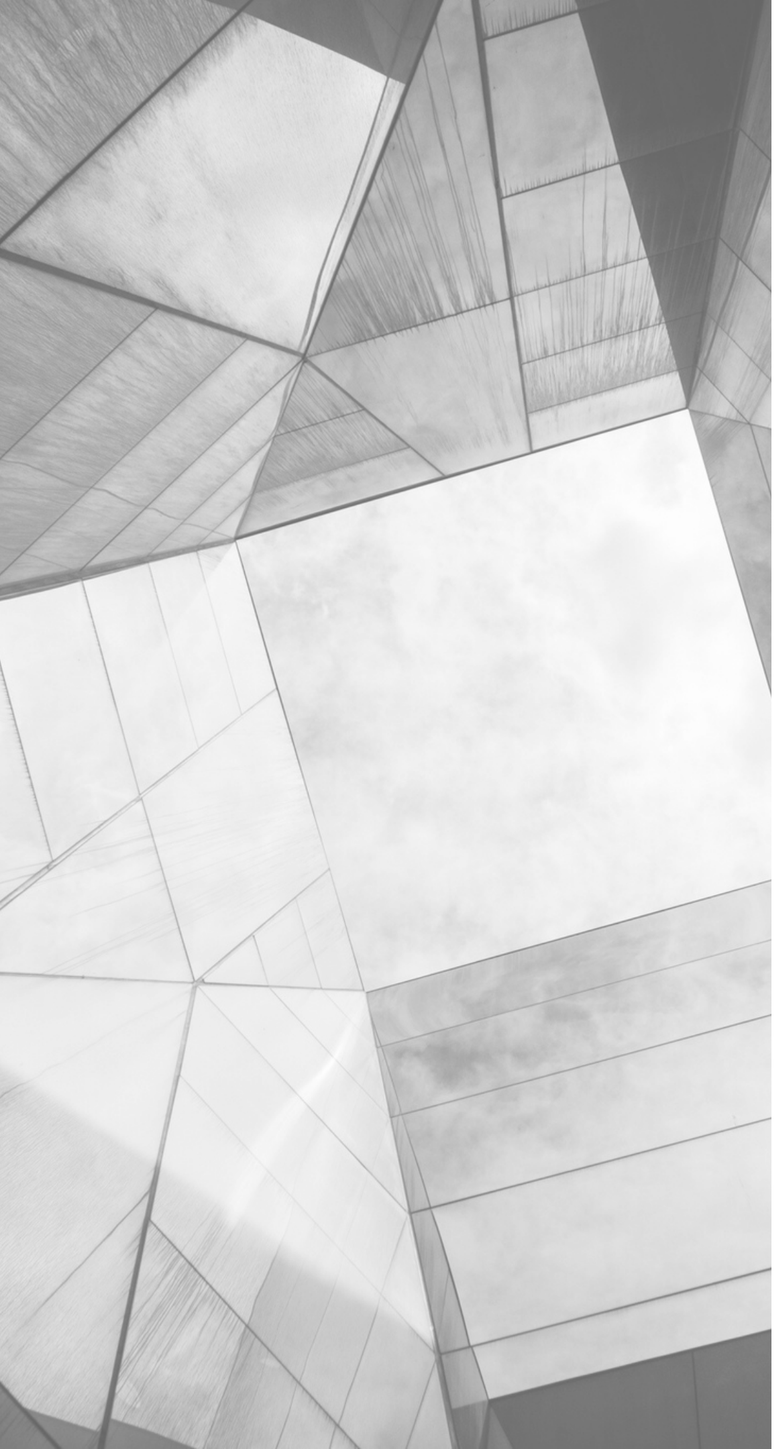
Agent Prompts not yet modified

Hallucinations can easily be fixed with custom prompts mentioned earlier



Other optimizations and improvements

1	**IMPROVE UI**	Refine the UI to look more sleek, and with buttons such as "try again"
2	**ADD STREAMING**	Display text as it's being generated to increase transparency and response time
3	**INTERMEDIATE STEPS**	Returned sources are not yet implemented into the Agent solution
4	**AGENT TOOLS**	Add more agent tools to increase functionality. Ex. formating tools
5	**OPTIMIZE TOKEN USAGE**	Many different ways to reduce tokens, such as a chatbot tool



Takeaways and Conclusion

What was learned and accomplished over the course of this project

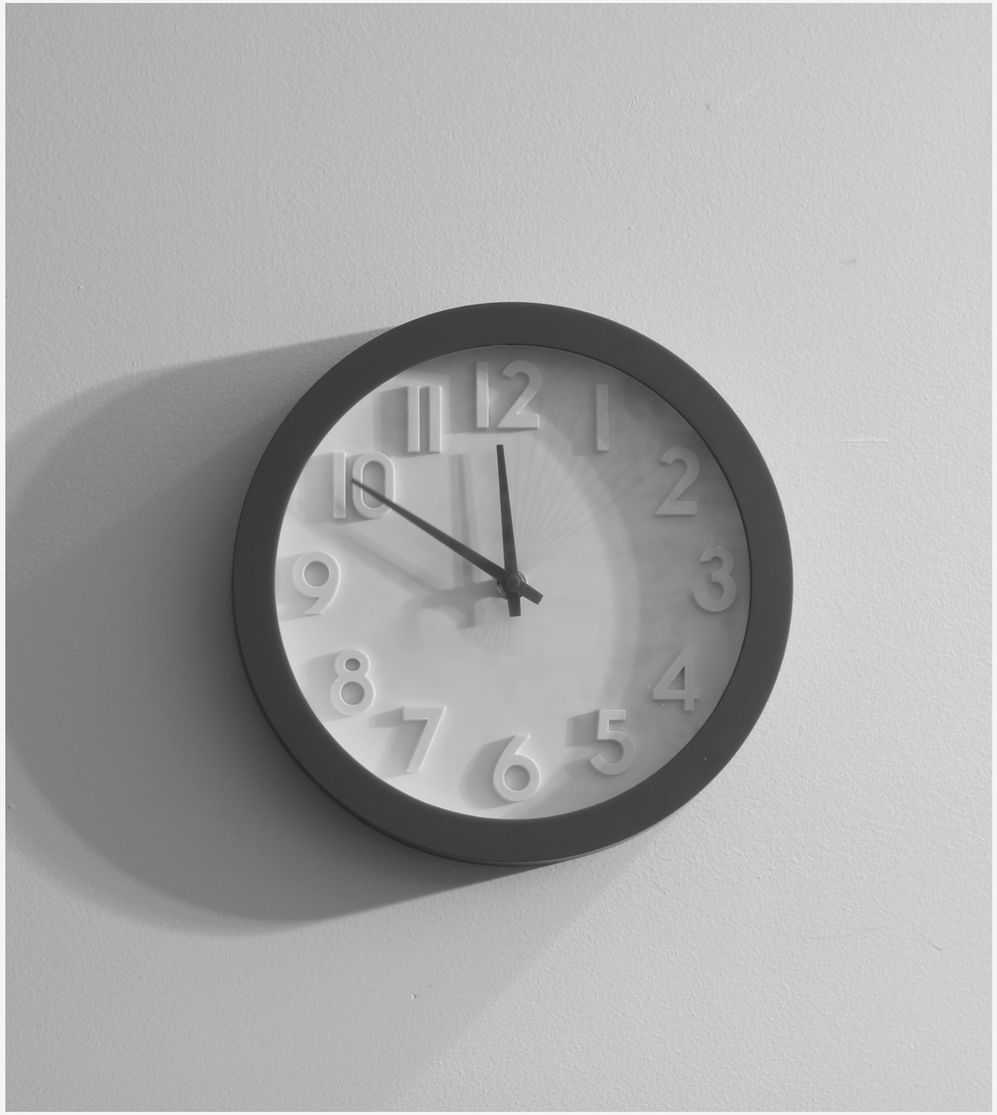
OBJECTIVE RECAP

Explored options and sucessfully communicated with our data

Determined that a Digital assistant to improve efficiency is very feasible

Understood strenghts and limitations

Learned lessons and recomendations for future development



EMBEDDINGS

Able to find relevant information easily

Provides sources, and good summaries and overviews of single nominations

QUERY WITH CHAIN AND AGENT

Can answer complex questions regarding different companies

Many potential areas of improvement

Conclusion

LEARNED RELEVANT SKILLS

Ability to communicate with data using Langchain and LLMS

WORKING STARTING POINT

Solid foundation to refine and improve on

CLEAR NEXT STEPS

Clear areas of improvement to work for

Thank You

Any Questions?

Special thanks to Yuan and Anton for helping along the process