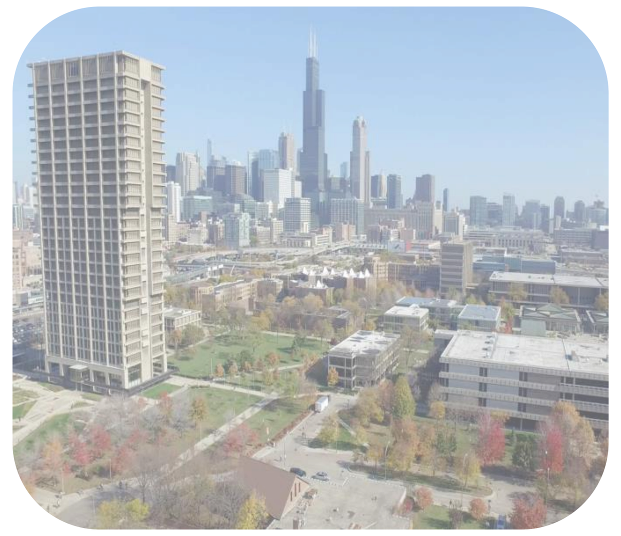


LLM-POWERED OLAP: TACKLING HIGH LEVEL BUSINESS QUESTIONS

Ainesh Panda (apanda4@uic.edu), Pragya Arora (parora27@uic.edu), Nithin Gowda (ngowda3@uic.edu), Chun Hung Lin (clin232@uic.edu), Ayushi Gaur (agaur8@uic.edu)

Faculty mentor: Prof. Fatemeh Sarayloo (fsaraylo@uic.edu)

University of Illinois at Chicago, Liautaud Graduate School of Business

informs ANALYTICS CONFERENCE

UIC BUSINESS

Introduction

Founded in 1980, CCC is a technology leader pioneering solutions that power insurers, automotive manufacturers, collision repairers, parts suppliers, lenders, fleet operators and more

CCC INTELLIGENT SOLUTIONS

Clientele:

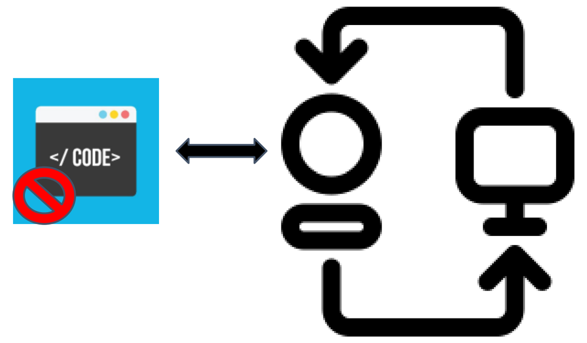
300+ insurers

27000 collision repair facilities processing 16 million repairs each year

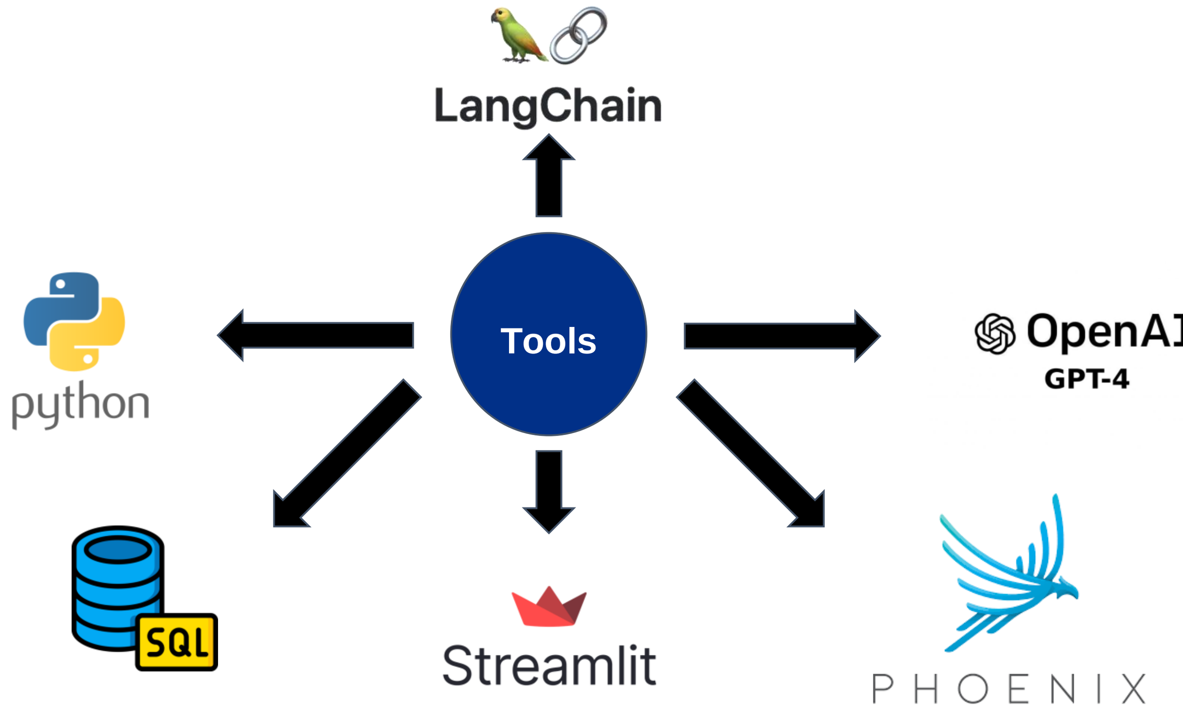
More than $1 trillion in historical claims data

Business Problem Framing

CCC wants to develop a framework to answer high-level business questions for its senior stakeholders who do not have in-depth knowledge of SQL and other technical tools.



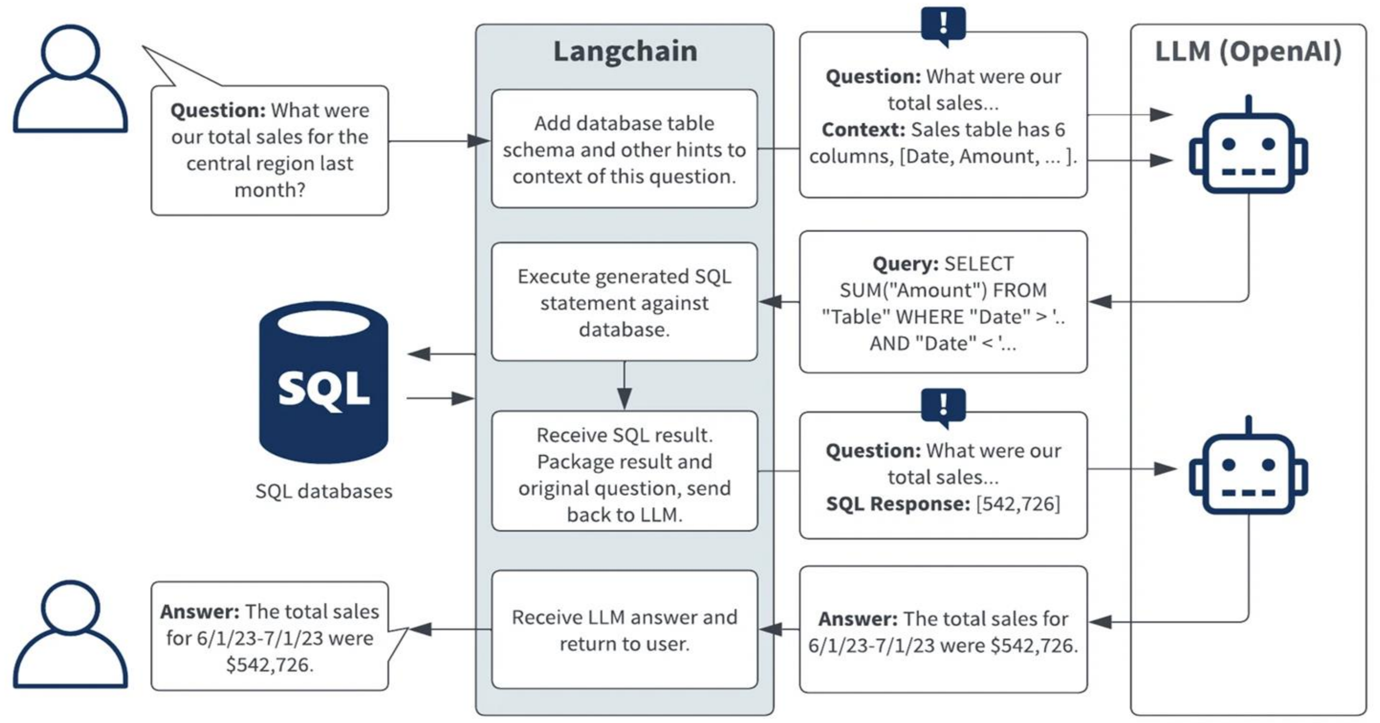
Analytical Problem Framing



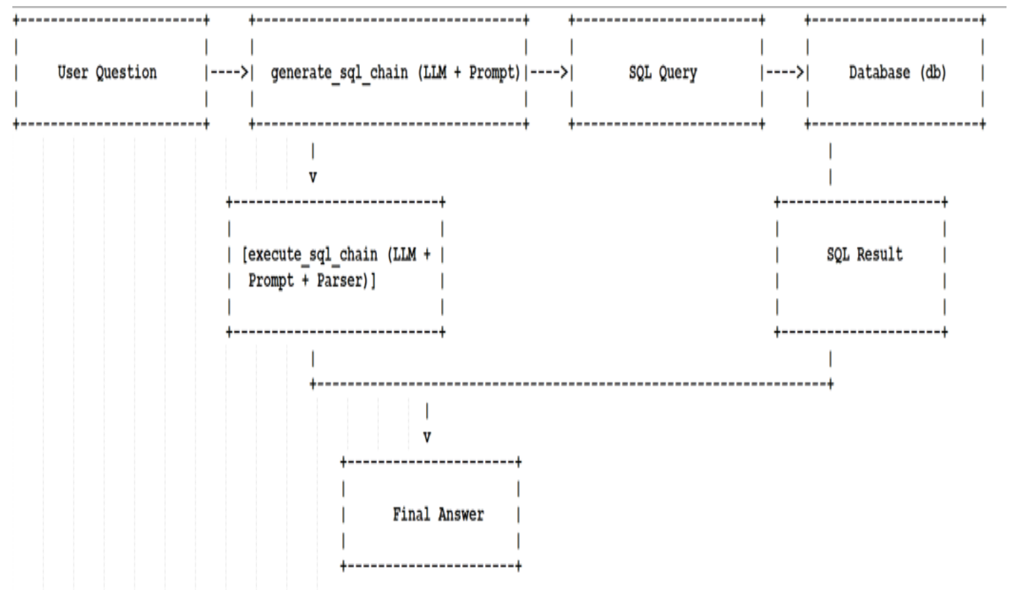
With the use of various technical tools, we can create a generative AI framework that can

1. Take the user input
2. Convert it into SQL code
3. Retrieve necessary data from the database
4. Convert it back into human language and feed it back to the user
5. Create a user interface

Methodology



The flow starts with the User asking a question, then leveraging LangChain a SQL chain is generated, which upon reaching the OpenAI framework would trigger the SQL sending the query in the database. SQL Chain is generated against the database with a parser and collectively we get both the SQL query as well as the Human form answer as a result.



Data

User Generated

15th January 2021 ⟷ ⟷ **19th February 2024**

Time Series Data

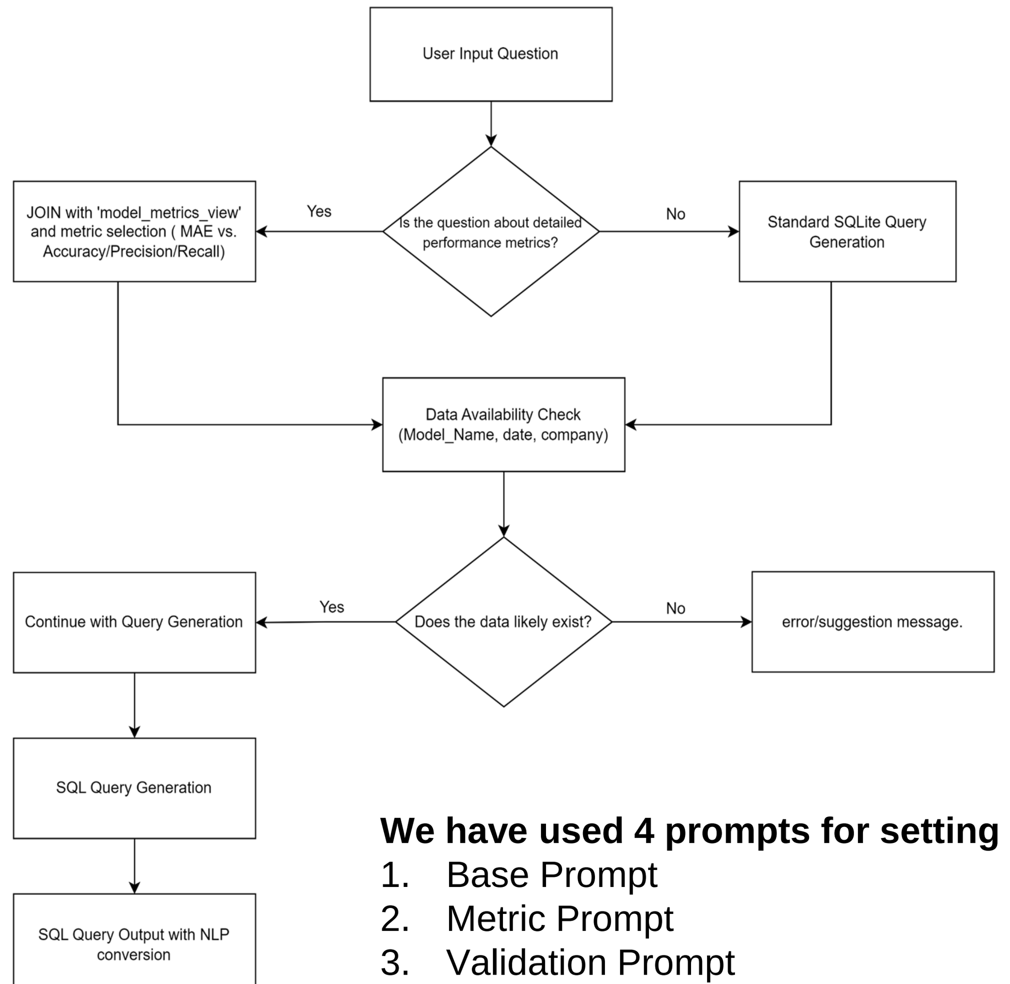
Consists of 50 Models 4797 rows of data

Regression: upto 3 versions, with daily input volume, and MAE as performance metric

Classification: upto 3 versions, with daily input volume, and Accuracy, Precision, Recall as performance metrics

Multi-Class: upto 3 versions, with daily input volume, and Accuracy, Precision, Recall as performance metrics

Model Building & Results



1. Meta-Programming for SQL Generation

2. Hallucination Mitigation Through Data Awareness

3. Few-shot Learning with Examples

Fine-tuning

01 Streamlined Input Prompts

Achieved a **42.2%** reduction in token count through conditional prompting

02 Expense Reduction

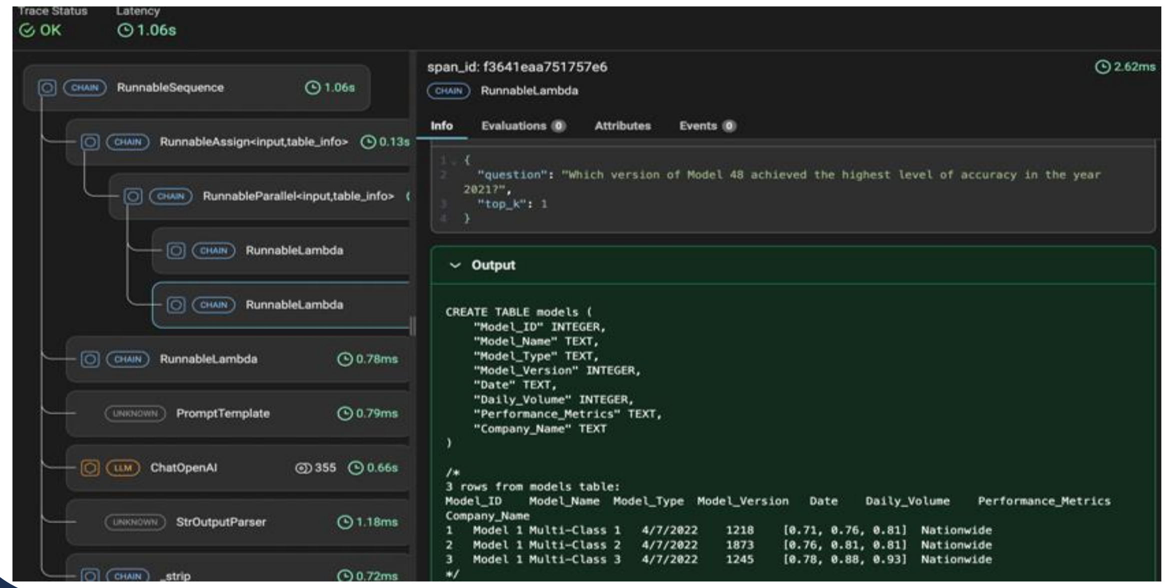
Resulting in **$35,900** in cost savings for 1 million queries

03 Few-shot Learning Integration

Improved query generation accuracy using Few-shot Learning

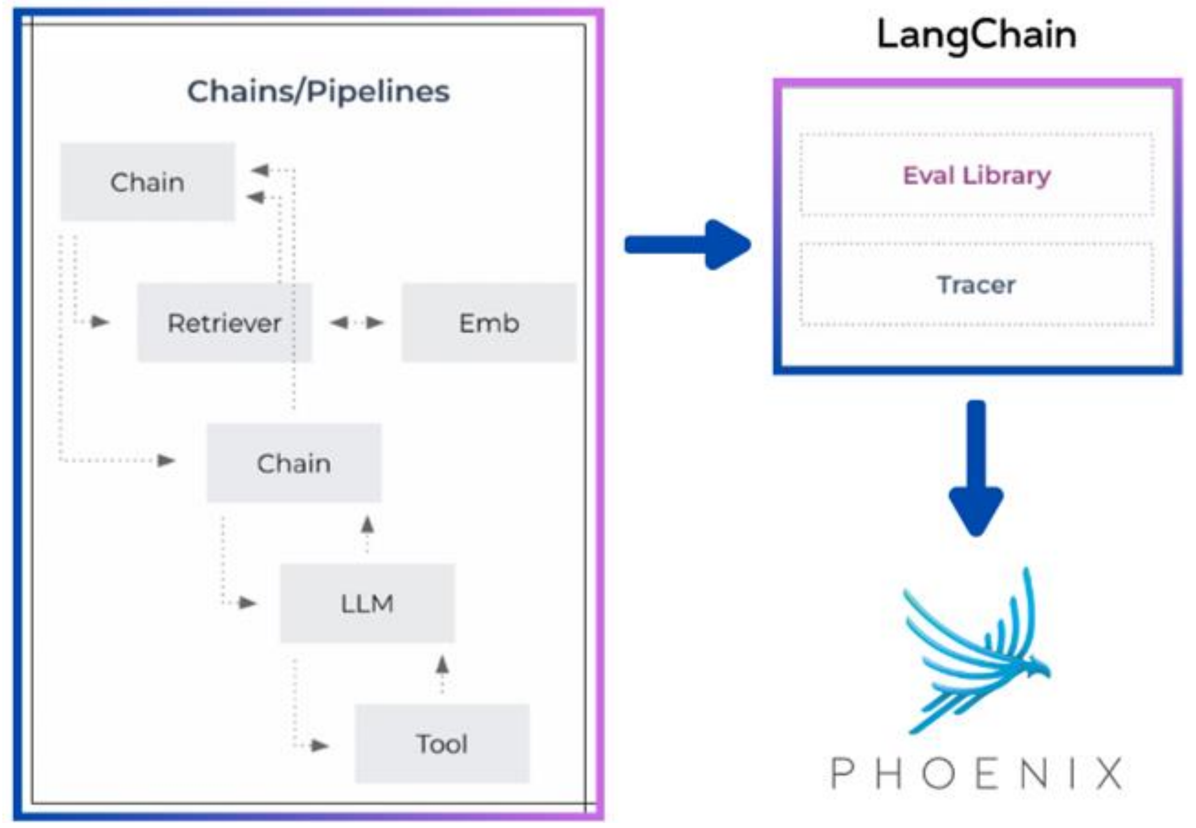
We have used 4 prompts for setting up prompt engineering:

1. Base Prompt
2. Metric Prompt
3. Validation Prompt
4. Answer Prompt

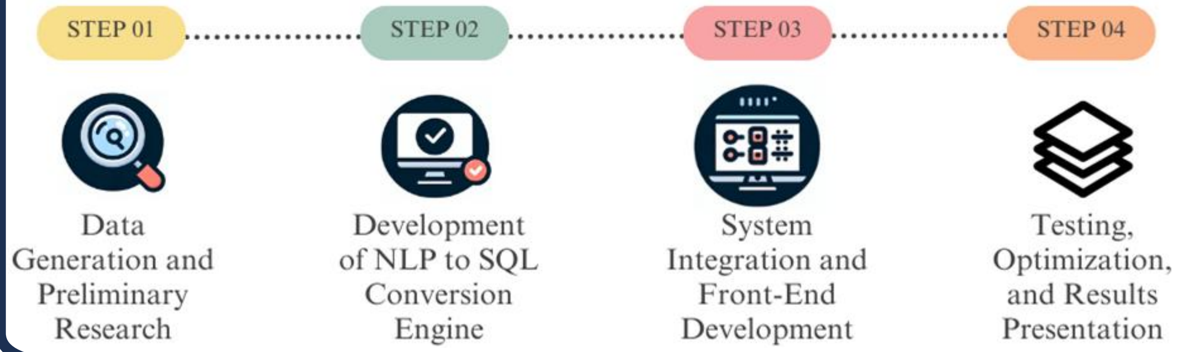


Testing: Phoenix

Tracing is a key feature of Phoenix that allows users to track and analyze the detailed journey of every single prediction. It can help us observe the root cause of the model's performance.



Deployment & Lifecycle Management



Deployment Server

- The Frontend was developed and customized using Streamlit libraries
- Using the Streamlit server, the app was successfully deployed as a web service.
- https://aicoc-guardian.streamlit.app/ - App link

Conclusion

Query Conversion

Developed an NLQ to SQL innovative end-to-end pipeline, reach high quality by using prompt engineering skills

Robust Integration & Testing

Integrated NLP techniques with OLAP engines, backed by rigorous development, testing, and refinement to ensure optimal performance and UI

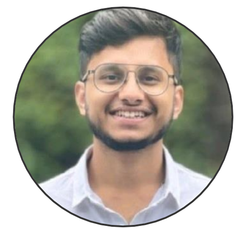
Acknowledgement

We wanted to thank Prof Sarayloo, and our technical supervisor Maryam Moghimi for giving us this opportunity, their guidance and support on this project

Ainesh Panda

Pragya Arora

Chun-Hung Lin



Nithin Gowda

Ayushi Gaur