

ML System Design

Tanishq Dubey

June 17, 2024

1 Introduction

This document presents the system design for an interactive AI system which integrates with Google Sheets to provide advanced data analysis and visualization capabilities. The system leverages a combination of a language model, text-to-SQL translation, and visualization tools to create an intuitive user experience. The implementation is provided [here](#).

2 Overview

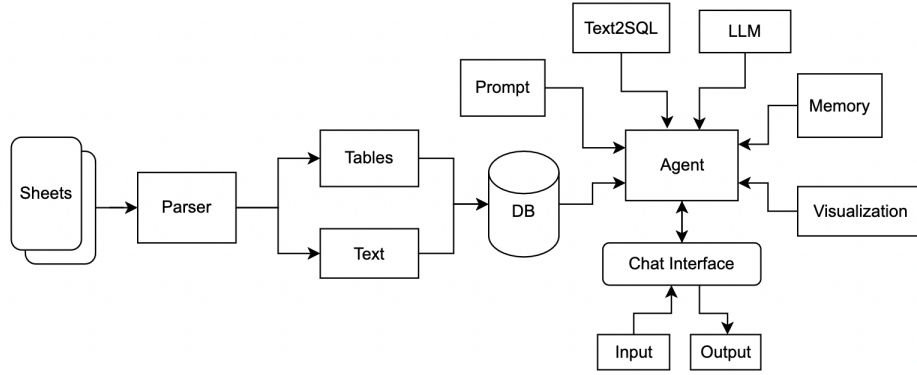


Figure 1: Overall Pipeline

3 Parser

The Parser component is responsible for interpreting and converting user inputs into structured queries. It extracts both text and all tables in CSV format. Given that the sheets file is unstructured, designing a parser can be challenging. We can create hand-tuned python scripts to detect tables in the unstructured excel sheet as the parser module. This will require a detailed look at various instances of the dataset to include all corner cases. In this implementation, however, GPT4o has been prompted to extract all the separate tabular data.

4 Database

4.1 SQL Database

The Database serves as the central repository for all operations over tabular data. This is a MySQL or PostgreSQL server constructed from the extracted tables using the parser.

4.2 Vector Database

There will also be a vector database which will additionally include the textual content from the sheets for RAG during conversational question answering. In the implementation provided, this step has been skipped but can be added easily.

5 Agent

The Agent is the core intelligence of the system, comprising several sub-components:

5.1 LLM

The Large Language Model (LLM) interprets user queries and generates human-like responses. We can use open-source as well as OpenAI models. In this specific implementation, GPT-4o has been used.

5.2 Text2SQL

The Text2SQL component translates natural language queries into SQL commands. It bridges the gap between the user's intent and the database, enabling precise data retrieval. This tool is provided to the LLM agent.

5.3 Memory

The Memory module retains context from previous interactions to provide a more seamless and coherent chat experience. It allows the system to remember past queries, preferences, and interactions, enabling follow-up questions and continuity in conversations. In the implementation provided, this step has been skipped but can be added easily.

5.4 Visualisation

The Visualisation component converts query results into graphical representations. It supports various types of charts and graphs, making it easier for users to interpret and analyze data. In the implementation provided, this step has been skipped but can be added easily.

5.5 Chat Interface

The Chat Interface is the user-facing component that facilitates interaction with the system. It provides a platform for users to input their queries and receive responses.