**Advanced Text to SQL**

Architecture:

A diagram of a software flowchart

Description automatically generated

**Problem Statement:**Our company have mysql db, it contains all ticket related data.Sample

Table names: closed\_tickets, open\_tickets, cr\_tickets, non\_cr\_tickets etc

Total Table count: 100 Tables.

Maximum Column Count: 40 columns.

Stakeholders need report about open tickets, closed tickets, how many tickets created, how many tickets closed, what is P1 ticket priority, which tickets taking more time etc  
Stakeholder instead of asking with developer for reports, directly chat with database and retrieve required data along visualisation reports.

How many few shot examples? ---250

What other embedding techniques tried apart from OpenAI Embeddings?

OpenAI Embeddings model: text-embedding-ada-002

Ollama embeddings model: nomic-embed-text  
  
Sentence Transformer embeddings : all-MiniLM-L6-v2

**Challenges of text to SQL project:**

**Challenge1:** Initially this project [hallucinations](https://www.k2view.com/what-are-ai-hallucinations/) is more because llm not aware of complete in depth db schema that’s why generating wrong queries.

Large and complex database schemas can be challenging for LLMs to comprehend and navigate effectively.

Solution: Hence we come with solution like embedding db tables schema details and storing in vector db and retrieving relevant table schema details for specific query.

**Challnege2:** understanding the true intent of a user's query can be difficult, especially when dealing with complex queries or domain-specific terminology.

Sometimes user can ask very complex questions and generating query for multiples table joins is most difficult.

**Challenge3:** DB values and llm generated values in sql query mismatching

select \* from phones where model=" Samsung"

But DB contains "Samsung India"

POC:

We are working and tried with Google's Text-to-SQL BERT tscholak/text-to-sql-t5-small

specifically, a fine-tuned version of Google's "T5-small" model, designed to translate natural language queries into SQL statements.

**Sample SQL Question and answers:**

**User Question:** What is the count of tickets for each priority level in the last month?

**SQL\_query:**

SELECT priority, COUNT(\*) AS ticket\_count FROM neutrino.tickets WHERE created\_date >= CURRENT\_DATE - INTERVAL '1 month' GROUP BY priority ORDER BY ticket\_count DESC;

**How have you handled ambiguity, context, and complex language constructs in the input queries?**

Ambiguity arises when the natural language input is unclear, has multiple interpretations, or lacks sufficient context. Here are strategies to address it:  
  
We are using query re-writing technique using llm for reducing Ambiguity problems.

**How do you handle evolving database schemas and keep your knowledge base up-to-date?**

**Recompute Embeddings**: Generate embeddings for the new table and column names and add into vector DB.

**Cross Interview Question**

1. Why did you choose Pinecone as the Vector Database in this pipeline? What are its advantages for this use case?
2. We have tried with Weaviate, Milvus But found more accuracy and good results with pinecone for semantic search results.
3. Pinecone is more effective for semantic search results.
4. Hybrid search capabilities.
5. Easy Integration and Deployment
6. Low Latency
7. Easy Scalability: handle high volumes of search requests without compromising performance
8. How do you handle schema changes in the database? If a table schema is updated, how does your system adapt to ensure accurate SQL generation?

We have setup one airflow scheduler job for every day , its fetching latest db table schema and update vector DB. Like this we will adopt new schema changes in the database.

1. How are embeddings generated for table schemas and few-shot examples? What ensures their relevance to the user's question?

A screenshot of a computer

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

1. What are embeddings?
2. An **embedding** is a numerical representation of text (or other data types) in a high-dimensional vector space.
3. These embeddings capture semantic relationships between words, phrases, or sentences, such that similar texts are mapped closer together in this space.
4. Cat and Dog embeddings closest each other and Cat and Car embeddings are far from each other.
5. How are embeddings generated?

OpenAI embeddings models like **text-embedding-ada-002** generate embeddings as follows:

1. **Input Sentence**: A sentence, like "OpenAI creates powerful AI models".
2. **Tokenization**: The sentence is broken into tokens (smaller chunks)
3. **Neural Network**: The model processes these tokens through a transformer-based architecture. Each token is passed through layers of self-attention and other transformations to learn contextual relationships.
4. **Output Vector**: The final output is a high-dimensional vector (e.g., a 1536-dimensional vector for text-embedding-ada-002), representing the sentence.

[0.123, -0.456, 0.789, ..., -0.231] # A vector of 1536 dimensions

These embeddings capture the meaning of the sentence in a way that is useful for tasks like search, classification, clustering, and recommendations.

1. How pinecone searches vectors from vector DB?
2. Approximate Nearest Neighbours (ANN) is a technique used to find the nearest vectors (most similar ones) to a given query vector in a high-dimensional space
3. Inverted File Index (IVF) is another method for ANN that divides the entire vector space into **clusters** and searches within the most relevant clusters only.
4. Why did you choose gpt-4o for this project? Did you experiment with other LLMs, and how did their performance compare in terms of SQL generation and query refinement?

I have tried with other llm models like llama3.1, llama3.2, mistral, gemma2 etc

Started facing issue with sql query generation and reasoning, output generation is not good.

That’s why we started using gpt-4o is really outstanding in reasoning and sql generation.

time saved in development and debugging justified the investment. GPT-4O's out-of-the-box performance was unmatched.

Googles text-to-sql-t5-small : This new thing I am trying now.

1. How does your system ensure that the generated SQL query is both syntactically and semantically correct before execution?
2. Use EXPLAIN keyword during sql execution for checking sql syntax is correct or not
3. Schema Awareness in LLM: Retrieve all tables and columns from DB and validate against query generated by LLM and checking Query semantically validation.
4. Did you fine-tune any LLM for this use case, or are you using a pre-trained model with zero/few-shot prompting? Why?

As of now, we haven’t started any fine tunning strategy.   
Currently we are use few shots prompting method.

1. If the Vector Database fetches irrelevant embeddings due to noise or poor similarity scoring, how does your system correct or filter this?
2. Normalize the embeddings by standard deviation for each dimension) before applying PCA
3. Before storing embeddings in the vector database, apply PCA to reduce dimensionality and minimize noise.
4. What mechanisms are in place to prevent SQL injection or malicious user queries being executed on the MySQL database?
5. Text to sql llm project db user have read only access.
6. Include constraints in the LLM prompt to ensure it generates only safe and specific SQL queries. Do not generate SQL queries that include DROP, ALTER, or DELETE commands.
7. If the query execution takes too long or times out, how does the system handle it? What are the fallback measures?

We didn’t face this issue, but we have already set db query execution timeout for more seconds.