Automating Text-to-SQL Conversion with Contextual Understanding

PAPER SUBMITTED BY

BHARATH GENJI MOHANA RANGA
MOWZLI SRE MOHAN DASS

Project Overview

In the modern data-driven world, accessing and querying databases is crucial for making informed decisions. However, a significant barrier exists for non-technical users who lack proficiency in SQL, the language used to interact with these databases. This project addresses this challenge by developing a system that translates natural language phrases into SQL queries.

Objectives and Significance

Objectives

- Develop Robust Translation
- Address Challenges
- Ensure Scalability

Significance

- Democratize Data Access
- Enhance Productivity
- Advance AI Research
- Enable Real-World Applications

Text-SQL Challenges

- Ambiguity in Natural Language
- Diverse Schema Structures
- Handling Context-Dependent Queries
- Synonym and Variation Mapping
- Limited Training Data
- Complex SQL Generation

Applications in Real-World Scenarios

Business Intelligence: Enabling business analysts to generate complex reports without deep SQL knowledge, thereby speeding up decision-making processes.

Education: Assisting students and researchers in accessing and manipulating database information without requiring extensive programming knowledge.

Healthcare Data Management: Streamlining data queries in medical databases, allowing healthcare professionals to retrieve patient data more efficiently without technical assistance.

Dataset Pathway

- KaggleDBQA
- Spider
- SEDE
- SQL-Eval

Human Evaluated
Needs more processing power
Non-diverse Schema Definition

Synthetic Data Generated using Ollama and Llama3:8b

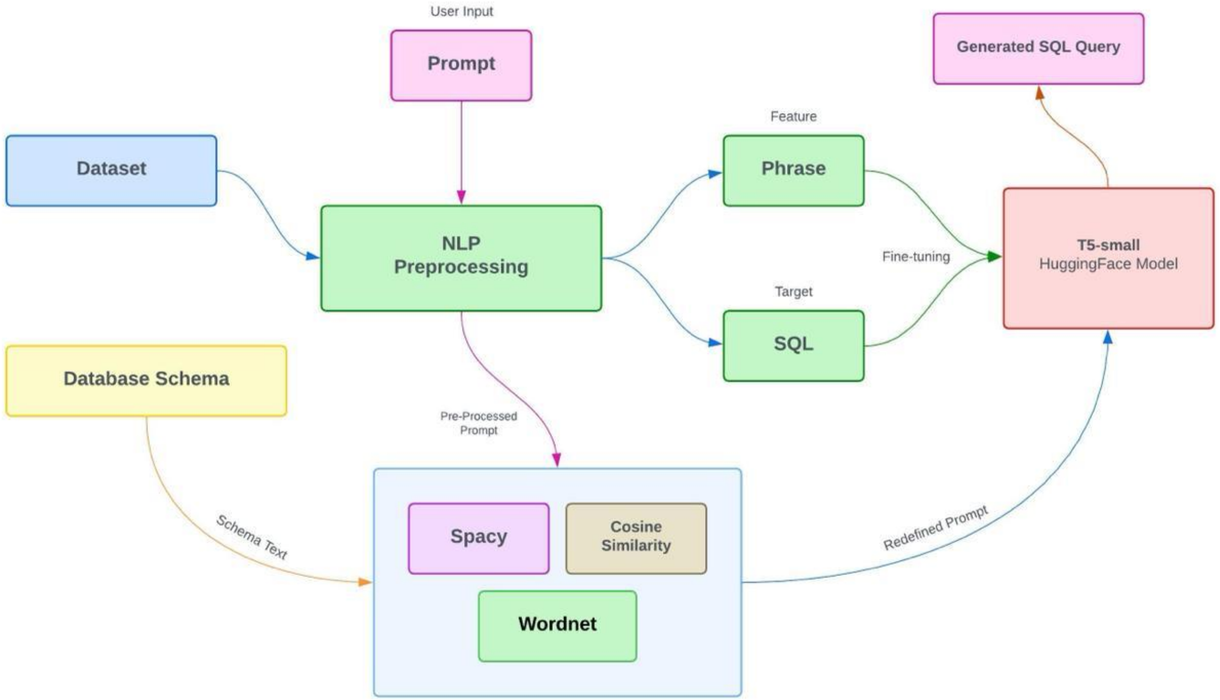
Generated using reliable LLM
Less Error %
Diverse Schema Definitions

Dataset Structure

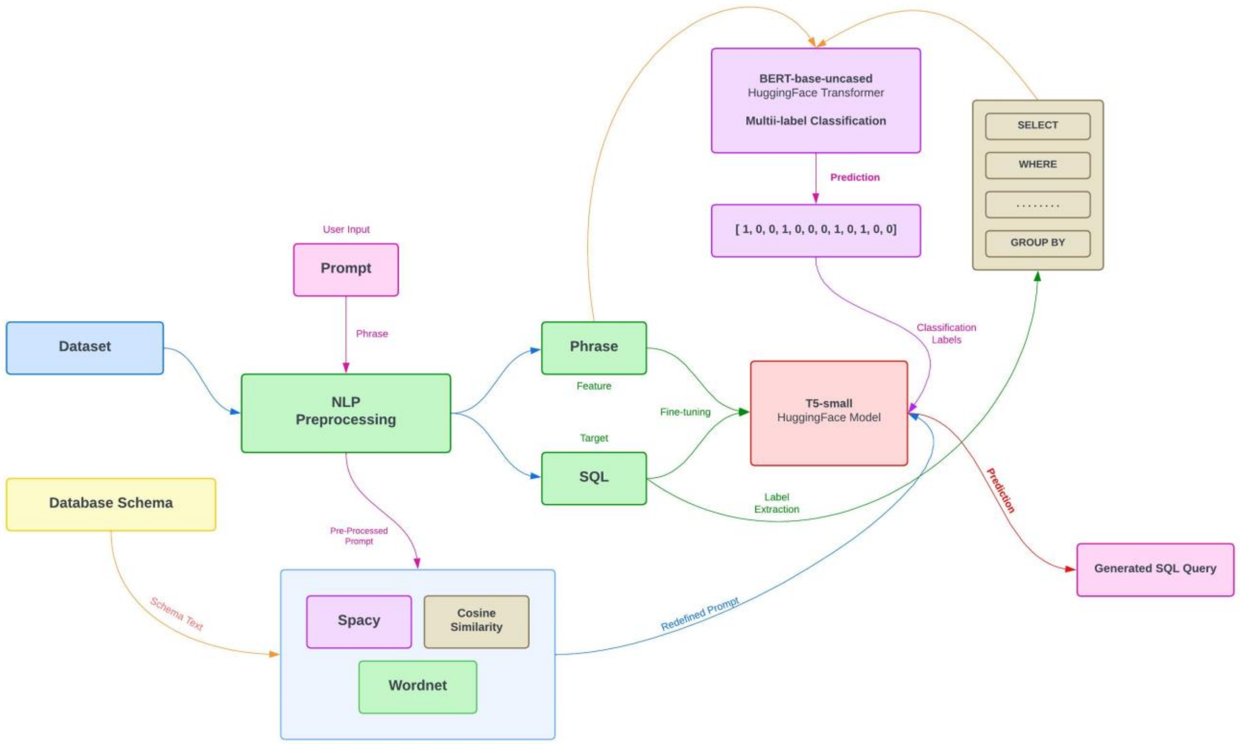
Phrase	SQL
Show all customer names	SELECT name FROM customers;
List all product categories with their descriptions	SELECT category, description FROM products;
Display order dates for all orders within the last year	SELECT order_date FROM orders WHERE order_date >= DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR);
Get employee names who work in the Marketing department	SELECT name FROM employees WHERE department = 'Marketing';
Retrieve the names of products with prices over $25	SELECT product_name FROM products WHERE price > 25;

60,000 rows of data is generated through Llama3:8b model using Ollama pipeline

Initial Workflow



Final Workflow



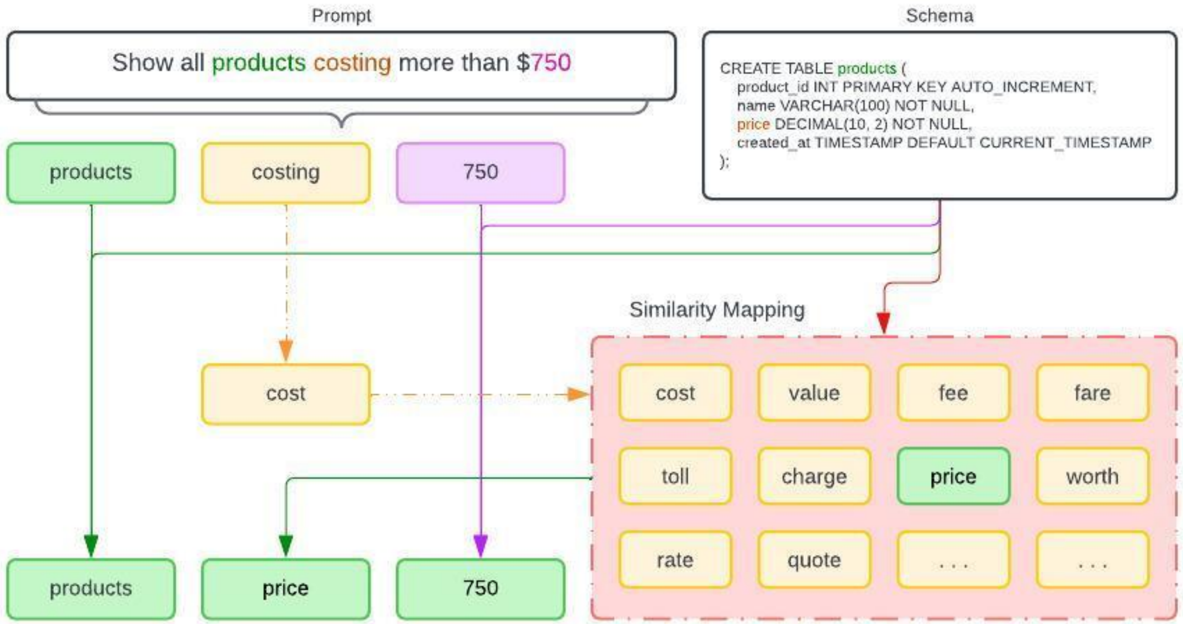
NLP Techniques

- Used Word Tokenizer to tokenize the words in the phrase
- PortStemmer to reduce the words to their base form
- Removed the stop words
- No other complex technics are involved

Phrase Prediction

- To reduce ambiguity, a similarity mapping system is leveraged
- Spacy and Cosine similarity functions are used to find the best match token
- User prompt tokens are matched with the schema and the prompt is rephrased
- This reduces the chances of generating irrelevant SQL

Phrase Prediction



Phrase prediction function incorporates cosine similarity and spacy functions.

This helps to identify the most relatable token from the schema and map it to the prompt

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$

Where the cosine value represents the high similarity nearing to 1 and low or no similarity nearing to 0

Label Extraction

- To give more contextual understanding, the labels are used
- This provides the model to directly depend on the correct label usage
- Label extraction is done on the dataset before training
- **SELECT, WHERE, GROUP BY, HAVING, ORDER BY, ASC, DESC, LIMIT, OFFSET, LIKE, BETWEEN, IN, IS NULL, IS NOT NULL** keywords are filtered from the **SQL** column
- They are extended as labels and flagged 1 or 0

Label Extraction

Phrase	SQL	SELECT	WHERE	GROUP BY	HAVING	ORDER BY	ASC	DESC	LIMIT	OFFSET	LIKE	BETWEEN	IN	IS NULL	IS NOT NULL
Show all cust	SELECT name	1	0	0	0	0	0	0	0	0	0	0	0	0	0
List all produ	SELECT categ	1	0	0	0	0	0	1	0	0	0	0	0	0	0
Display order	SELECT order	1	1	0	0	0	0	0	0	0	0	0	1	0	0
Get employe	SELECT name	1	1	0	0	0	0	0	0	0	0	0	1	0	0
Retrieve the	SELECT produ	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Show all cust	SELECT phon	1	0	0	0	0	0	0	0	0	0	0	0	0	0
List all order	SELECT order	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Display the n	SELECT name	1	1	0	0	0	0	0	0	0	0	0	1	0	0
Get the emai	SELECT emai	1	1	0	0	0	0	0	0	0	0	0	0	0	0

The labels are extracted and extended to labels and flagged
This will later be used for Multilabel Classification

Classification Pipeline

- A simple BERT base-uncased-model is leveraged
- CLS Token Embedding is implemented to represent each input sequence
- Two 1D Convolutional Layer is added to capture local feature within token embedding
  - Conv1 is used to map the 768 Channels Token Embedding to 256 Channels
  - Conv2 reduced the dimension 128 Channels
  - Global Average Pooling is enforced to reduce the output vector to a 128 fixed size vector

Classification Pipeline - Contd.

- Attention mechanism is introduced to focus on important tokens along with SoftMax activation to normalize these weights
- Two fully connected Layer combines the whole model
  - First Layer reduces the dimension with ReLU and dropout for regularization
  - Second Layer outputs the final logits for the classification task
- A Dropout of 0.5 is gracefully set to prevent overfitting
- BCEWithLogitsLoss is used to compute the logits to array of corresponding binary flags

Classification Metrics

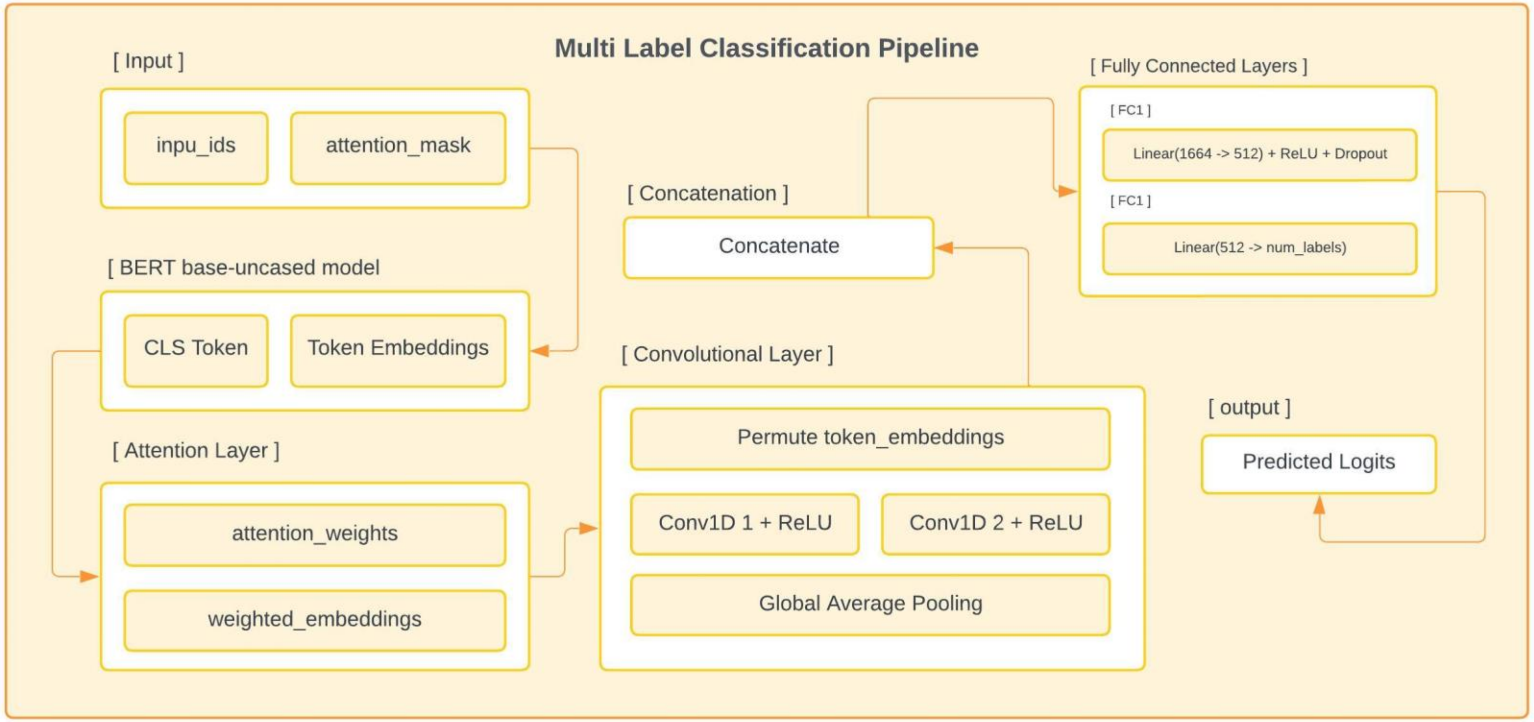
Train Loss	Test Loss	Precision	Recall	F1-Score	Subset Accuracy	Jaccard Score
0.0844	0.0706	0.9380	0.5708	0.7172	0.7512	0.8412

```
--------------------------------------------------
Epoch 15/20 | Train Loss: 0.0979, Time: 179.31s | Test Loss: 0.0809, Time: 48.86s | Precision: 0.9202,
Recall: 0.5136, F1-Score: 0.6735 | Subset Accuracy: 0.7455, Jaccard: 0.7870, Hamming Loss: 0.0248
--------------------------------------------------
Epoch 16/20 | Train Loss: 0.0944, Time: 179.30s | Test Loss: 0.0782, Time: 48.87s | Precision: 0.9278,
Recall: 0.5385, F1-Score: 0.6948 | Subset Accuracy: 0.7500, Jaccard: 0.8140, Hamming Loss: 0.0242
--------------------------------------------------
Epoch 17/20 | Train Loss: 0.0912, Time: 179.33s | Test Loss: 0.0761, Time: 48.89s | Precision: 0.9305,
Recall: 0.5550, F1-Score: 0.7043 | Subset Accuracy: 0.7502, Jaccard: 0.8252, Hamming Loss: 0.0240
--------------------------------------------------
Epoch 18/20 | Train Loss: 0.0890, Time: 179.32s | Test Loss: 0.0740, Time: 48.85s | Precision: 0.9340,
Recall: 0.5610, F1-Score: 0.7098 | Subset Accuracy: 0.7509, Jaccard: 0.8310, Hamming Loss: 0.0235
--------------------------------------------------
Epoch 19/20 | Train Loss: 0.0862, Time: 179.33s | Test Loss: 0.0723, Time: 48.87s | Precision: 0.9362,
Recall: 0.5655, F1-Score: 0.7135 | Subset Accuracy: 0.7510, Jaccard: 0.8364, Hamming Loss: 0.0232
--------------------------------------------------
Epoch 20/20 | Train Loss: 0.0844, Time: 179.33s | Test Loss: 0.0706, Time: 48.87s | Precision: 0.9380,
Recall: 0.5708, F1-Score: 0.7172 | Subset Accuracy: 0.7512, Jaccard: 0.8412, Hamming Loss: 0.0227
--------------------------------------------------
```

We still have room for improving the Recall, given more diverse data

Because of the nature of the Dataset (From Llama3:8b), the hamming loss is very low and ignored for metrics

Multi Label Classification Model



Show all customer names

[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Text-SQL Prediction

- T5ForConditionalGeneration model and T5Tokenizer is used
- The model takes a Phrase and Label as input and outputs a relevant SQL
- AdamW optimizer is used to train the model with a learning rate of 0.0001
- Cross-Entropy Loss is used to minimize the difference between predicted tokens and ground truth SQL tokens
- The model employs beam search decoding to generate SQL queries during prediction, balancing accuracy and fluency

Training and Strength

- Combines phrases, labels, and additional metadata to enhance SQL query generation.
- Custom Seq2SeqDataset handles preprocessing and tokenization for inputs and targets.
- Uses PyTorch's DataLoader for efficient batching and shuffling during training.
- By combining natural language phrases with metadata (labels), the model can leverage both linguistic understanding and structured information.

Prediction Metrics

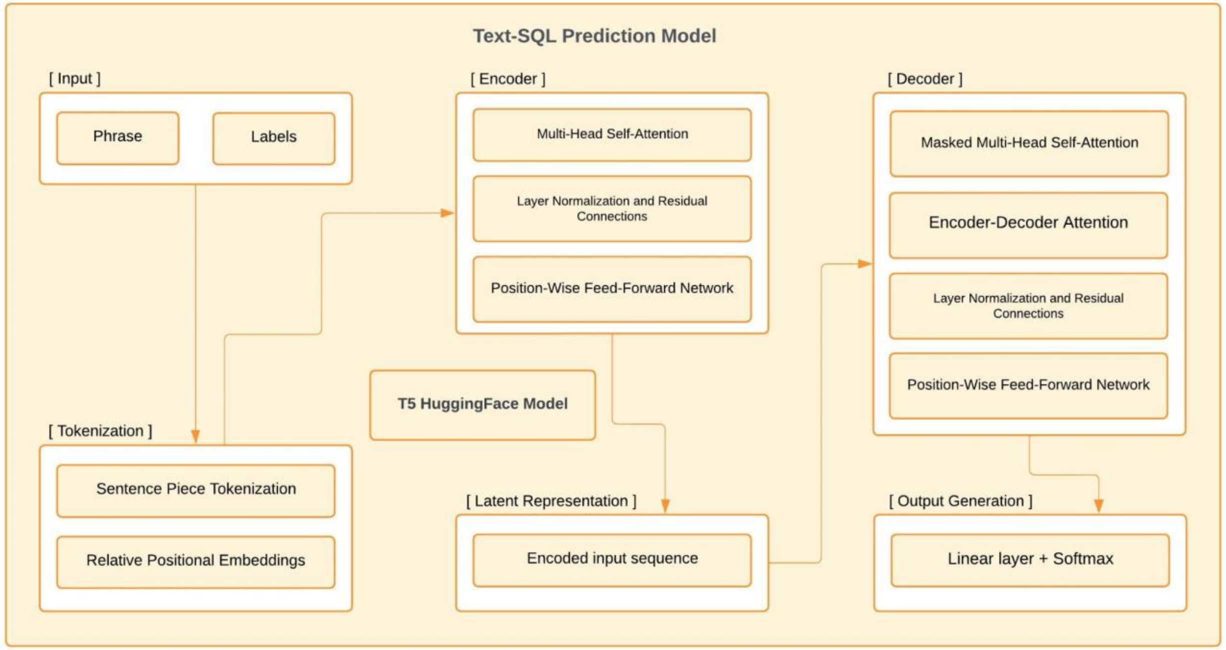
```
Epoch 1: Train Loss: 0.7079, Val Loss: 0.1236
Epoch 2: Train Loss: 0.0772, Val Loss: 0.0612
Epoch 3: Train Loss: 0.0483, Val Loss: 0.0484
Epoch 4: Train Loss: 0.0369, Val Loss: 0.0432
Epoch 5: Train Loss: 0.0298, Val Loss: 0.0405
Epoch 6: Train Loss: 0.0250, Val Loss: 0.0387
Epoch 7: Train Loss: 0.0211, Val Loss: 0.0378
Epoch 8: Train Loss: 0.0182, Val Loss: 0.0383
Epoch 9: Train Loss: 0.0167, Val Loss: 0.0378
Epoch 10: Train Loss: 0.0145, Val Loss: 0.0379
```

The Model performed well, during the training with balanced decrease in Train and Test losses, which shows the model is not overfitting

```
BLEU Score on Validation Set: 56.8654
```

The trained model is then evaluated by BLEU score

Text-SQL Prediction Model



Show all customer names

+

[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

SELECT name FROM customers

Challenges and Solutions

- Handling Ambiguous Phrases
- Schema Constraints and Query Validation
- Improving Training and Prediction Accuracy
- Handling two different models and executing at optimal memory

Conclusion and Summary of Findings

The project successfully developed a system that translates natural language into SQL queries, addressing the barrier faced by non-technical users.

Key challenges such as ambiguity in natural language, diverse schema structures, and context-dependent queries were systematically tackled.

The implementation of models like T5 for conditional generation and BERT for classification enhanced the accuracy and fluency of SQL generation.

Through synthetic and human-evaluated data, the system achieved high precision, though improvements in recall are still needed.

Future Work and Enhancements

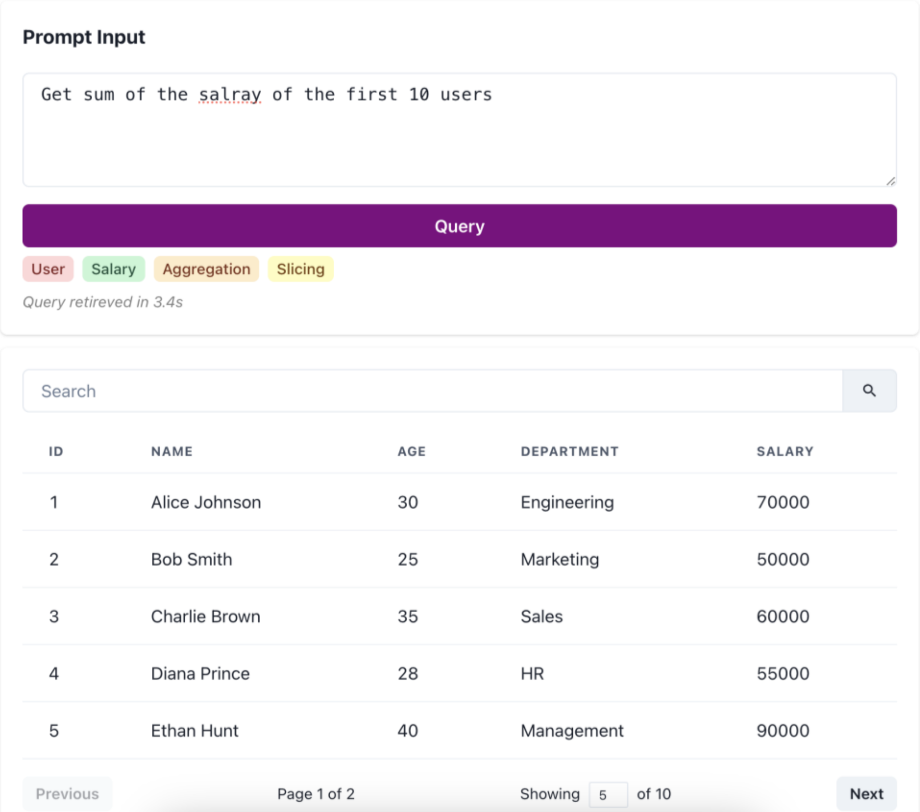
Data Diversity: Incorporating a more diverse set of training data to improve model recall and overall accuracy.

Model Optimization: Further refining the machine learning models to enhance efficiency and reduce errors in SQL generation.

Real-time Adaptability: Developing capabilities for the system to adapt to real-time changes in database schema and query requirements.

User Interface: Enhancing the user interface to make it more intuitive for non-technical users, possibly incorporating voice-to-SQL functionalities.

Future Concept



Thank You