

A Survey of NL2SQL with Large Language Models: Where are we, and where are we going?

Speaker: Jiahao He

School of Information
Renmin University of China

March 27, 2025

Motivating Example

NL query: Find the *names* of all *customers* who checked out *books* on exactly 3 different *genres* on *Labor Day in 2023*.

Database:

Customer		
CustomerId	Name	...

Book				
BookId	Title	LiteralGenres	SubjectGenres	...

BookOrder			
CustomerID	BookID	OrderDate	...

Additional Information: Note that *Labor Day* stands for May 1st.

Motivating Example

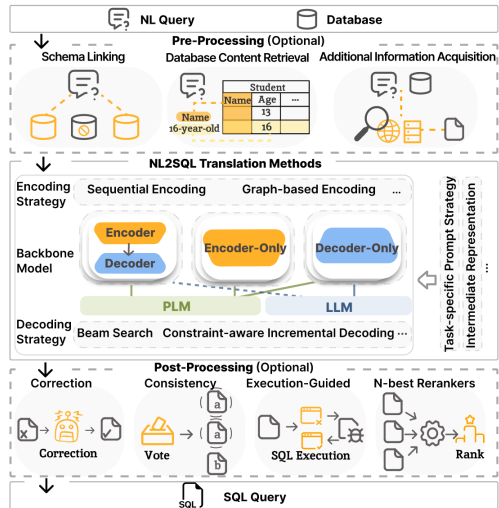
1. Find key components in NL: **entities**, **attributes**, **temporal information**, specific conditions...
2. Find relevant tables, columns and, cell values in DB: **Customer**, **Book**, **BookOrder**, **Name**, **LiteralGenres**, **SubjectGenres**...
3. Writing SQL base on NL and DB understanding:

```
SELECT FROM Customer  
NATURAL JOIN BookOrder  
NATURAL JOIN Book  
WHERE OrderDate='2023-05-01'  
GROUP BY CustomerId  
HAVING COUNT(DISTINCT  
LiteralGenre)=3
```

```
SELECT FROM Customer  
WHERE CustomerId = (SELECT  
CustomerId  
FROM BookOrder  
NATURAL JOIN Book  
WHERE OrderDate= '2023-05-01'  
GROUP BY CustomerId  
HAVING COUNT(DISTINCT  
SubjectGenre)=3)
```

Challenges and Solutions

- C1: **Uncertain natural language query:** lexical and syntactic ambiguity. . .
- C2: **Complex database and dirty content:** complex relations, ambiguity attributes. . .
- C3: **NL2SQL translation:** multiple possible queries. . .
- C4: **technical challenges in developing NL2SQL solutions:** efficiency, reliability. . .



Preprocessing Strategies for NL2SQL

A. Schema Linking

- String Matching-based: Identifies matches when candidates are identical or when one is a substring of the other. -**synonyms and vocabulary variations**.
- Neural Network-based: Use neural networks to learn the relationship between the NL and the DB schema. -**hard to generalize across databases**.
- In-Context Learning: Utilizes LLMs' ability to understand complex language patterns and relationships within data schemas. -**computationally expensive**.

Example

Given the database schema and question, perform the following actions:

1 - Rank all the table base on the possibiliy of being used in the SQL according to the question from the most relevant to the least relevant. Table or its columns that matches more with the question's words is highly relevant and must be placed ahead.

...

Preprocessing Strategies for NL2SQL

B. Database Content Retrieval: extracting cell values for specific SQL clauses

- String Matching-based: Similiar with schema linking.-synonyms and computationally expensive
- Neural Network-based: Capture complex data and semantic features through layers of nonlinear transformations, helping to resolve synonym issues -ambiguous NL.
- Index-based: Enabling faster access to relevant cell values. -frequent changes.

Example

NL: "Show me the names of all employees in the Sales department."

SQL: SELECT Employees.Name

FROM Employees

JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID

WHERE Departments.DepartmentName = 'Sales';

Preprocessing Strategies for NL2SQL

C. Additional Information Acquisition: fetch additional information for the translation.

- Sample-based: Use few-shot examples from extensive domain knowledge
- Retrieval-based: Retrieval similar information.

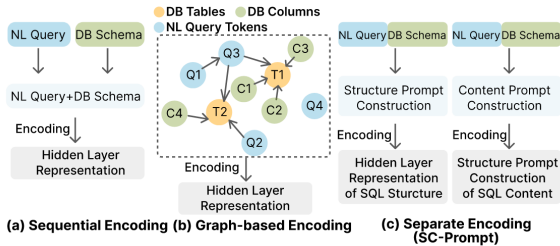
Example

NL: Find the *names* of all *customers* who checked out *books* on exactly 3 different *genres* on *Labor Day in 2023*.

Additional Information: Note that *Labor Day* stands for *May 1st* in China and *September 4th* in the US.

NL2SQL Translation Methods

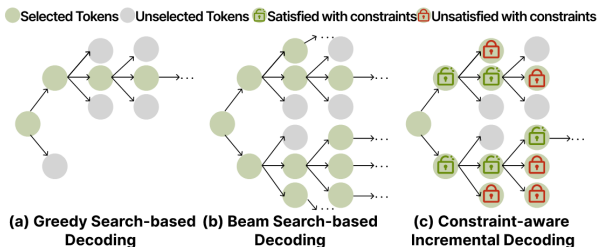
A. *Encoding Strategy*: transforming NL and database schema into a structured format suitable for language model processing.



- 1) Sequential Encoding Strategy: Both the NL and the database schema are treated as token sequences.
- 2) Graph-based Encoding Strategy: Represents both NL and database schema as interconnected graphs.
- 3) Separate Encoding Strategy: Different parts of the NL are encoded separately and then combined to form the final SQL.

NL2SQL Translation Methods

B. Decoding Strategy: Transforming encoder-generated representations into SQL.



- 1) Greedy Search-based Decoding: Select the token with the highest probability at each decoding step.
- 2) Beam Search-based Decoding Strategy: Retains multiple candidate sequences and expands the top-k tokens with the highest probabilities until a complete output sequence is generated.
- 3) Constraint-aware Incremental Decoding Strategy: As the model predicts the next token, it also checks the syntactic correctness of the partial query .

NL2SQL Translation Methods

C. Task-specific Prompt Strategy

- 1) Chain-of-Thought (CoT) Prompting: Use COT process to do entity and context retrieval, schema selection, SQL generation, and revision.
- 2) Decomposition Strategy: divides the NL2SQL task into sequential subtasks, allowing each sub-module to concentrate on a specific generation step.

NL2SQL Translation Methods

D. Intermediate Representation for NL2SQL Translation

Intermediate Representation:

```
SELECT film.title  
WHERE count(film_actor.*)>5
```

(a) SQL-like Syntax Language (e.g. NATSQL)

```
SELECT [column] ----- [column] title  
FROM [table] ----- [table] film  
JOIN [table] ----- [table] film_actor  
ON [table].[column] ----- [table].[column] film.film_id  
=[table].[column] ----- [table].[column] film_actor.film_id  
GROUP BY [column] ----- [column] film_id  
HAVING count([column]) > n [column] * [n] 5
```

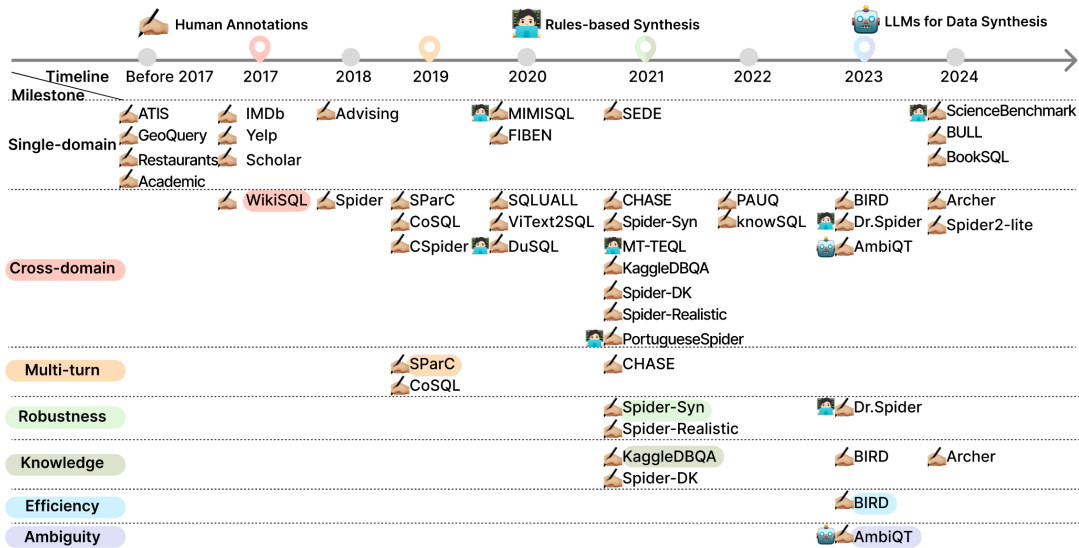
(b) SQL-like Sketch Structure (e.g. SC-Prompt)

- 1) SQL-like syntax language: Use simplified SQL-like structure.
- 2) SQL-like sketch structure: Use SQL-like sketches that mirror SQL structure for parsing, enabling diverse NL queries to be mapped into a specific sketch space.

Postprocessing Strategies for NL2SQL

- A. SQL Correction Strategies:* Mainly focus on correcting syntax errors in the generated SQL queries.
- B. Output Consistency:* Ensure that the generated SQL query is consistent with the NL query.
- C. Execution Guided Strategies:* Refine SQL based on execution results, ensuring the query retrieves data correctly.
- D. N-best Reranker Strategies:* Reorder the top-N model outputs, often leveraging a larger model (BERT-based reranker) or additional knowledge sources.

NL2SQL Benchmarks



Evaluation and Error Analysis

Metrics:

- *Execution Accuracy(EX)*: $EX = \sum_{i=1}^N \mathbb{1}(V_i = \hat{V}_i)/N$, $\mathbb{1}(\cdot) = 1$ when \cdot is true, and 0 otherwise.
- *String-Match Accuracy(SM)*: $SM = \sum_{i=1}^N \mathbb{1}(Y_i = \hat{Y}_i)/N$.
- *Component-Match Accuracy(CM)*: $CM^C = \sum_{i=1}^N \mathbb{1}(Y_i^C = \hat{Y}_i^C)/N$.
- *Exact-Match Accuracy(EM)*: $EM = \sum_{i=1}^N \mathbb{1}(\wedge_{C_k \in \mathbb{C}} Y_i^{C_k} = \hat{Y}_i^{C_k})/N$
- *Valid Efficiency Score(VES)*:

$$VES = \frac{1}{N} \sum_{i=1}^N \mathbb{1}(V_i = \hat{V}_i) \cdot \mathbf{R}(Y_i, \hat{Y}_i), \quad \mathbf{R}(Y_i, \hat{Y}_i) = \sqrt{\mathbf{E}(Y_i)/\mathbf{E}(\hat{Y}_i)}$$

- *Query Variance Testing(QVT)*:

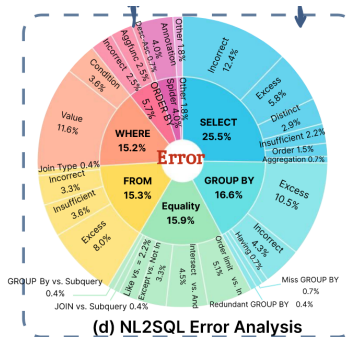
$$QVT = \frac{1}{N} \sum_{i=1}^N \left(\frac{\sum_{j=1}^{m_i} \mathbb{1}(\mathcal{F}(Q_{ij}) = Y_i)}{m_i} \right)$$

Evaluation and Error Analysis

Evaluation toolkits: MT-TEQL, NL2SQL360...

Error analysis taxonomy:

1. *Error Location*: identify specific SQL components where errors occur, such as the SELECT or WHERE clause.
2. *Cause of Error*: focus on the underlying reasons for the error.



Practical Guidance for NL2SQL Solutions

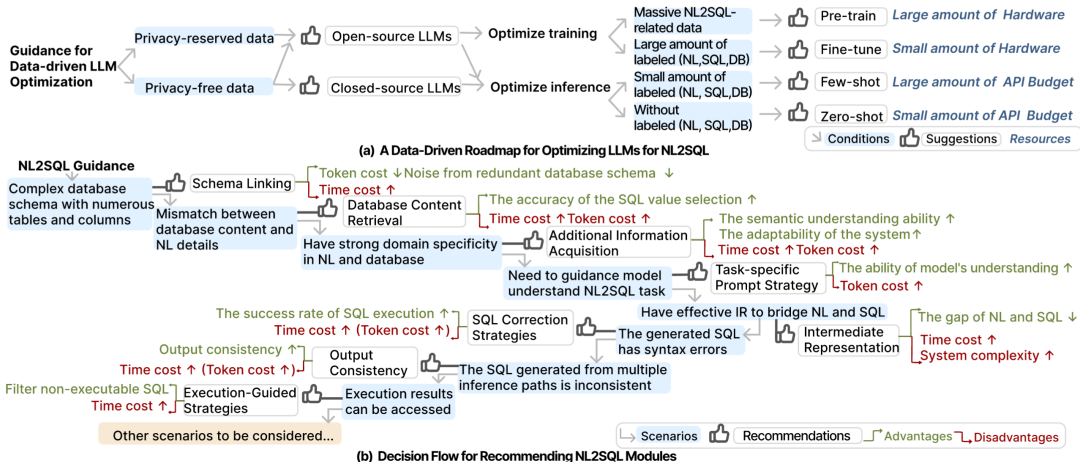


Fig. 11: A Data-Driven Roadmap and a Decision Flow for Recommending NL2SQL modules.

Open Problems

The open NL2SQL problems:

- Database retrieval: find relevant databases to access.
- Handling heterogeneous schemas: integrating data with varied structures and terminologies
- Answer aggregation: inferring final answers from multiple SQL queries
- Domain Adaption: generalizing models across domains
- Scalability and efficiency: managing large data volumes while maintaining performance

Developing cost-effective NL2SQL methods:

Make NL2SQL solutions trustworthy:

- Interpreting NL2SQL solutions: Understanding the reasoning behind a NL2SQL model's performance
- NL2SQL debugging tools: a visual debugger for NL2SQL
- Interactive NL2SQL Tools: focus on each part before assembling the whole SQL.

The End