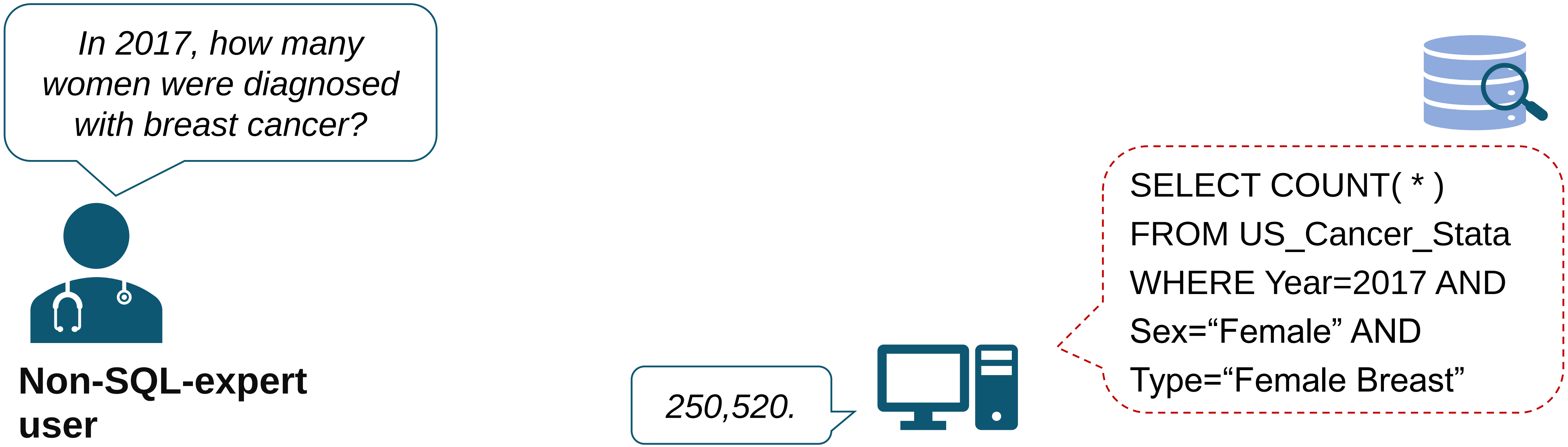
Enhancing Text-to-SQL Capabilities of Large Language Models through Tailored Promptings

Zhao Tan, Xiping Liu, Qing Shu, Xi Li, Changxuan Wan, Dexi Liu, Qizhi Wan, Guoqiong Liao

LREC-COLING 2024

江西財經大學
JIANGXI UNIVERSITY OF FINANCE AND ECONOMICS

Text-to-SQL Semantic Parsing



Two Paradigms for Developing Text-to-SQL Parsers

Supervised Learning

Collect large-scale data
↓
Train an model for hours
↓
Evaluate with unseen data

In-Context Learning

Task-tailored Promptings
↓
LLMs
↓
Evaluate with unseen data

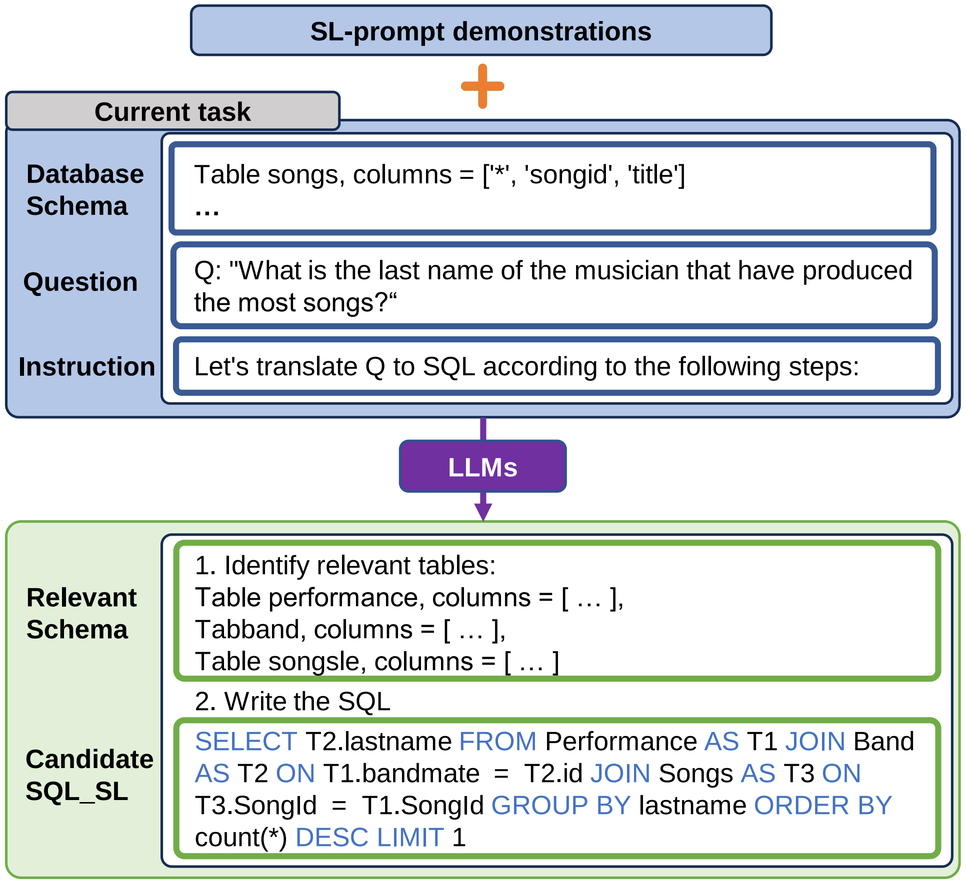
Two Pain Points for Text-to-SQL Semantic Parsing

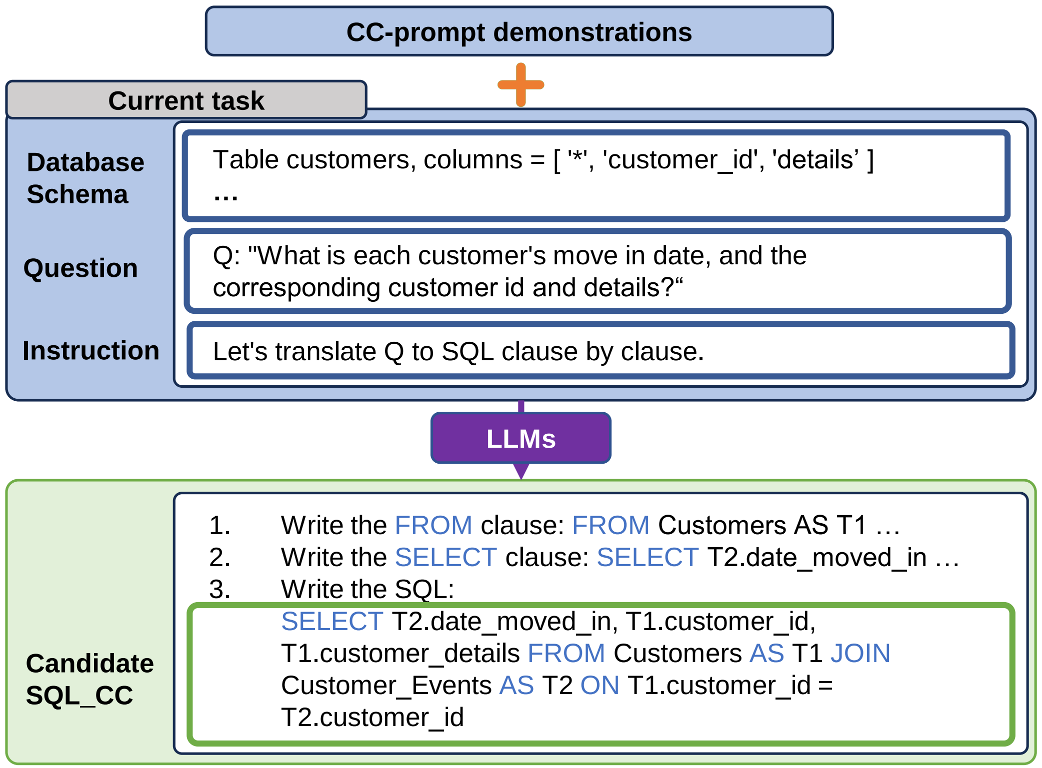
- Schema Linking
- SQL Generation

Enhancing Text-to-SQL Capabilities of Large Language Models through Tailored Promptings

Our Prompting Methods

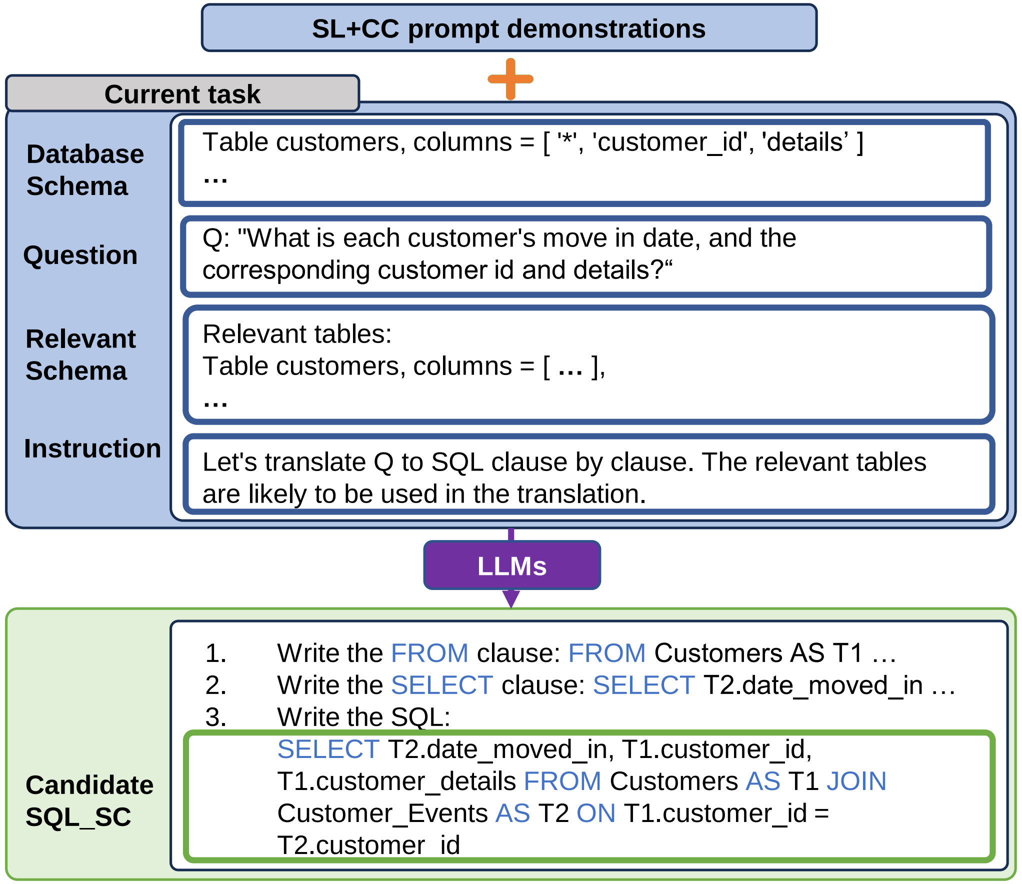
- Schema Linking → **SL-prompt**
- SQL Generation→ **CC-prompt**



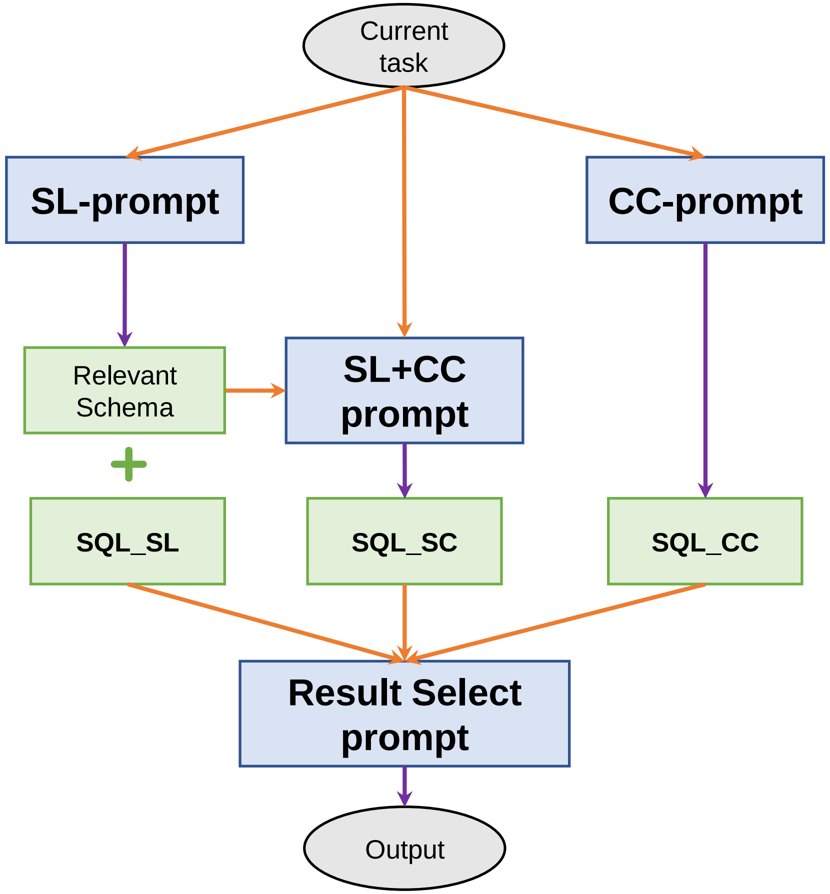


Our Prompting Methods

- Combining prompt → **SL+CC prompt**



- Overview



Our Prompting Methods

- Candidate Select → **Result Select prompt**

Instruction
Given the following information: a database schema, a natural language question, and some SQL queries, your task is to perform the following steps:

Database Schema
Database Schema:
Table model_list, columns = ['*', 'modelid', 'maker', 'model'] ,
...

Question
Q: " Which model of the car has the minimum horsepower?"

Candidate SQL1 (SQL_SL)
```
SELECT model_list.model FROM cars_data JOIN car_names ON cars_data.id =
car_names.makeid JOIN model_list ON car_names.model = model_list.model JOIN (SELECT
MIN(horsepower) AS min_hp FROM cars_data) AS min_hp_table ON cars_data.horsepower =
min_hp_table.min_hp
```

Candidate SQL2 (SQL_CC)
```
SELECT model FROM model_list WHERE modelid IN (SELECT id FROM cars_data WHERE
horsepower = (SELECT MIN(horsepower) FROM cars_data))
```

Candidate SQL3 (SQL_SC)
```
SELECT T3.model FROM cars_data AS T1 JOIN car_names AS T2 ON T1.id = T2.makeid
JOIN model_list AS T3 ON T2.model = T3.model ORDER BY T1.horsepower ASC LIMIT 1
```

Tasks
1. Explain both each SQL separately.
2. Determine which one SQL correctly answers the given question.

→ **LLMs** →

1. Explanation of SQL queries:
SQL1: This SQL query attempts to find the car model with the minimum horsepower by joining the three tables cars_data, cars_names, and model_list, along with a subquery. However, if there are multiple cars with a minimum horsepower, then this query will return models for all of those cars, not just one.
SQL2: This SQL query attempts to select the model by selecting it from the model_list table, where the modelid is in a list generated by a subquery. However, the problem with this query is that it incorrectly assumes that the id field in the cars_data table.
SQL3: This SQL query is designed to find the car model with the lowest horsepower. It does this by joining the three tables cars_data, car_names, and model_list. The query then …

2. The SQL query that correctly answers the given question is SQL3.

Exprimental Setup

- LLMs
  - GPT-3.5-Turbo
  - GPT-4
- Datasets:
  - Spider (Yu et al., 2018)
  - Spider-DK (Gan et al., 2021a)
  - Spider-SYN (Gan et al., 2021b)
  - Spider-Realistic (Deng et al., 2021)
- Evaluation Metric
  - Execution Accuary (Yu et al., 2018)
  - Test-Suite Accuary (Zhong et al., 2020)

Exprimental Results

Methods	EX	TS
Few-shot + CodeX (Rajkumar et al., 2022)	67.0	55.1
Zero-shot + ChatGPT (Liu et al., 2023)	70.1	60.1
Coder-Reviewer + CodeX(Zhang et al., 2022)	74.5	-
MBR-Exec (Shi et al., 2021)	75.2	-
T5-3B + PICARD (Scholak et al., 2021)	79.3	69.4
RASAT + PICARD (Li et al., 2023b)	80.5	70.3
LEVER + CodeX (Ni et al., 2023)	81.9	-
RESDSQL-3B + NatSQL (Li et al., 2023a)	84.1	73.5
Self-Debug + CodeX (Chen et al., 2023)	84.1	-
SPDS + CodeX (Nan et al., 2023)	84.4	-
DIN-SQL + GPT-4 (Pourreza and Rafiei, 2023)	**85.1**	**74.2**
Ours (GPT-3.5)	78.6	68.3
Ours (GPT-4)	**86.2**	**76.9**

Exprimental Results

Methods	Spider-DK		Spider-Syn		Spider-Realistic	
	EX	TS	EX	TS	EX	TS
T5-3B + PICARD (Scholak et al., 2021)	62.5	-	69.8	61.8	71.4	61.7
RASAT + PICARD (Wang et al., 2020)	63.9	-	70.7	62.4	71.9	62.6
RESDSQL-3B + NatSQL (Li et al., 2023a)	**66.0**	-	**76.9**	**66.8**	**81.9**	**70.1**
Zeroshot + ChatGPT (Liu et al., 2023)	62.6	-	58.6	48.5	63.4	49.2
Ours (GPT-3.5)	63.9	-	67.1	57.6	70.7	58.3
Ours (GPT-4)	**67.2**	-	**78.1**	**68.6**	**82.8**	**70.6**

Exprimental Results

Execution accuracy					
Methods	**Easy**	**Medium**	**Hard**	**Extra-hard**	**EX**
Few-shot + GPT-3.5	91.1	78.5	58.0	46.4	72.9
Ours (GPT-3.5)	**91.5**	**85.4**	**67.0(11.9 ↑)**	**53.6**	**78.6**
Few-shot + GPT-4	90.7	84.7	76.7	54.8	80.0
Ours (GPT-4)	**92.7**	**91.2**	**84.1**	**65.1(9.0 ↑)**	**86.2**

Test-suit accuracy					
Methods	**Easy**	**Medium**	**Hard**	**Extra-hard**	**TS**
Few-shot + GPT-3.5	90.3	67.6	42.6	26.4	62.3
Ours (GPT-3.5)	**90.7**	**77.3**	**52.8 (9.0 ↑)**	**27.1**	**68.3**
Few-shot + GPT-4	86.7	73.1	59.2	31.9	67.4
Ours (GPT-4)	**90.4**	**82.2**	**71.8**	**48.2(16.3 ↑)**	**76.9**

Analysis of each promptins

Methods	Easy	Medium	Hard	Extra-hard	EX
SL-prompt + GPT-3.5	**92.7**	79.3	68.8	48.2	75.7
CC-prompt + GPT-3.5	91.5	79.1	63.1	48.2	74.4
SL+CC prompt + GPT-3.5	88.7	**82.0**	**70.5**	**51.8**	**76.8**
SL-prompt + GPT-4	**96.0**	87.6	79.0	65.7	84.6
CC-prompt + GPT-4	93.1	86.5	78.4	59.0	82.3
SL+CC prompt + GPT-4	92.7	**89.4**	**80.7**	**66.9**	**85.1**

Analysis of each promptins

SL-prompt	CC-prompt	SL+CC prompt	SUM
	GPT-3.5		
✓	✗	✗	21
✗	✓	✗	27
✗	✗	✓	29
✗	✗	✗	157
✓	✓	✓	669
	GPT-4		
✓	✗	✗	9
✗	✓	✗	6
✗	✗	✓	16
✗	✗	✗	111
✓	✓	✓	791

(1034-157)/1034 = 84.8%

(1034-111)/1034 = 89.2%

Analysis of SL-prompt Structure

	SL-prompt structure	EX	TS
(a)	1. Identify relevant **table names**: $t_a, t_b, \ldots$ 2. Identify relevant **column names**: $c_1^{t_a}, c_2^{t_a}, c_1^{t_b}, \ldots$ 3. Write SQL: *SELECT* ...	66.9	61.3
(b)	1. Identify relevant **tables**: $(t_a : c_1^{t_a}, \ldots, c_{\lvert C \rvert}^{t_a}), (t_b : c_1^{t_b}, \ldots, c_{\lvert C \rvert}^{t_b}), \ldots$ 2. Identify relevant **column names**: $c_1^{t_a}, c_2^{t_a}, c_1^{t_b}, \ldots$ 3. Write SQL: *SELECT* ...	70.9	65.0
(c)	1. Identify relevant **tables**: $(t_a : c_1^{t_a}, \ldots, c_{\lvert C \rvert}^{t_a}), (t_b : c_1^{t_b}, \ldots, c_{\lvert C \rvert}^{t_b}), \ldots$ 2. Write SQL: *SELECT* ...	**75.7**	**68.5**

Analysis of CC-prompt Structure

	CC-prompt structure	EX	TS
(a)	1. Write the **SELECT** clause: *SELECT* ... 2. Write the **FROM** clause: *FROM* ... 3. Write other clauses: ... 4. Write SQL: *SELECT* ...	73.5	64.4
(b)	1. Write the **FROM** clause: *FROM* ... 2. Write the **SELECT** clause: *SELECT* ... 3. Write other clauses: ... 4. Write SQL: *SELECT* ...	73.7	64.6
(c)	1. Write the **FROM** clause: *FROM* ... 2. Write other clauses: ... 3. Write the **SELECT** clause: *SELECT* ... 4. Write SQL: *SELECT* ...	**74.4**	**65.1**

Our Contributions and Findings

- Tailored prompting methods for Text-to-SQL parsing
  - *SL-prompt, CC-prompt, SL+CC prompt*
  - *A in-deep analysis of promptings structure*

- A guideline for designing prompting methods for Text-to-SQL parsing
  - *More details in our paper!*

Thank you!

Email: tanzhao325@gmail.com