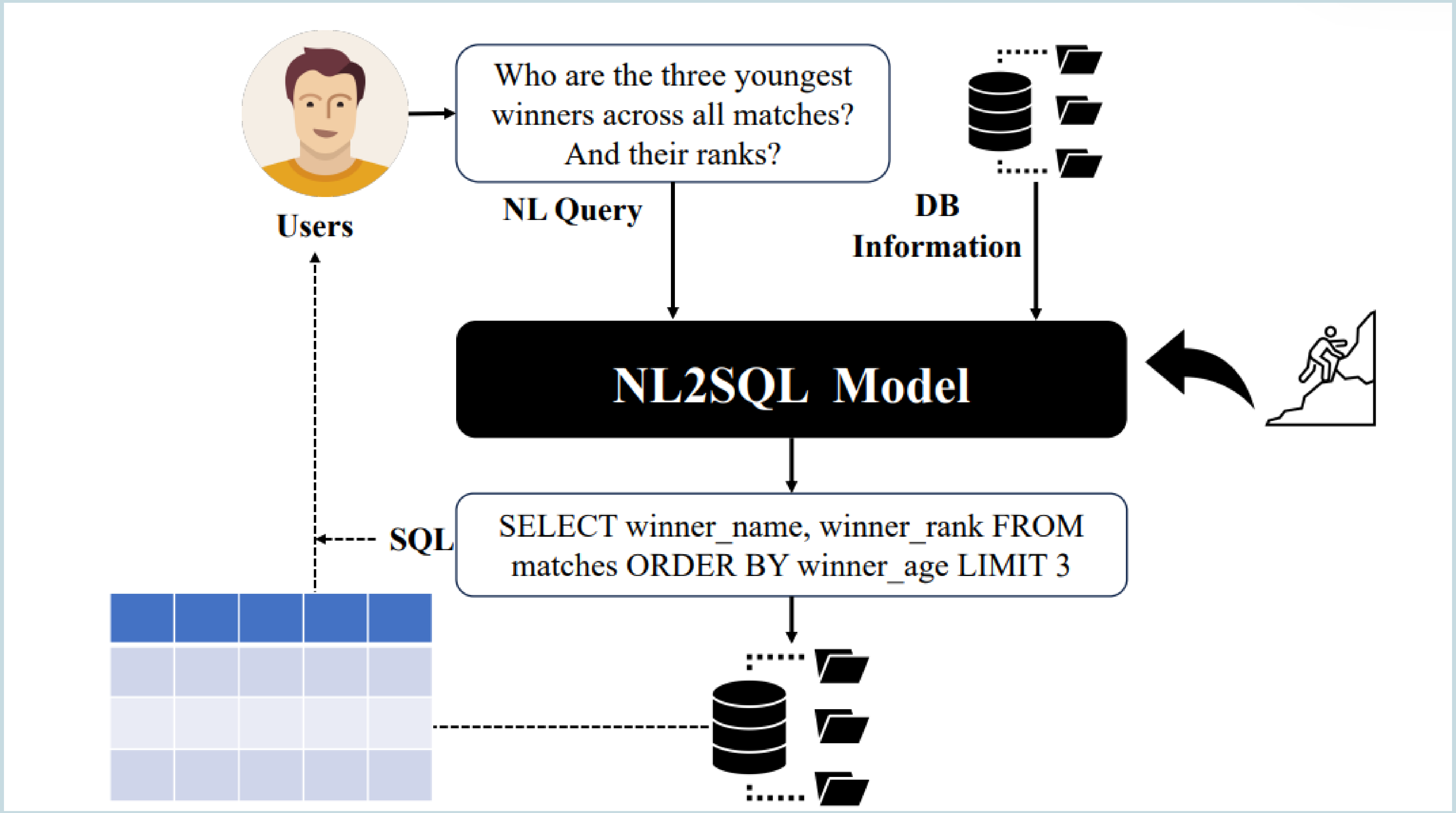
NL2SQL USING LARGE LANGUAGE MODEL

Anubhav Binit

OUTLINE

WHAT IS NATURAL LANGUAGE TO SQL (NL2SQL)?


Who are the three youngest winners across all matches? And their ranks?
Users
NL Query
DB Information
NL2SQL Model
SQL
SELECT winner_name, winner_rank FROM matches ORDER BY winner_age LIMIT 3

NL2SQL TASK CHALLENGES

C1. Ambiguous NL Query

Natural language queries can be vague or have multiple interpretations.

Example: *"List the top customers."* → Top by what? Revenue, number of purchases, or something else?

C2. Database Schema Dependency

The correct SQL depends on the database structure, which varies by case.

Example: Same NL query may need different SQL if column or table names differ across databases.

C3. Complex Schema

Large or deeply relational schemas make it hard to find the right tables and columns.

Example: A schema with 50+ tables and foreign keys is hard to navigate

C4. Multiple Possible SQL Queries

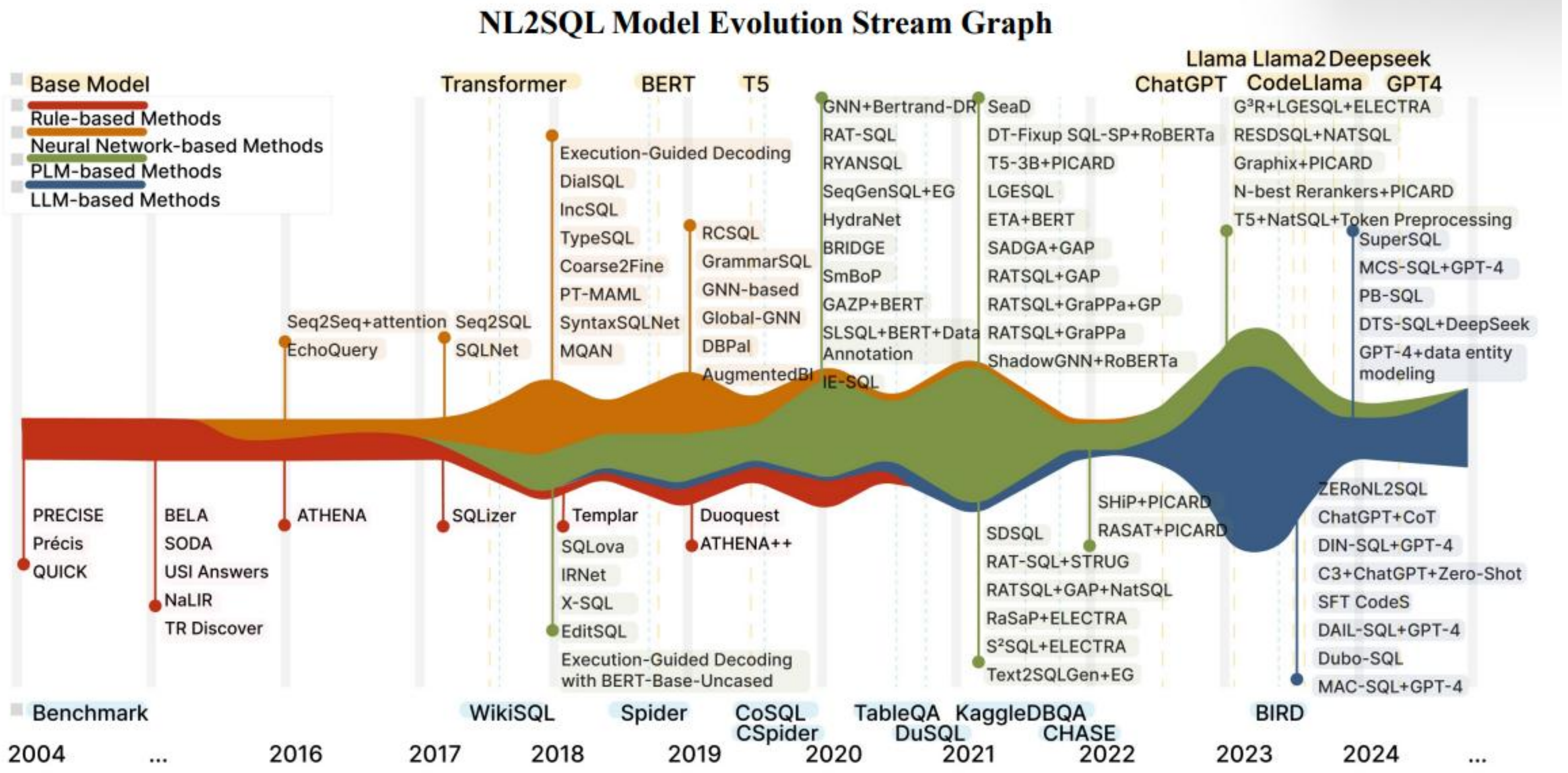
Different SQL queries can give valid answers to the same NL query.

Example: *"Show all employees in the Sales department."* → Can use JOINs or subqueries.

C5. Database Domain Adaptation

A model trained on one domain (e.g., movies) may not perform well on another (e.g., healthcare).

Example: Column names and query patterns change between domains.



Liu, X., Shen, S., Li, B., Ma, P., Jiang, R., Zhang, Y., Fan, J., Li, G., Tang, N., & Luo, Y. (2024). *A Survey of NL2SQL with Large Language Models: Where are we, and where are we going?*

AN OVERVIEW OF LANGUAGE MODEL-POWERED NL2SQL

- Pre-Processing
  - Schema Linking
  - Database Content Retrieval
  - Additional Information Acquisition

- NL2SQL Translation Methods
  - Encoding Strategy
  - Decoding Strategy
  - Task-specific Prompt Strategies
  - Intermediate Representation

- Post-Processing
  - Correction
  - Consistency
  - Execution- Guided
  - N-best Rerankers

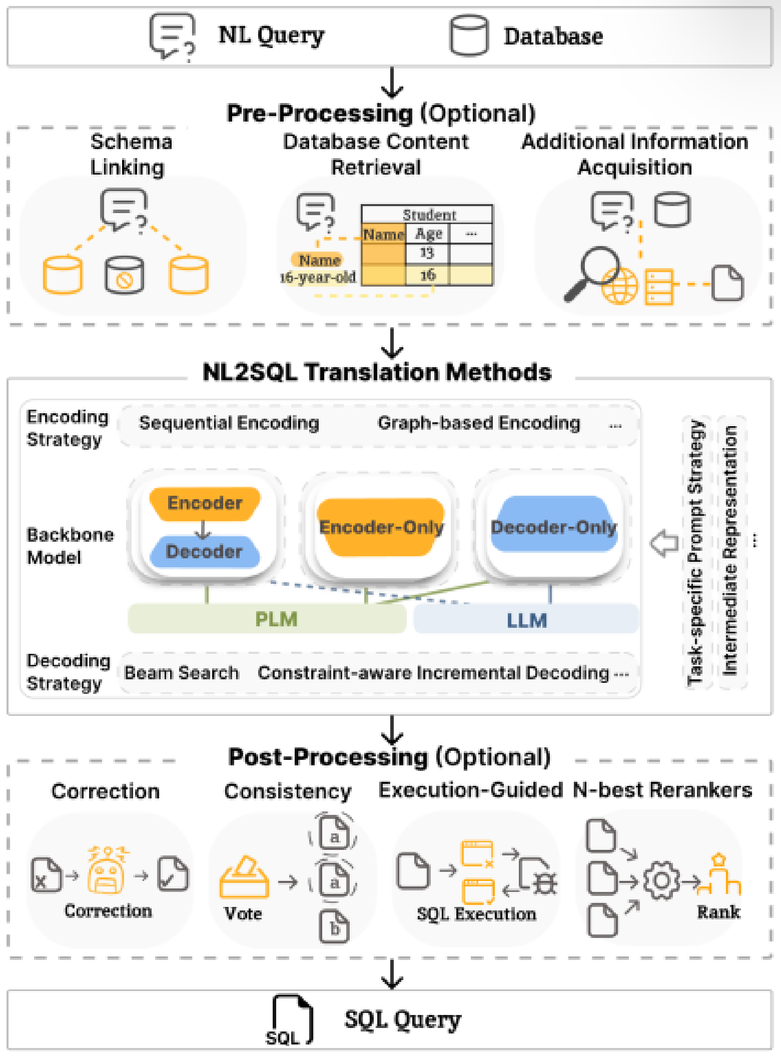
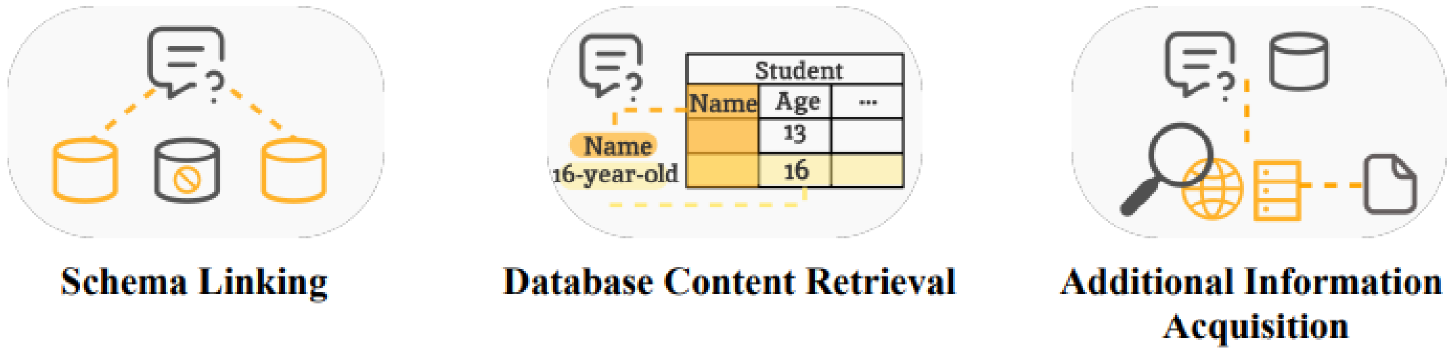


Figure: An Overview of NL2SQL Methods in the LM Era

PRE-PROCESSING

- The pre-processing ensures the accurate mapping and processing of key information within the limited input, by identifying the tables and columns related to the given NL query.

- The pre-processing serves as an enhancement to the model's inputs in the NL2SQL translation.


Student
Name
Age
...
13
16
Name
16-year-old
Schema Linking
Database Content Retrieval
Additional Information Acquisition

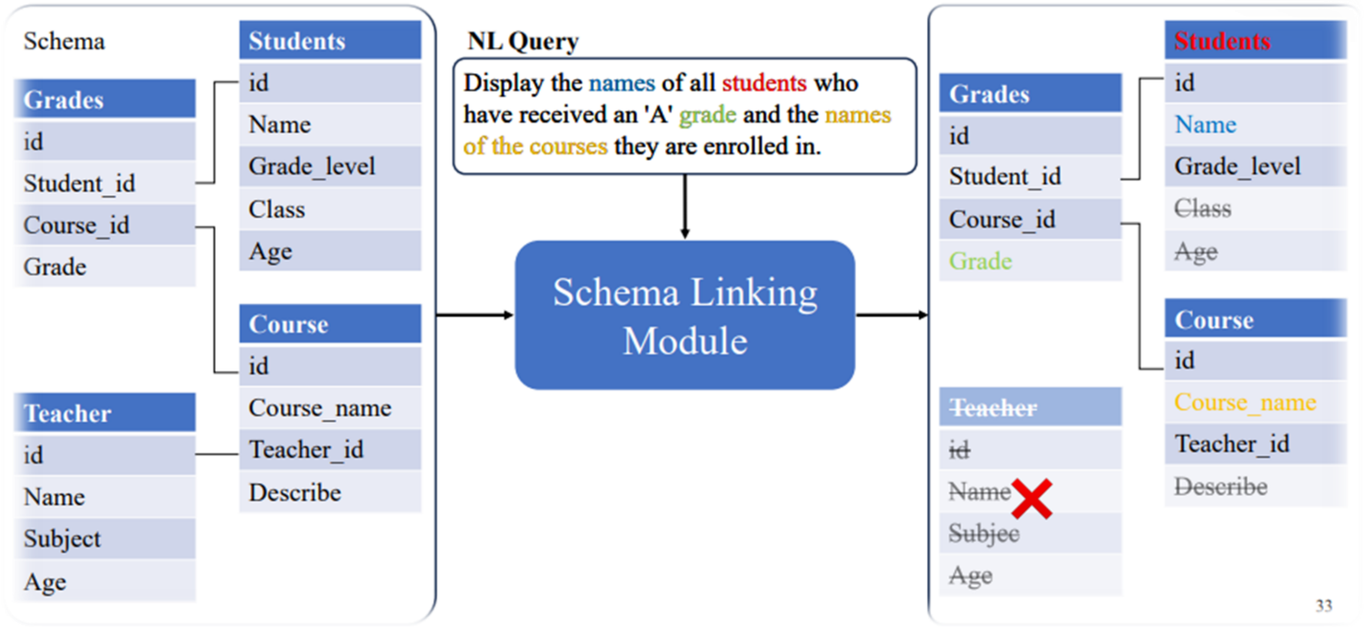
SCHEMA LINKING

Motivation:

- Schema complexity and presence of irrelevant tables with NL in database.
- The model has input limitations, e.g., the input limit of ChatGPT is 4096 tokens.

Goal:

- Identify the tables and columns related to the given NL.



METHOD CLASSIFICATION OF SCHEME LINKING

Method classification：

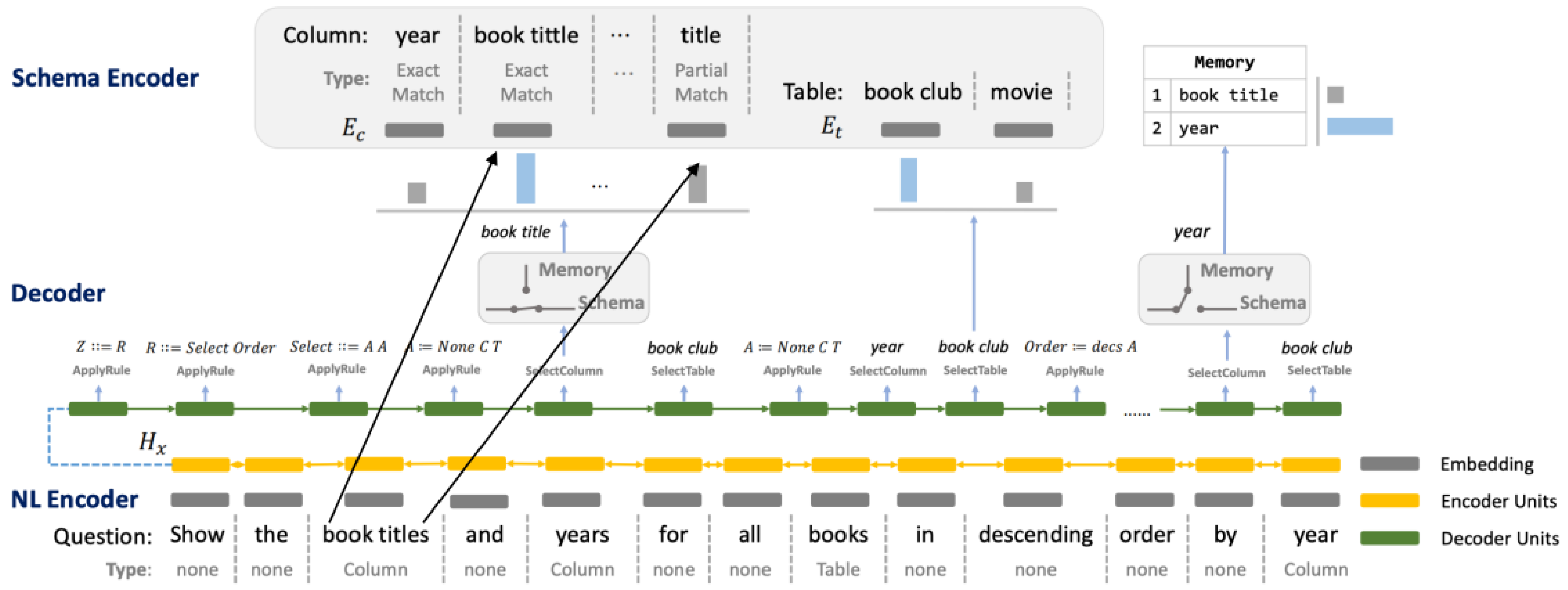
- String Matching-based Schema Linking
- Neural Network-based Schema Linking
- In-Context Learning for Schema Linking

STRING MATCHING-BASED SCHEMA LINKING

IRNet designs string-based schema linking and an intermediate representation called SemQL.

• It uses n-grams (1-6 length) extracted from user queries as query candidates

• It evaluates the string-level similarity between the query candidates and the database schema, then classifies them as Exact or Partial matches.



Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, Dongmei Zhang:
Towards Complex Text-to-SQL in Cross-Domain Database with Intermediate Representation. ACL (1) 2019: 4524-4535

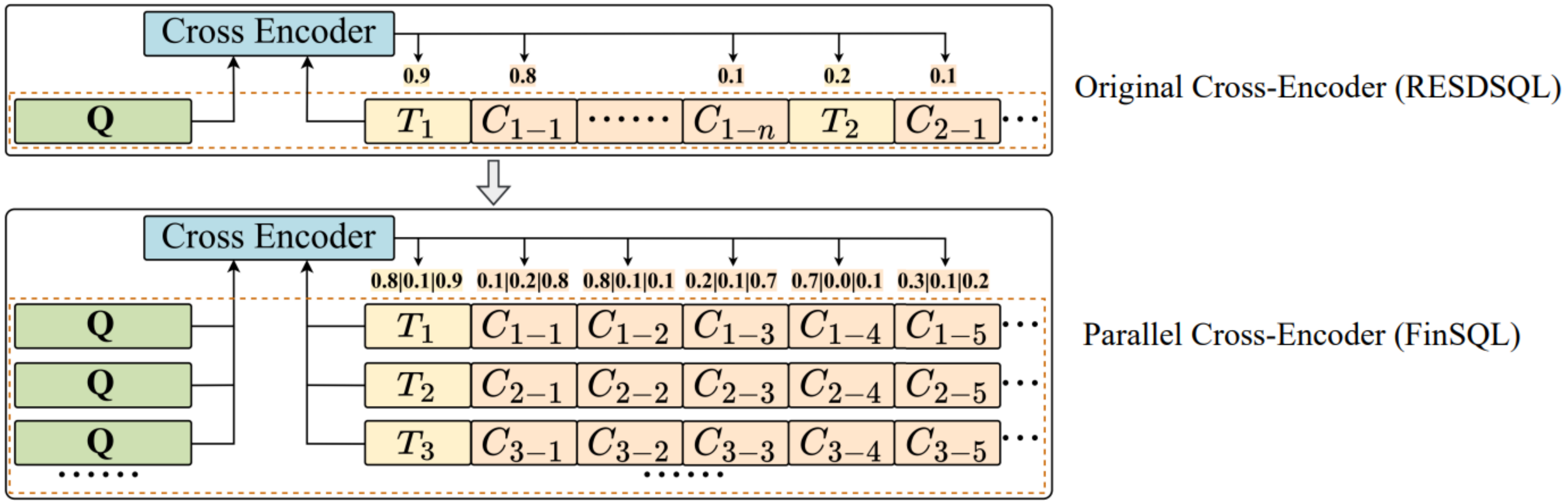
NEURAL NETWORK-BASED SCHEMA LINKING

RESDSQL proposes a ranking-enhanced encoding and skeleton-aware decoding framework.

- A cross-encoder is trained to classify tables and columns based on the input query.

FinSQL, a model-agnostic LLMs-based NL2SQL framework for financial analysis.

- Organizing the tables into a batch.
- Devising a Parallel Cross-Encoder to retrieve relevant tables and columns from hundreds of schema items


Cross Encoder
0.9
0.8
0.1
0.2
0.1
Q
T_1
C_{1-1}
C_{1-n}
T_2
C_{2-1}
Original Cross-Encoder (RESDSQL)
Cross Encoder
0.8|0.1|0.9
0.1|0.2|0.8
0.8|0.1|0.1
0.2|0.1|0.7
0.7|0.0|0.1
0.3|0.1|0.2
Q
T_1 C_{1-1} C_{1-2} C_{1-3} C_{1-4} C_{1-5}
T_2 C_{2-1} C_{2-2} C_{2-3} C_{2-4} C_{2-5}
T_3 C_{3-1} C_{3-2} C_{3-3} C_{3-4} C_{3-5}
Parallel Cross-Encoder (FinSQL)

Chao Zhang, Yuren Mao, Yijiang Fan, Yu Mi, Yunjun Gao, Lu Chen, Dongfang Lou, Jinshu Lin: FinSQL: Model-Agnostic LLMs-based Text-to-SQL Framework for Financial Analysis. SIGMOD Conference Companion 2024: 93-105

IN-CONTEXT LEARNING FOR SCHEMA LINKING

MAC-SQL is a multi-agent collaborative framework for NL2SQL.
• The Selector agent performs the schema linking task.
• The Selector is activated only when the length of the database schema prompt exceeds a specified threshold

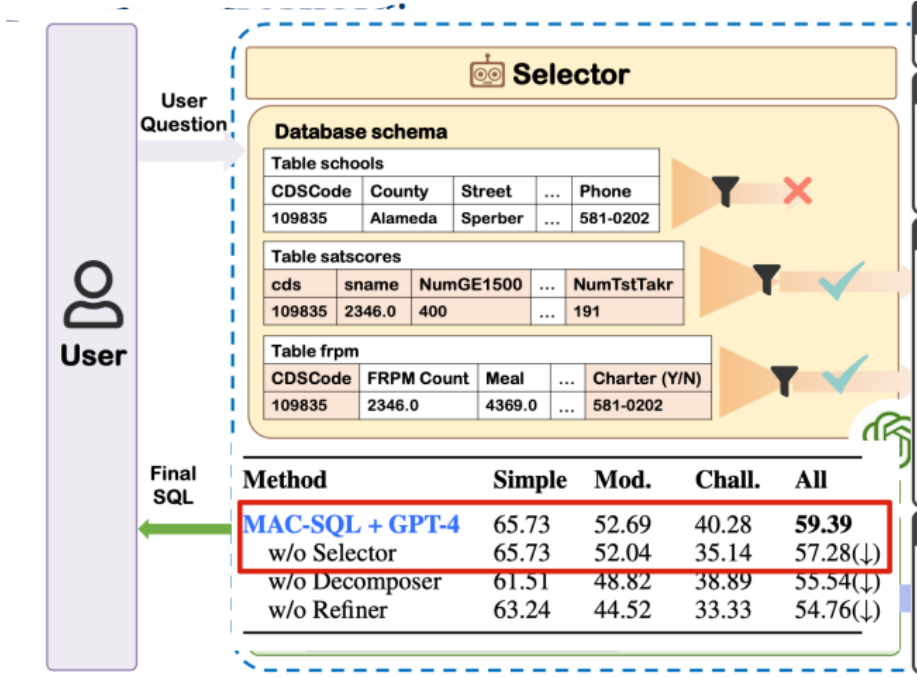


Table schools

CDSCode	County	Street	...	Phone
109835	Alameda	Sperber	...	581-0202

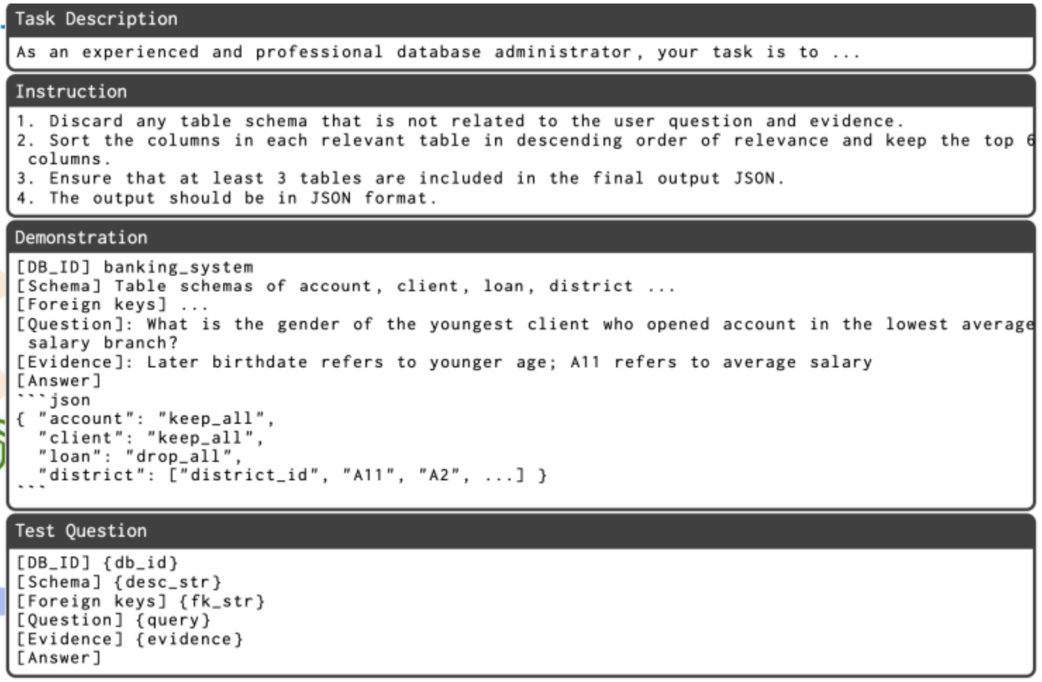
Table satscores

cds	sname	NumGE1500	...	NumTstTakr
109835	2346.0	400	...	191

Table frpm

CDSCode	FRPM Count	Meal	...	Charter (Y/N)
109835	2346.0	4369.0	...	581-0202

Method	Simple	Mod.	Chall.	All
MAC-SQL + GPT-4	65.73	52.69	40.28	**59.39**
w/o Selector	65.73	52.04	35.14	57.28(↓)
w/o Decomposer	61.51	48.82	38.89	55.54(↓)
w/o Refiner	63.24	44.52	33.33	54.76(↓)



~~~
Task Description
As an experienced and professional database administrator, your task is to ...

Instruction
1. Discard any table schema that is not related to the user question and evidence.
2. Sort the columns in each relevant table in descending order of relevance and keep the top 6
 columns.
3. Ensure that at least 3 tables are included in the final output JSON.
4. The output should be in JSON format.

Demonstration
[DB_ID] banking_system
[Schema] Table schemas of account, client, loan, district ...
[Foreign keys] ...
[Question]: What is the gender of the youngest client who opened account in the lowest average
 salary branch?
[Evidence]: Later birthdate refers to younger age; A11 refers to average salary
[Answer]
```json
{ "account": "keep_all",
  "client": "keep_all",
  "loan": "drop_all",
  "district": ["district_id", "A11", "A2", ...] }
```

Test Question
[DB_ID] {db_id}
[Schema] {desc_str}
[Foreign keys] {fk_str}
[Question] {query}
[Evidence] {evidence}
[Answer]
~~~

Bing Wang, Changyu Ren, Jian Yang, Xinnian Liang, Jiaqi Bai, Qian-Wen Zhang, Zhao Yan, Zhoujun Li: MAC-SQL: A Multi-Agent Collaborative Framework for Text-to-SQL. CoRR abs/2312.11242 (2023)

DATABASE CONTENT RETRIEVAL

Motivation:

- Due to the large databases and input limitations of the model, retrieving cell values is resource-intensive.
- Effectively handle some SQL clauses such as WHERE and JOIN .

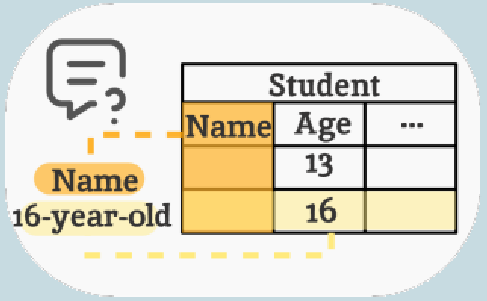
Goal:

- Efficiently retrieve cell values related to the given NL.

METHOD CLASSIFICATION OF DB CONTENT RETRIEVAL

Method classification ：

- String Matching-based Database Content Retrieval
- Neural Network-based Database Content Retrieval
- Index Strategy for Database Content Retrieval

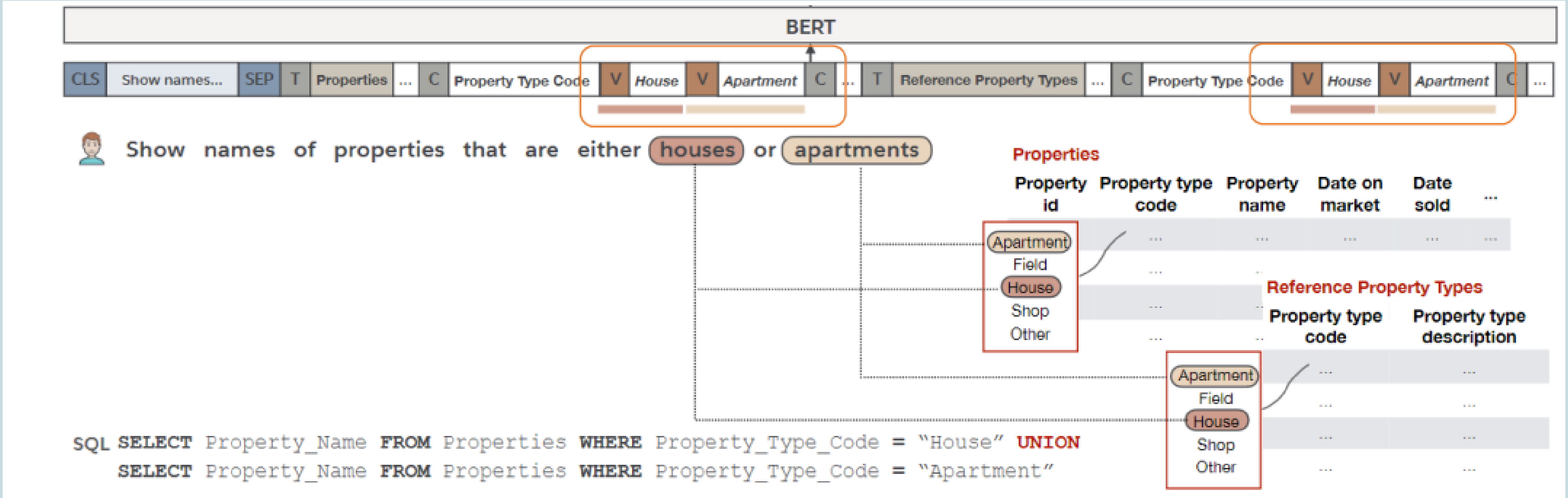


Database Content Retrieval

STRING MATCHING-BASED DATABASE CONTENT RETRIEVAL

BRIDGE designs an anchor text matching to automatically extract cell values mentioned in the NL.

• It uses a heuristic method to calculate the maximum sequence match between the problem and the cell values to determine the matching boundary.



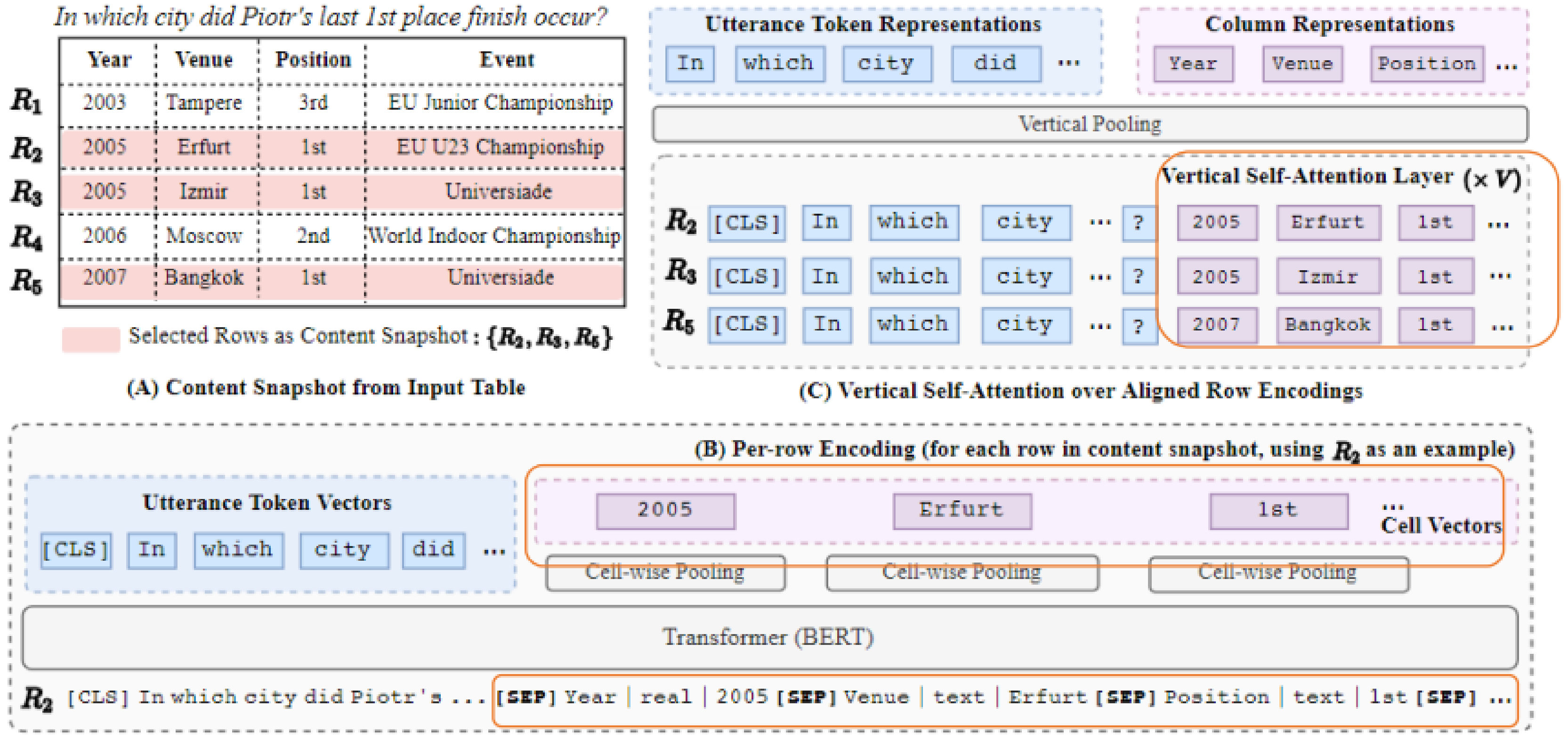
```
SQL SELECT Property_Name FROM Properties WHERE Property_Type_Code = "House" UNION
    SELECT Property_Name FROM Properties WHERE Property_Type_Code = "Apartment"
```

Xi Victoria Lin, Richard Socher, Caiming Xiong:
Bridging Textual and Tabular Data for Cross-Domain Text-to-SQL Semantic Parsing. EMNLP (Findings) 2020: 4870-4888

NEURAL NETWORK-BASED DATABASE CONTENT RETRIEVAL

TABERT uses a method called database content snapshots.

- It uses an attention mechanism to manage information between cell values across different rows.



In which city did Piotr's last 1st place finish occur?

	Year	Venue	Position	Event
R_1	2003	Tampere	3rd	EU Junior Championship
R_2	2005	Erfurt	1st	EU U23 Championship
R_3	2005	Izmir	1st	Universiade
R_4	2006	Moscow	2nd	World Indoor Championship
R_5	2007	Bangkok	1st	Universiade

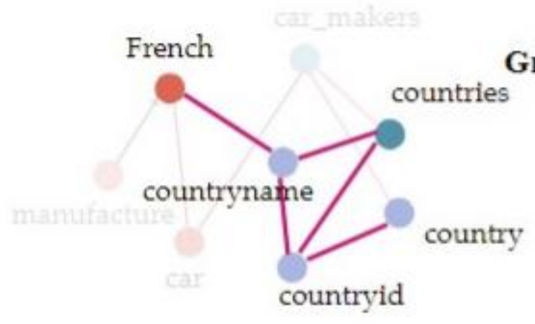
Selected Rows as Content Snapshot : $\{R_2, R_3, R_5\}$
(A) Content Snapshot from Input Table
Utterance Token Representations
In which city did ...
Column Representations
Year Venue Position ...
Vertical Pooling
Vertical Self-Attention Layer (× V)
R_2 [CLS] In which city ... ? 2005 Erfurt 1st ...
R_3 [CLS] In which city ... ? 2005 Izmir 1st ...
R_5 [CLS] In which city ... ? 2007 Bangkok 1st ...
(C) Vertical Self-Attention over Aligned Row Encodings
(B) Per-row Encoding (for each row in content snapshot, using R_2 as an example)
Utterance Token Vectors
[CLS] In which city did ...
2005 Erfurt 1st ...
Cell Vectors
Cell-wise Pooling
Cell-wise Pooling
Cell-wise Pooling
Transformer (BERT)
R_2 [CLS] In which city did Piotr's ... [SEP] Year | real | 2005 [SEP] Venue | text | Erfurt [SEP] Position | text | 1st [SEP] ...

Pengcheng Yin, Graham Neubig, Wen-tau Yih, Sebastian Riedel:
TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data. ACL 2020: 8413-8426

NEURAL NETWORK-BASED DATABASE CONTENT RETRIEVAL

Graphix-T5 improves structural reasoning capabilities by modeling the relationship between cell values and the NL query.

• it uses a value-match relation that defines the relationship between cell values and questions.



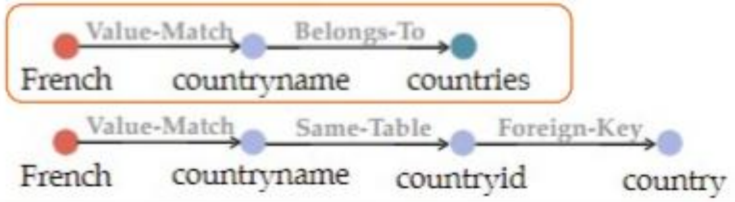
Question: *How many French car manufacturers are there?*

T5-3B: SELECT COUNT(*) FROM car_makers WHERE **country** = "France"

Graphix-T5-3B: SELECT COUNT(*) FROM car_makers AS T1 JOIN countries AS T2 ON T1.country = T2.countryid WHERE T2.countryname = "France"

Gold: SELECT COUNT(*) FROM **car_makers** AS T1 JOIN **countries** AS T2 ON T1.**country** = T2.**countryid** WHERE T2.**countryname** = 'France';

Muti-hop Path



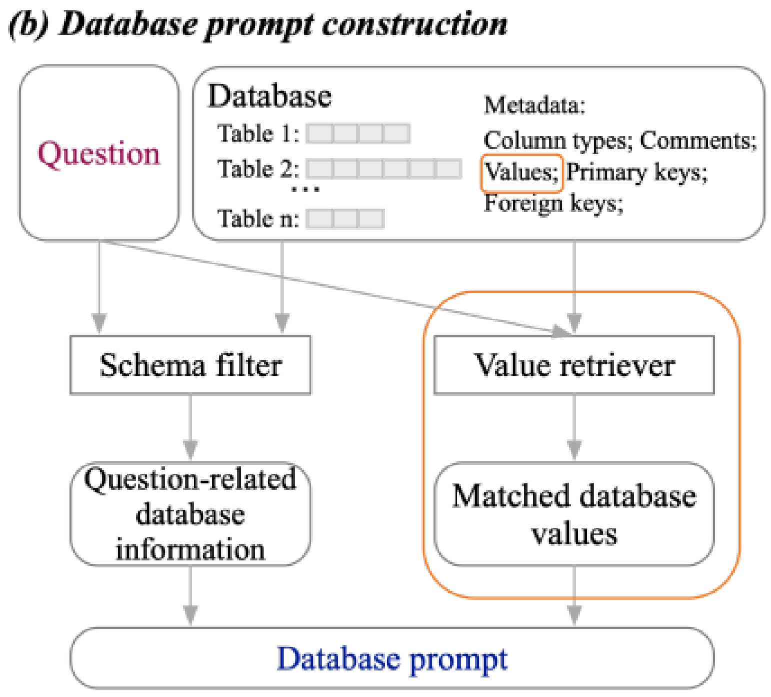
Question	Column	VALUE-MATCH	x is part of the candidate cell values of column y.

Jinyang Li, Binyuan Hui, Reynold Cheng, Bowen Qin, Chenhao Ma, Nan Huo, Fei Huang, Wenyu Du, Luo Si, Yongbin Li: Graphix-T5: Mixing Pre-Trained Transformers with Graph-Aware Layers for Text-to-SQL Parsing. CoRR abs/2301.07507 (2023)

INDEX STRATEGY FOR DATABASE CONTENT RETRIEVAL

CodeS introduces a coarse-to-fine cell value matching approach.

- It builds the index for all values using BM25.
- The index identifies candidate values that are relevant to NL.

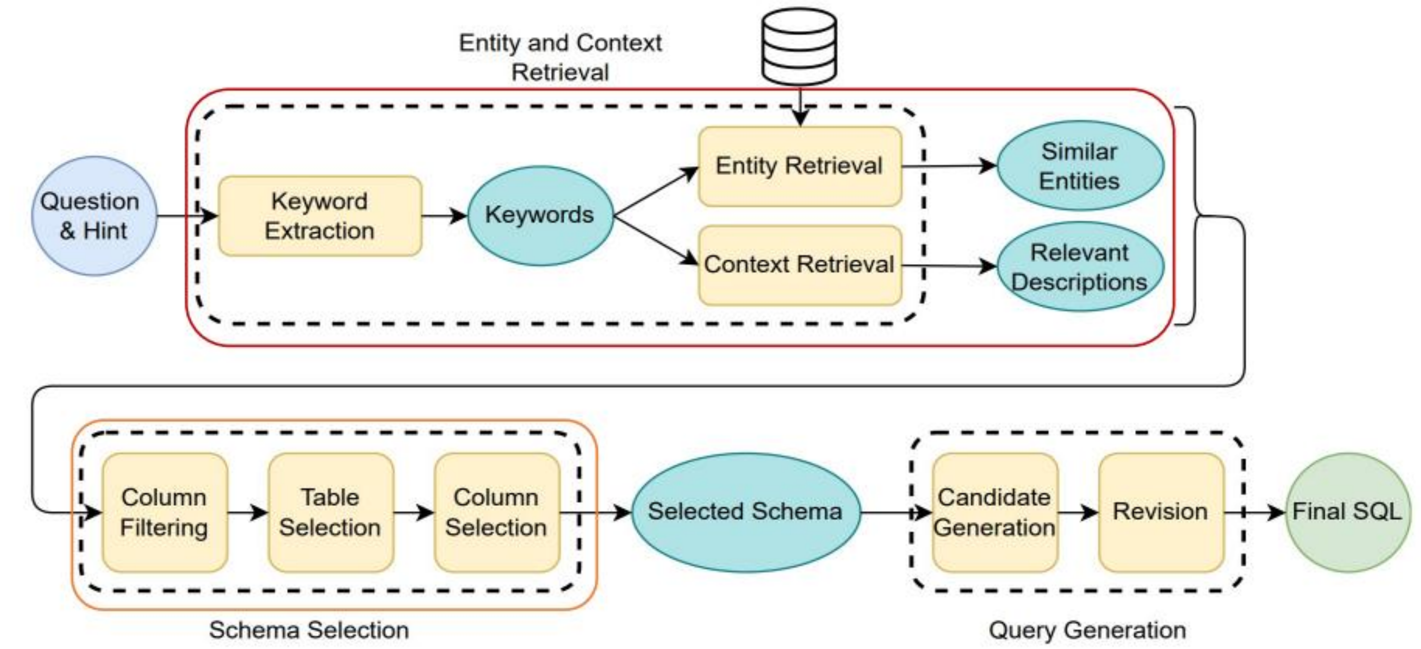


Haoyang Li, Jing Zhang, Hanbing Liu, Ju Fan, Xiaokang Zhang, Jun Zhu, Renjie Wei, Hongyan Pan, Cuiping Li, Hong Chen: CodeS: Towards Building Open-source Language Models for Text-to-SQL. CoRR abs/2402.16347 (2024)

INDEX STRATEGY FOR DATABASE CONTENT RETRIEVAL

CHESS uses a Locality-sensitive Hashing algorithm for approximate nearest neighbor searches.

- It indexes unique cell values to quickly identify the top similar values related to the NL query



Shayan Talaei, Mohammadreza Pourreza, Yu-Chen Chang, Azalia Mirhoseini, Amin Saberi:
CHESS: Contextual Harnessing for Efficient SQL Synthesis. CoRR abs/2405.16755 (2024) Shayan Talaei, Mohammadreza

ENCODING STRATEGY

Motivation:

• Converts unstructured and semi-structured data into a form that can be processed for generating SQL queries.
• The encoding process captures the semantic meaning of the NL input and the structural information of the database schema, enabling the model to understand and map the user's intent to the corresponding SQL query.

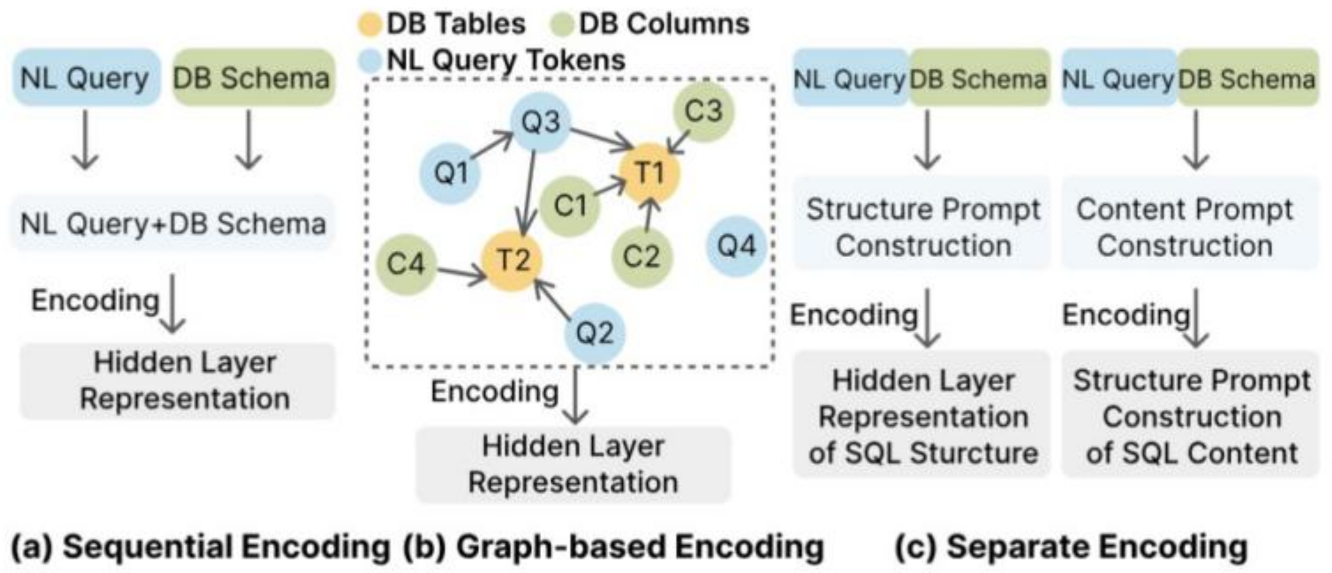
Goal:

• Transform NL and database schema into a structured format that can be effectively utilized by a language model.

METHOD CLASSIFICATION OF ENCODING STRATEGY

Method classification ：

- Sequential Encoding Strategy
- Graph-based Encoding Strategy
- Separate Encoding Strategy

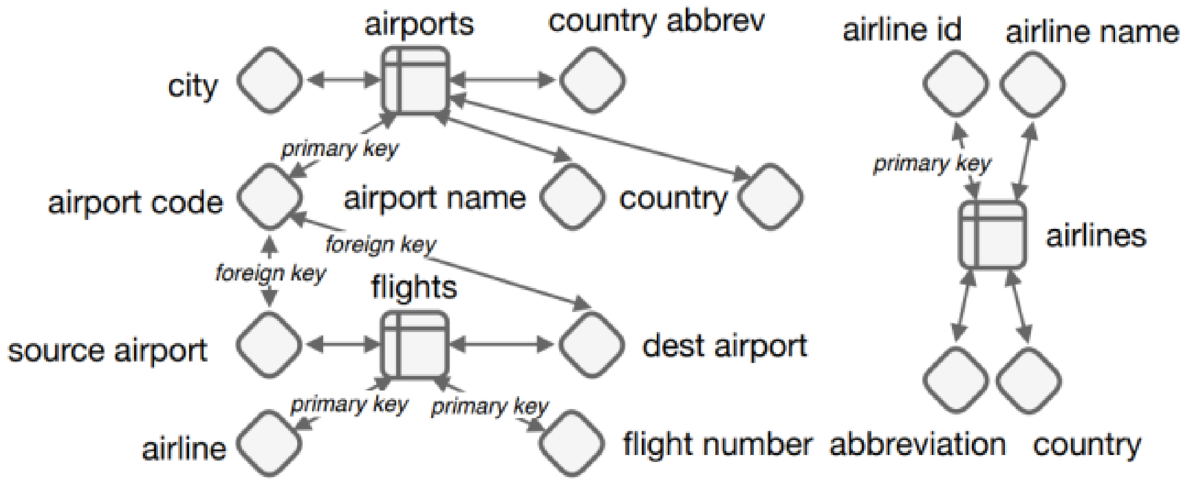


An Overview of the Encoding Strategies

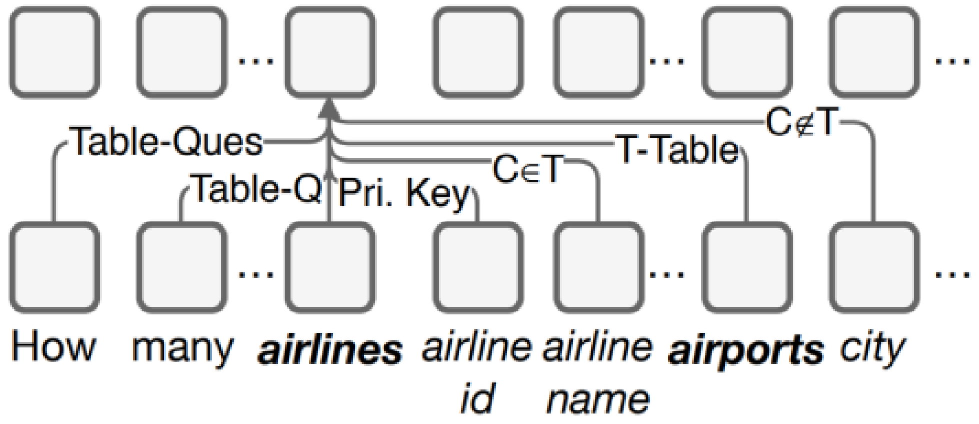
GRAPH-BASED ENCODING STRATEGY

RAT-SQL introduces a relation-aware self-attention mechanism.
• It allows the model to explicitly consider and utilize predefined relational information when jointly encoding the question and the database schema.
• These relationships are represented as a graph structure, and it can more effectively capture the structural information in the schema and its alignment with the NL query

An illustration of an example schema as a graph G



One RAT layer in the schema encoder.

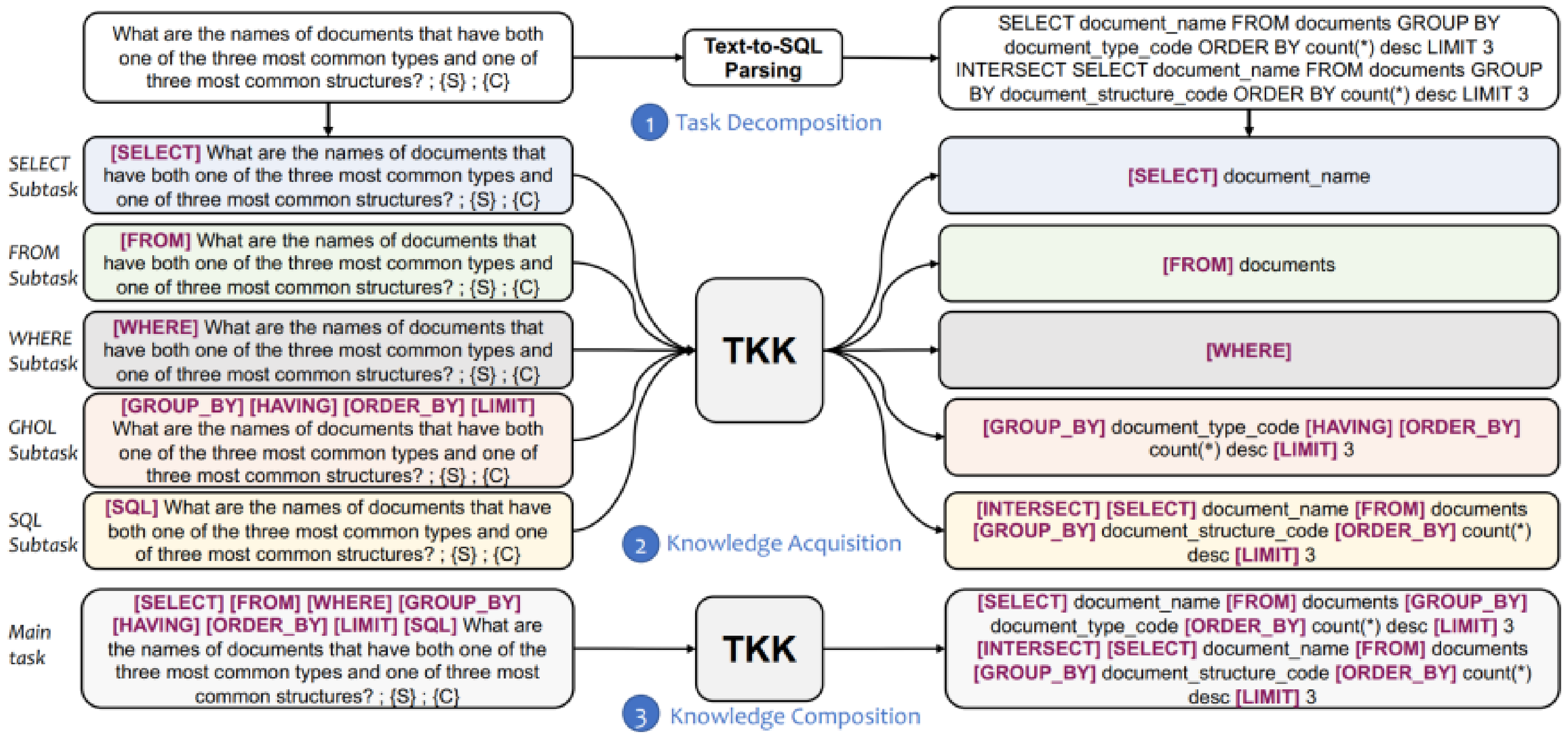


Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, Matthew Richardson:
RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers. ACL 2020: 7567-7578

SEPARATE ENCODING STRATEGY

TKK employs task decomposition and multi-task learning strategies in encoding.

- It breaks down the complex NL2SQL task into multiple subtasks.
- In this way, it can progressively acquire and combine knowledge.


What are the names of documents that have both one of the three most common types and one of three most common structures? ; {S} ; {C}
Text-to-SQL Parsing
SELECT document_name FROM documents GROUP BY document_type_code ORDER BY count(*) desc LIMIT 3 INTERSECT SELECT document_name FROM documents GROUP BY document_structure_code ORDER BY count(*) desc LIMIT 3
1 Task Decomposition
SELECT Subtask
[SELECT] What are the names of documents that have both one of the three most common types and one of three most common structures? ; {S} ; {C}
FROM Subtask
[FROM] What are the names of documents that have both one of the three most common types and one of three most common structures? ; {S} ; {C}
WHERE Subtask
[WHERE] What are the names of documents that have both one of the three most common types and one of three most common structures? ; {S} ; {C}
GHOL Subtask
[GROUP_BY] [HAVING] [ORDER_BY] [LIMIT] What are the names of documents that have both one of the three most common types and one of three most common structures? ; {S} ; {C}
SQL Subtask
[SQL] What are the names of documents that have both one of the three most common types and one of three most common structures? ; {S} ; {C}
TKK
[SELECT] document_name
[FROM] documents
[WHERE]
[GROUP_BY] document_type_code [HAVING] [ORDER_BY] count(*) desc [LIMIT] 3
[INTERSECT] [SELECT] document_name [FROM] documents [GROUP_BY] document_structure_code [ORDER_BY] count(*) desc [LIMIT] 3
2 Knowledge Acquisition
Main task
[SELECT] [FROM] [WHERE] [GROUP_BY] [HAVING] [ORDER_BY] [LIMIT] [SQL] What are the names of documents that have both one of the three most common types and one of three most common structures? ; {S} ; {C}
TKK
[SELECT] document_name [FROM] documents [GROUP_BY] document_type_code [ORDER_BY] count(*) desc [LIMIT] 3 [INTERSECT] [SELECT] document_name [FROM] documents [GROUP_BY] document_structure_code [ORDER_BY] count(*) desc [LIMIT] 3
3 Knowledge Composition

Chang Gao, Bowen Li, Wenxuan Zhang, Wai Lam, Binhua Li, Fei Huang, Luo Si, Yongbin Li: Towards Generalizable and Robust Text-to-SQL Parsing. EMNLP (Findings) 2022: 2113-2125

DECODING STRATEGY

Motivation:

• The choice of decoding strategy directly affects the quality and performance of the generated SQL queries.
• An excellent decoding strategy not only produces syntactically correct SQL queries but also ensures that the semantics of the SQL queries align with the NL and can even optimize the execution efficiency of the queries.

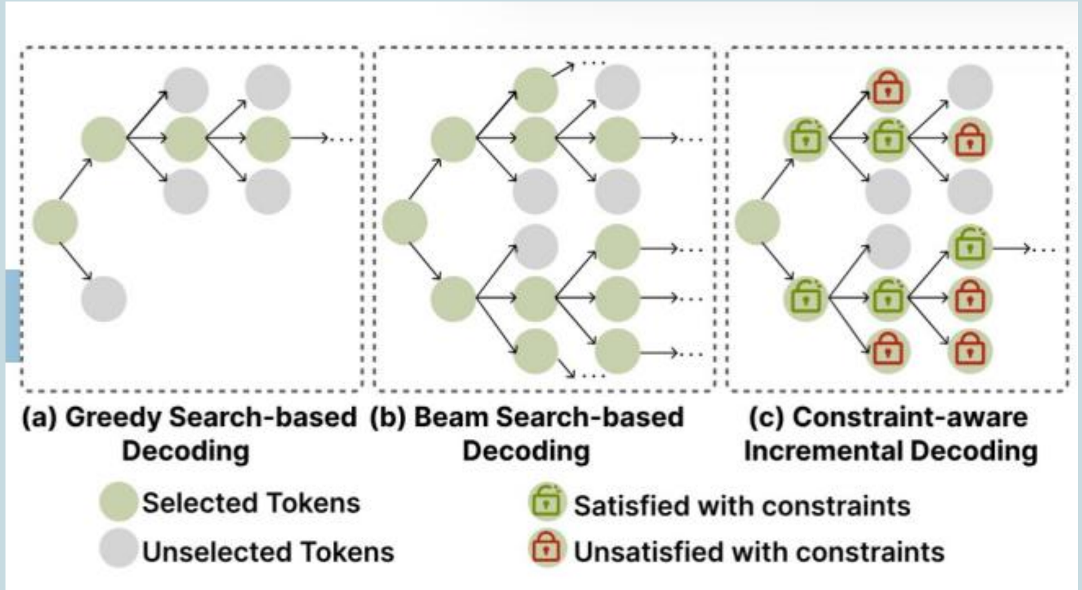
Goal:

• Convert the representations generated by the encoder into the target SQL queries.

METHOD CLASSIFICATION OF DECODING STRATEGY

Method classification ：

- Greedy search-based decoding strategy
- Beam search-based decoding strategy
- Constraint-aware incremental decoding strategy


(a) Greedy Search-based Decoding
(b) Beam Search-based Decoding
(c) Constraint-aware Incremental Decoding
Selected Tokens
Unselected Tokens
Satisfied with constraints
Unsatisfied with constraints

An Overview of the Decoding Strategies

GREEDY SEARCH-BASED DECODING STRATEGY

The greedy search-based decoding strategy is a simple and fast approach for decoding.

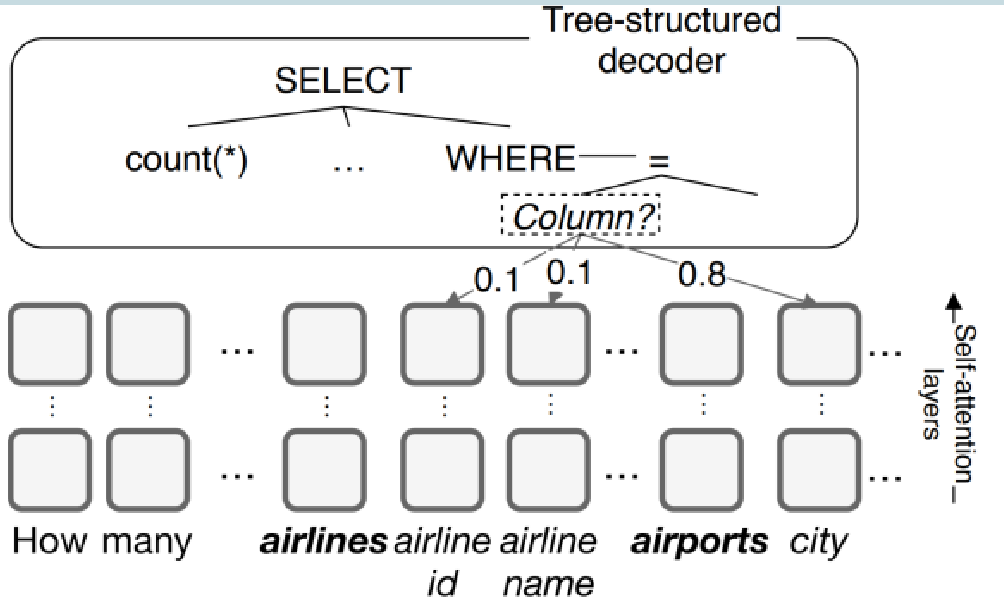
• At each decoding step, greedy search selects the token with the highest current probability as the output.

• This strategy builds the final output sequence by continuously choosing the locally optimal solution.

• Since the default decoding strategy of GPT series models (e.g., GPT-4) is greedy search-based decoding, NL2SQL solutions based on GPT fall into this category.

BEAM SEARCH-BASED DECODING STRATEGY

RAT-SQL combines relation-aware graph structure encoding and generation techniques.
• During the decoding process, RAT-SQL uses beam search to generate multiple candidate SQL queries
• These queries are then reranked, and the optimal query is selected based on graph structure information



Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, Matthew Richardson:
RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers. ACL 2020: 7567-7578

CONSTRAINT-AWARE INCREMENTAL DECODING STRATEGY

PICARD (Parsing Incrementally for Constrained Auto-Regressive Decoding) introduces the constraint-aware incremental decoding strategy.

• The constraint-aware incremental decoding strategy is specifically designed for NL2SQL tasks.

• This strategy aims to ensure the generation of syntactically correct SQL queries by incorporating constraints during the decoding process.

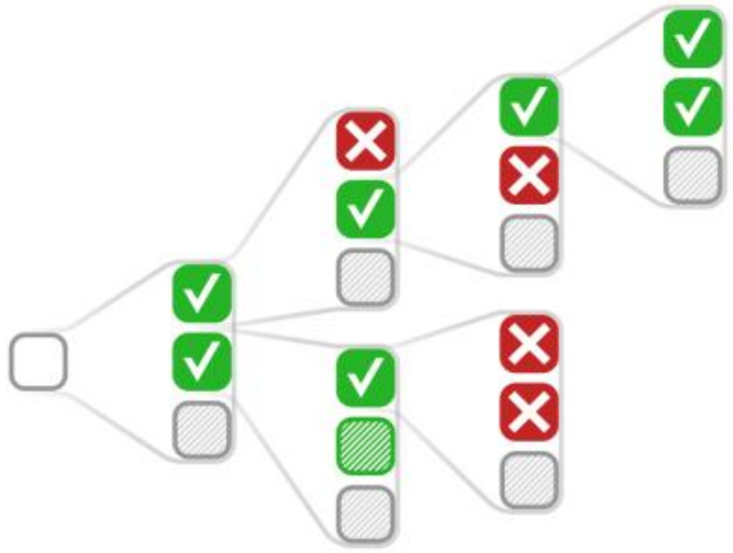


Illustration of constrained beam search with beam size 2 and PICARD.

Torsten Scholak, Nathan Schucher, Dzmitry Bahdanau:
PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models. EMNLP (1) 2021: 9895-9901

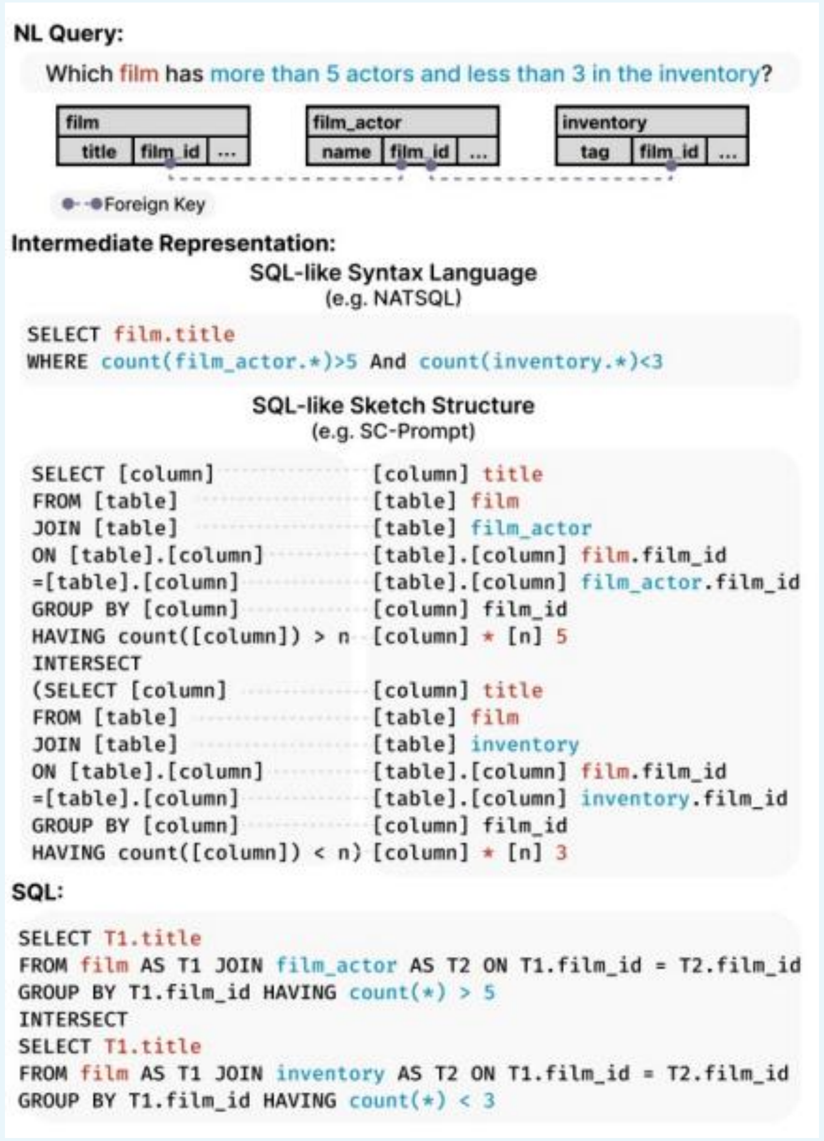
INTERMEDIATE REPRESENTATION

Motivation:

• Design a grammar-free intermediate representation compared to SQL as the bridge between the "free-form" NL query and the "constrained and formal" SQL query.

Goal:

• Capture the essential components and relationships of an NL query without the strict syntax rules of SQL.



```
SELECT film.title
WHERE count(film_actor.*)>5 And count(inventory.*)<3
```

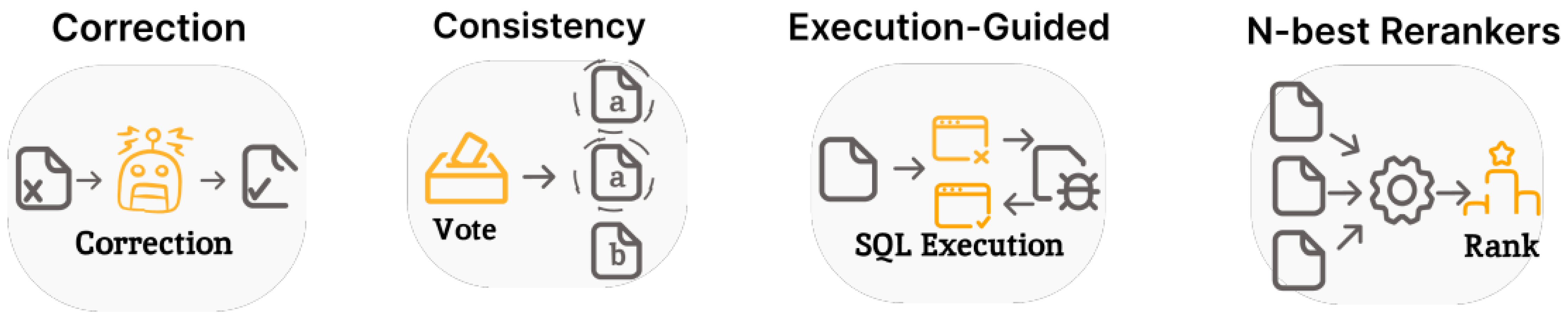
```
SELECT [column]                [column] title
FROM [table]                   [table] film
JOIN [table]                   [table] film_actor
ON [table].[column]            [table].[column] film.film_id
=[table].[column]              [table].[column] film_actor.film_id
GROUP BY [column]              [column] film_id
HAVING count([column]) > n     [column] * [n] 5
INTERSECT
(SELECT [column]               [column] title
FROM [table]                   [table] film
JOIN [table]                   [table] inventory
ON [table].[column]            [table].[column] film.film_id
=[table].[column]              [table].[column] inventory.film_id
GROUP BY [column]              [column] film_id
HAVING count([column]) < n)    [column] * [n] 3
```

```
SELECT T1.title
FROM film AS T1 JOIN film_actor AS T2 ON T1.film_id = T2.film_id
GROUP BY T1.film_id HAVING count(*) > 5
INTERSECT
SELECT T1.title
FROM film AS T1 JOIN inventory AS T2 ON T1.film_id = T2.film_id
GROUP BY T1.film_id HAVING count(*) < 3
```

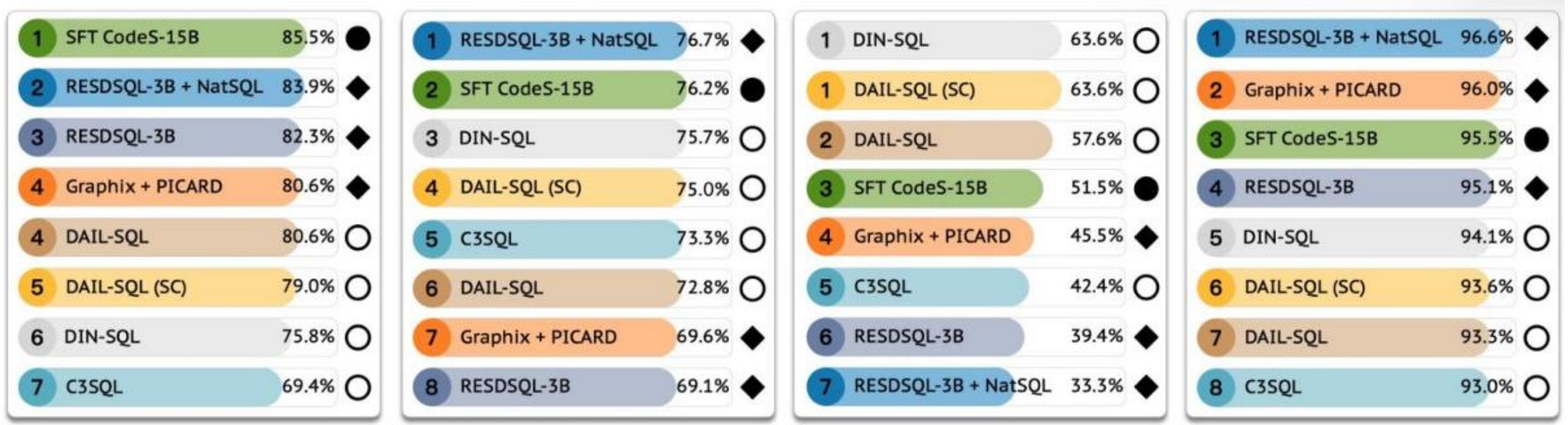
POST PROCESSING

Motivation:

- Post-processing is a crucial step to refine the generated SQL queries, ensuring they meet user expectations more accurately.
- This involves enhancing the initial SQL output using various strategies.



NL2SQL360: LEADERBOARD



1	SFT CodeS-15B	85.5%	●
2	RESDSQL-3B + NatSQL	83.9%	◆
3	RESDSQL-3B	82.3%	◆
4	Graphix + PICARD	80.6%	◆
4	DAIL-SQL	80.6%	○
5	DAIL-SQL (SC)	79.0%	○
6	DIN-SQL	75.8%	○
7	C3SQL	69.4%	○

(a) Test Results in Domain of *Competition*

1	RESDSQL-3B + NatSQL	76.7%	◆
2	SFT CodeS-15B	76.2%	●
3	DIN-SQL	75.7%	○
4	DAIL-SQL (SC)	75.0%	○
5	C3SQL	73.3%	○
6	DAIL-SQL	72.8%	○
7	Graphix + PICARD	69.6%	◆
8	RESDSQL-3B	69.1%	◆

(b) Test Results in SQL with JOIN

1	DIN-SQL	63.6%	○
1	DAIL-SQL (SC)	63.6%	○
2	DAIL-SQL	57.6%	○
3	SFT CodeS-15B	51.5%	●
4	Graphix + PICARD	45.5%	◆
5	C3SQL	42.4%	○
6	RESDSQL-3B	39.4%	◆
7	RESDSQL-3B + NatSQL	33.3%	◆

(c) Test Results in Nested SQL

1	RESDSQL-3B + NatSQL	96.6%	◆
2	Graphix + PICARD	96.0%	◆
3	SFT CodeS-15B	95.5%	●
4	RESDSQL-3B	95.1%	◆
5	DIN-SQL	94.1%	○
6	DAIL-SQL (SC)	93.6%	○
7	DAIL-SQL	93.3%	○
8	C3SQL	93.0%	○

(d) Test Results in Query Variance

Figure 3: NL2SQL Models on Spider from Different Angles (○: Prompting LLM, ●: Fine-tuning LLM, ◆: Fine-tuning PLM).

Li B, Luo Y, Chai C, et al. The Dawn of Natural Language to SQL: Are We Fully Ready?. VLDB 2024

REFERENCES

- Liu, X., Shen, S., Li, B., Ma, P., Jiang, R., Zhang, Y., Fan, J., Li, G., Tang, N., & Luo, Y. (2024). *A Survey of NL2SQL with Large Language Models: Where are we, and where are we going?*
- Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, Dongmei Zhang:Towards Complex Text-to-SQL in Cross-Domain Database with Intermediate Representation. ACL (1) 2019: 4524-4535
- Chao Zhang, Yuren Mao, Yijiang Fan, Yu Mi, Yunjun Gao, Lu Chen, Dongfang Lou, Jinshu Lin:FinSQL: Model-Agnostic LLMs-based Text-to-SQL Framework for Financial Analysis. SIGMOD Conference Companion 2024: 93-105
- Shayan Talaei, Mohammadreza Pourreza, Yu-Chen Chang, Azalia Mirhoseini, Amin Saberi:CHESS: Contextual Harnessing for Efficient SQL Synthesis. CoRR abs/2405.16755 (2024) Shayan Talaei, Mohammadreza
- Bing Wang, Changyu Ren, Jian Yang, Xinnian Liang, Jiaqi Bai, Qian-Wen Zhang, Zhao Yan, Zhoujun Li:MAC-SQL: A Multi-Agent Collaborative Framework for Text-to-SQL. CoRR abs/2312.11242 (2023)
- Pengcheng Yin, Graham Neubig, Wen-tau Yih, Sebastian Riedel:TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data. ACL 2020: 8413-8426
- Haoyang Li, Jing Zhang, Hanbing Liu, Ju Fan, Xiaokang Zhang, Jun Zhu, Renjie Wei, Hongyan Pan, Cuiping Li, Hong Chen:CodeS: Towards Building Open-source Language Models for Text-to-SQL. CoRR abs/2402.16347 (2024)
- Chang Gao, Bowen Li, Wenxuan Zhang, Wai Lam, Binhua Li, Fei Huang, Luo Si, Yongbin Li:Towards Generalizable and Robust Text-to-SQL Parsing. EMNLP (Findings) 2022: 2113-2125
- Torsten Scholak, Nathan Schucher, Dzmitry Bahdanau:PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models. EMNLP (1) 2021: 9895-9901

THANK YOU