BIRD: 针对大规模数据库的大型NL2SQL基准测试

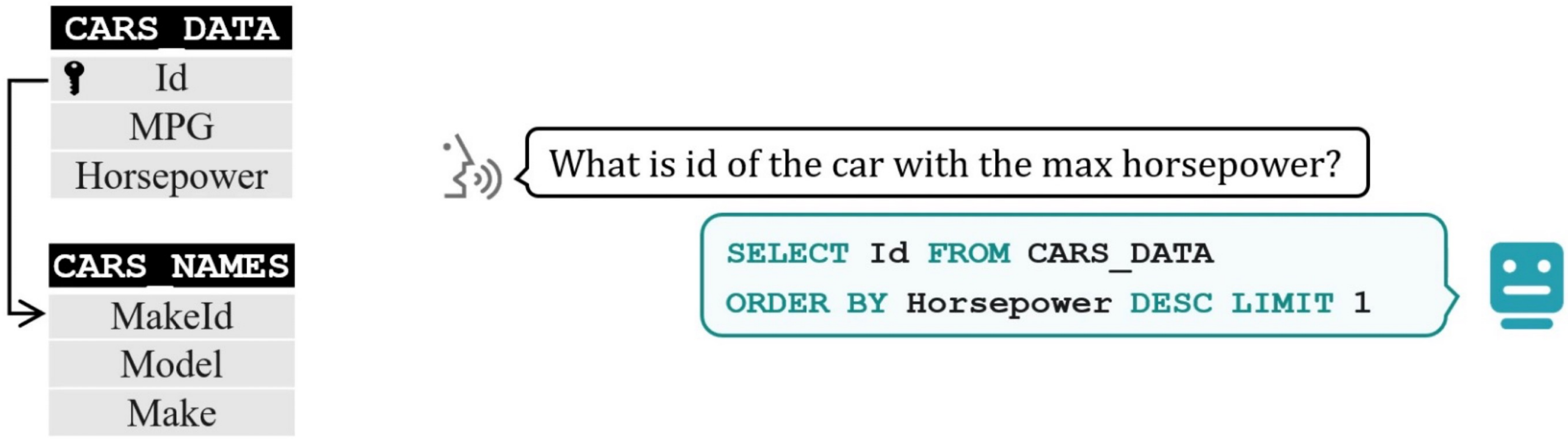
马晨昊
数据科学学院
香港中文大学（深圳）

Content

- Graphix-T5 with history context
- BIRD: Real-world Text-to-SQL Bench
- Discussions

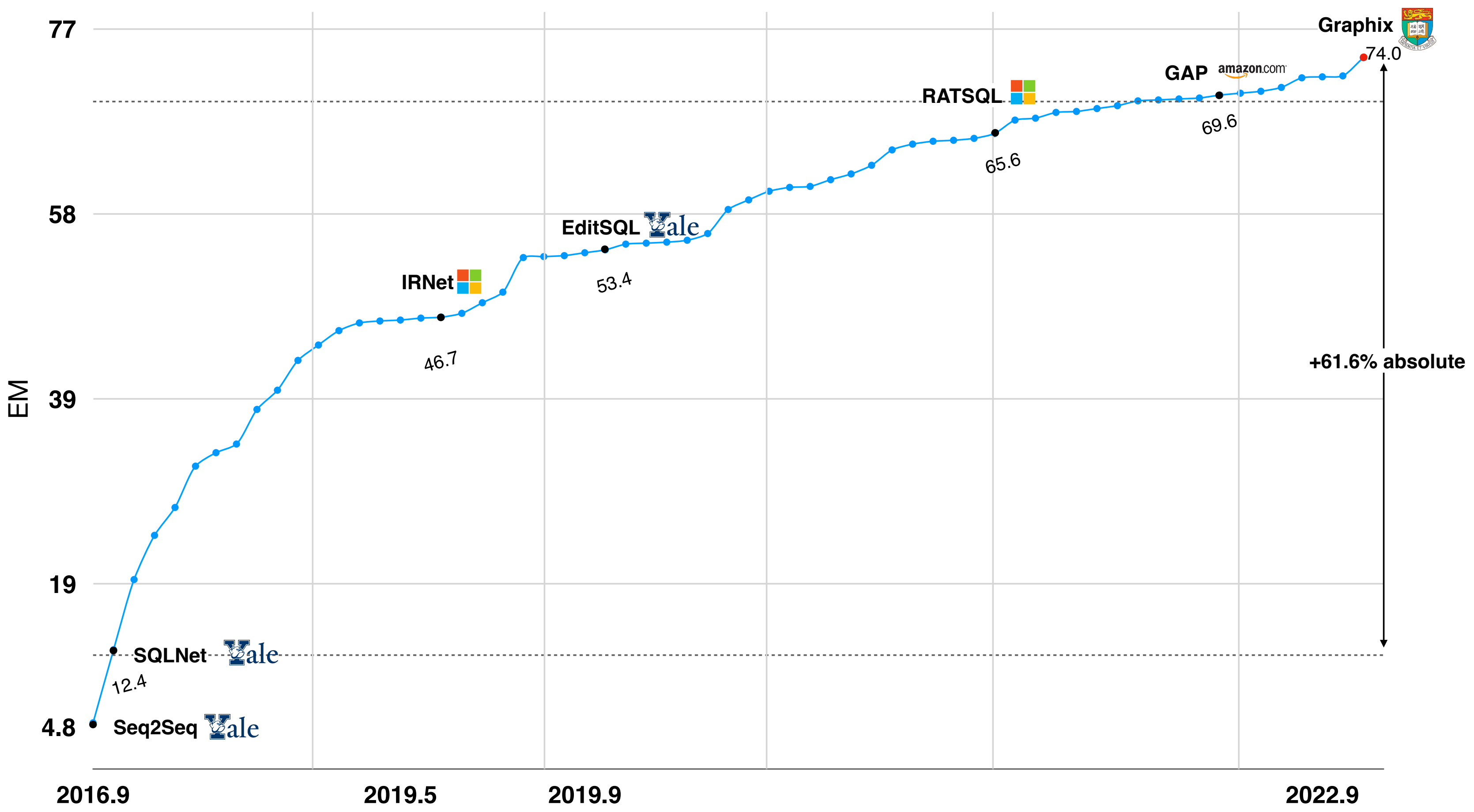
Text-to-SQL Parsing

- Text-to-SQL, which aims at converting **natural language questions** into **executable SQL queries**, has garnered increasing attention, as it can assist end users in efficiently extracting vital information from databases without need for the technical background.



Unlocking Tech Growth by Valuable Benchmark

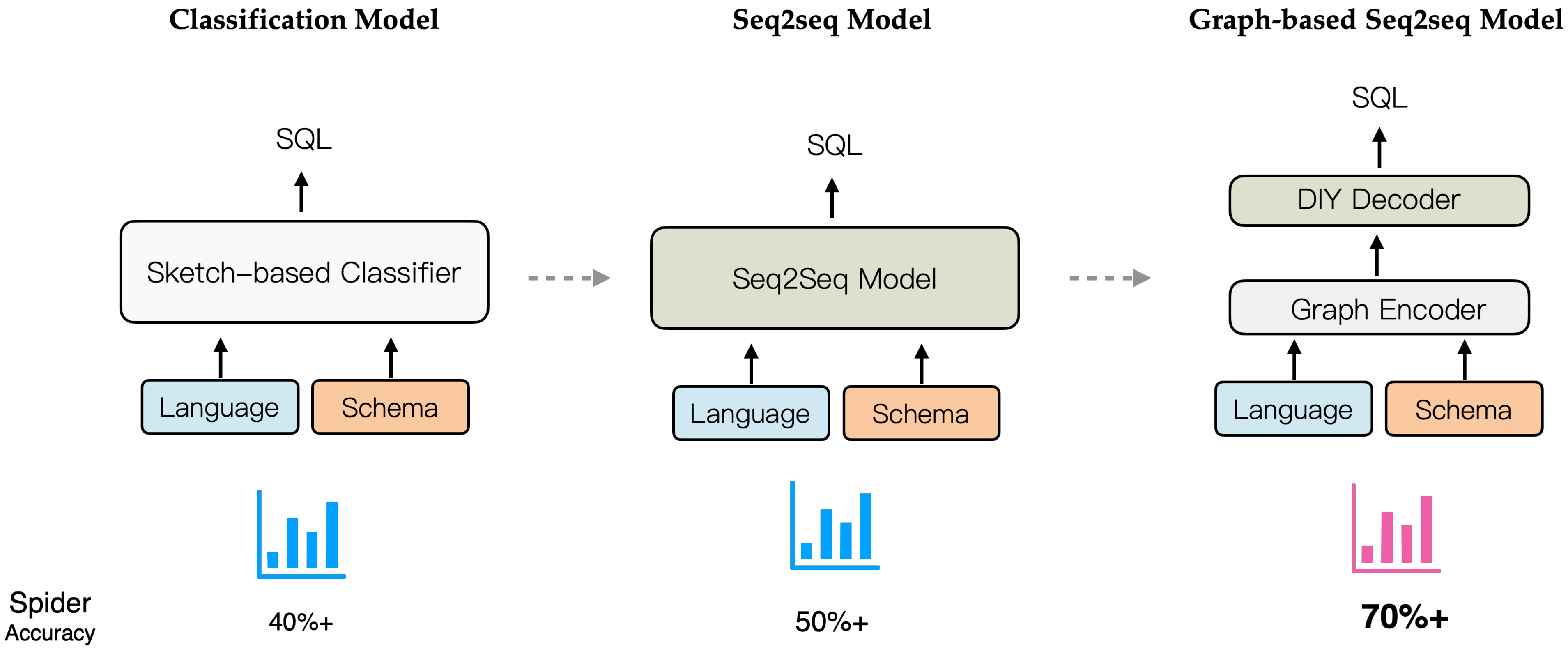
- Leveraging a valuable benchmark can significantly enhance technical growth in the realm of Text-to-SQL.



In the past 5 years, more than 60 submissions for **Spider** have been made, driving the development of text-to-SQL approaches.

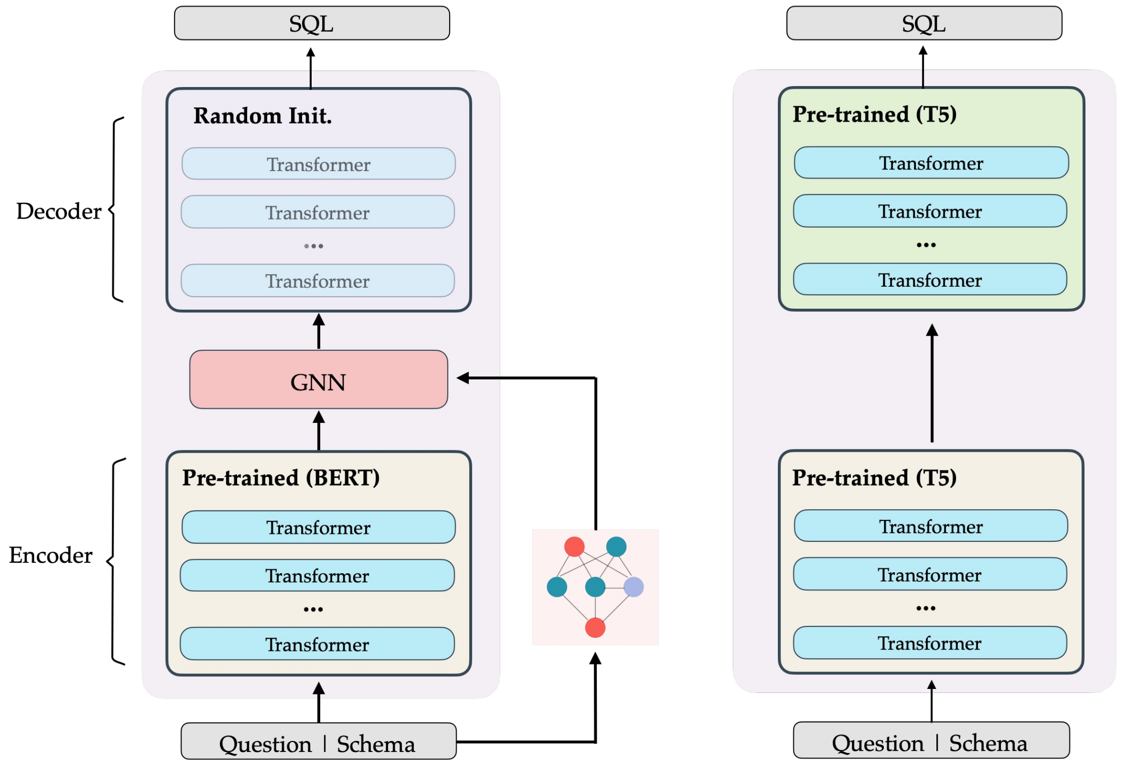
Text-to-SQL Model Evolution:

- Graph-based encoder with PLM shows the most effectiveness on Spider, which is a large-scale cross-domain text-to-SQL benchmark, in recent years.



Text-to-SQL Model Evolution:

- The Text-to-Text PLMs (i.e., T5, BART) recently demonstrate their portability and potency on text-to-SQL missions by allowing for simple fine-tuning.

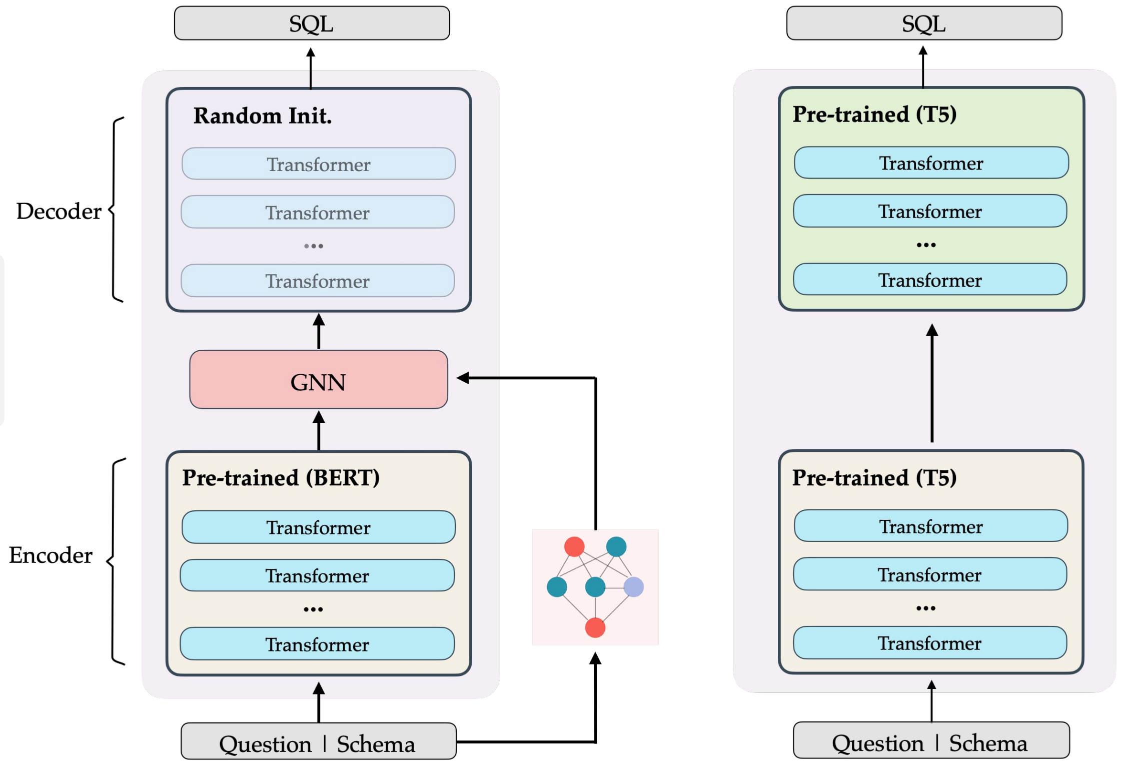


(a) RATSQL / LGESQL (b) T5

Text-to-SQL Model Evolution:

• The Text-to-Text PLMs (i.e., T5, BART) recently demonstrate their portability and potency on text-to-SQL missions by allowing for simple fine-tuning.

You have to summarize grammar rules for each dataset manually



(a) RATSQL / LGESQL

(b) T5

Portable for different dataset !!

Challenges of T5 (Text-to-Text PLM):

• One of T5's challenges for text-to-SQL tasks is the **hallucinations**, which results in incorrect SQLs, especially when dealing with challenging cases.
Hallucinations exist even

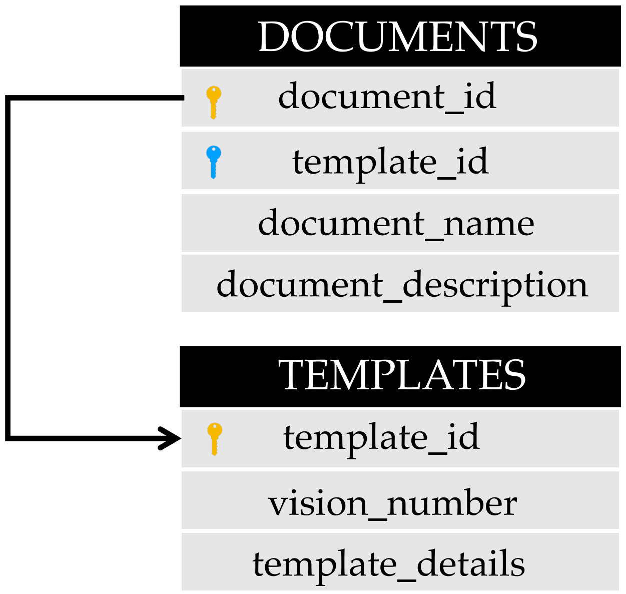
List paper IDs, paper names, and paper descriptions for all papers.

T5-3B:

```
SELECT paper_id, paper_name,
paper_description FROM documents;
```

Gold:

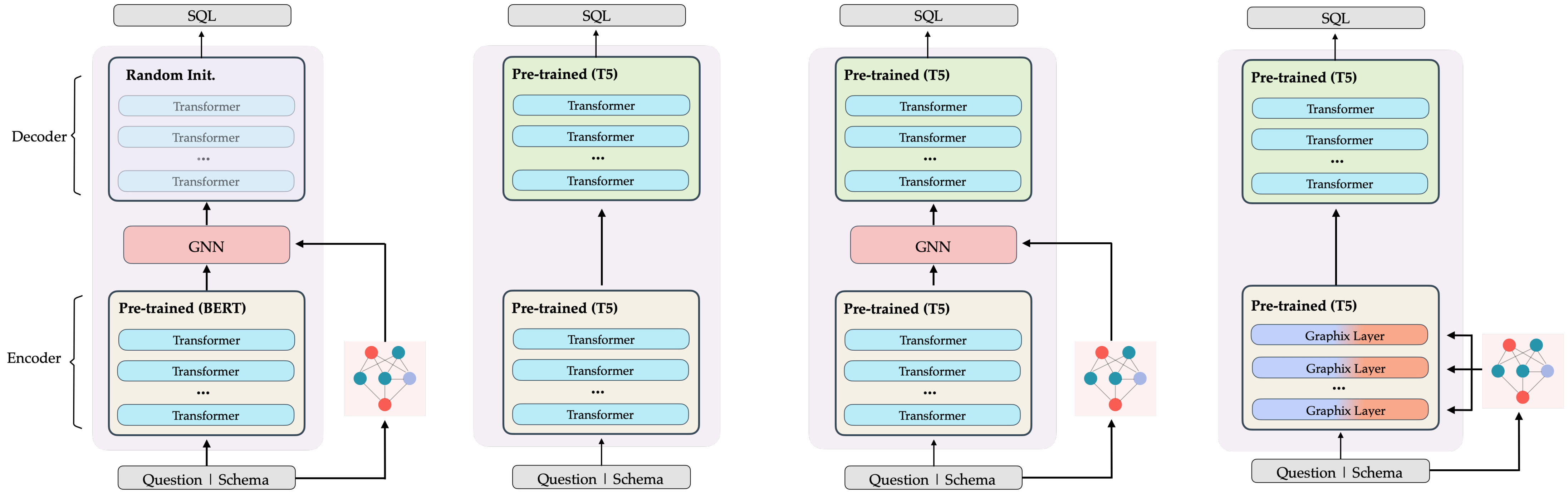
```
SELECT document_id, document_name,
document_description FROM documents;
```



Method: Graphix-T5 (AAAI 2023 Oral)

Previous work & our method:

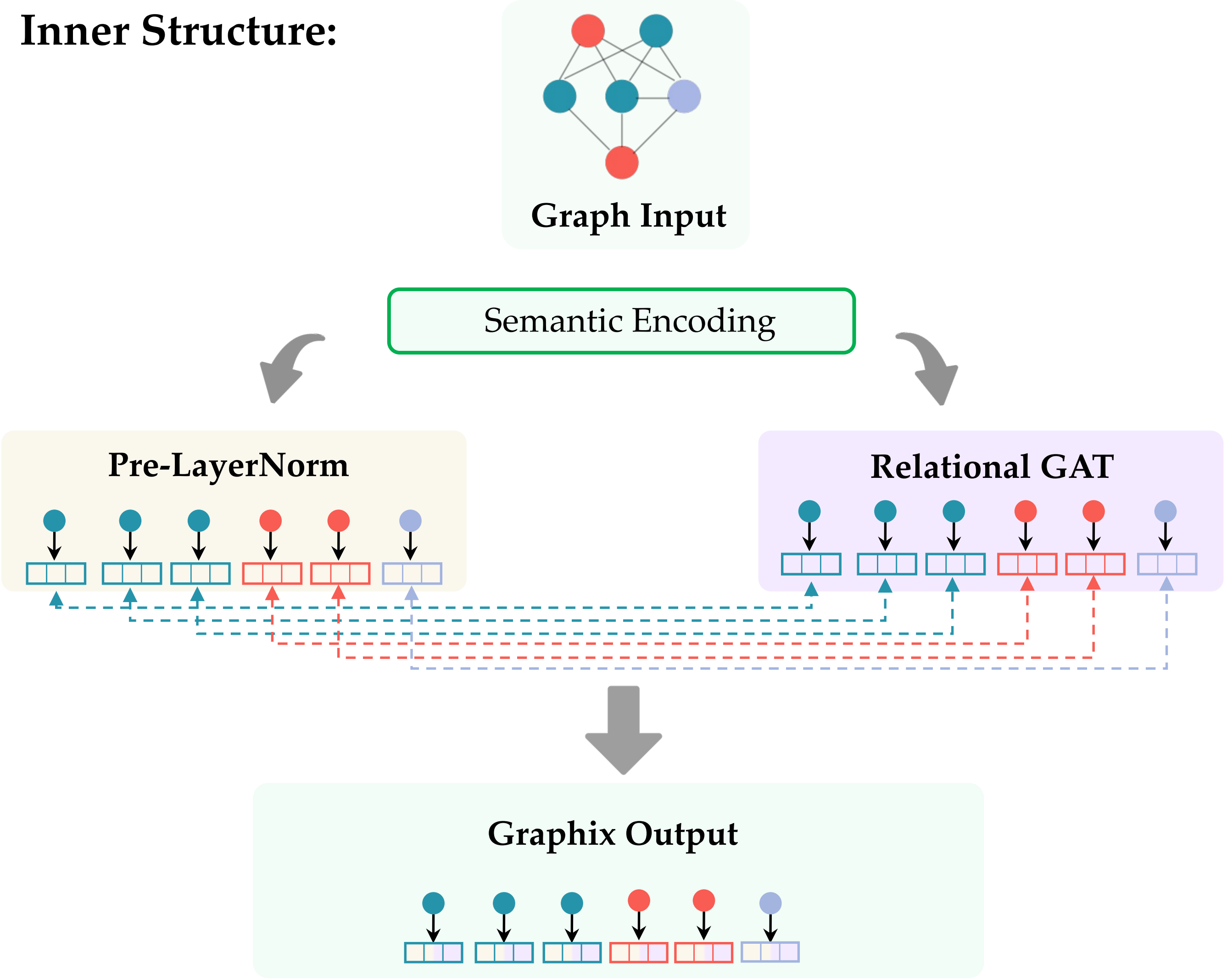
(a) RATSQL [pre-trained BERT-encoder → graph-based module → randomly initialized decoder].
(b) T5 [pre-trained T5-encoder → pre-trained T5-decoder] and the proposed variant
(c) GNN- T5 [pre-trained T5-encoder → graph-based module → pre-trained T5-decoder]
(d) GRAPHIX-T5 [semi-pre-trained graphix-module → pre-trained T5-decoder] via multi-hop reasoning.



(a) RATSQL (b) T5 (c) GNN-T5 (d) Graphix-T5

Method: Graphix-T5

Inner Structure:



Semantic Representations:

$$\widetilde{\mathcal{H}}_{\mathcal{S}}^{(l)} = \mathbf{LayerNorm}(\widehat{\mathcal{H}}_{\mathcal{S}}^{(l)} + \mathbf{FFN}(\widehat{\mathcal{H}}_{\mathcal{S}}^{(l)})),$$

Structural Representations: (Relational GAT)

$$\begin{aligned}
\vec{\alpha}_{ij} &= \frac{e_i^{init}\widetilde{\mathbf{W}}_Q \left(e_j^{init}\widetilde{\mathbf{W}}_K + \phi\left(r_{ij}\right)\right)^\top}{\sqrt{d_z}}, \\
\alpha_{ij} &= \mathbf{softmax}_j\left(\vec{\alpha}_{ij}\right), \\
\hat{e}_i^{init} &= \sum_{j\in\widetilde{\mathcal{N}}_i} \alpha_{ij}\left(e_j^{init}\widetilde{\mathbf{W}}_V + \phi(r_{ij})\right), \\
\hat{e}_i^{(l)} &= \mathbf{LayerNorm}(e_i^{init} + \hat{e}_i^{init}\widetilde{\mathbf{W}}_O), \\
\tilde{e}_i^{(l)} &= \mathbf{LayerNorm}(\hat{e}_i^{(l)} + \mathbf{FFN}(\hat{e}_i^{(l)})),
\end{aligned}$$

Joint Representations:

$$\widetilde{\mathcal{H}}_{\mathcal{M}}^{(l)} = \widetilde{\mathcal{H}}_{\mathcal{S}}^{(l)} + \widetilde{\mathcal{E}}_{\mathcal{G}}^{(l)},$$

Method: Graphix-T5

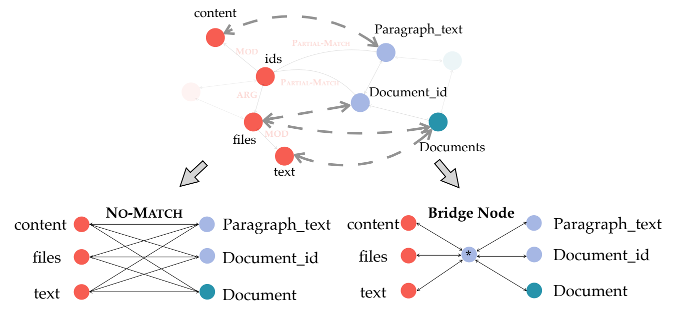
Pre-defined Relations:

Source x	Target y	Relation Type	Description
Question	Question	MODIFIER	y is a modifier of x.
Question	Question	ARGUMENT	y is the source token of x under the syntax dependency outside of modifier.
Question	Question	DISTANCE-1	y is the nearest (1-hop) neighbor of x.
Column	Column	FOREIGN-KEY	y is the foreign key of x.
Column	Column	SAME-TABLE	x and y appears in the same table.
Column	*	BRIDGE	x and y are linked when y is the special column token '*'.
Table	Column	HAS	The column y belongs to the table x.
Table	Column	PRIMARY-KEY	The column y is the primary key of the table x.
Table	*	BRIDGE	x and y are connected when y is the special column token '*'.
Question	Table	EXACT-MATCH	x is part of y, and y is a span of the entire question.
Question	Table	PARTIAL-MATCH	x is part of y, but the entire question does not contain y.
Question	Column	EXACT-MATCH	x is part of y, and y is a span of the entire question.
Question	Column	PARTIAL-MATCH	x is part of y, but the entire question does not contain y.
Question	Column	VALUE-MATCH	x is part of the candidate cell values of column y.
Question	*	BRIDGE	x and y are linked when y is the special column token '*'.

Table 6: The checklist of main types of relations used in GRAPHIX-T5. All relations above are asymmetric.

Bridge Node Mode:

N x M → N + M (neighbors)



(a) No-Match Mode (b) Bridge Node Mode

Figure 3: Figure shows the circumstances when entities in the question are hard to string-match the schema items. (a) is the strategy to solve this case by NO-MATCH Mode, which fully connects schema nodes with all token nodes. (b) is our solution to add a bridge node to link the question and schema nodes via less number of edges.

Experiments:

- **Performance on 4 datasets and compositional generalization:**

MODEL	EM	EX
RAT-SQL + BERT ♡	69.7	-
RAT-SQL + Grappa ♡	73.9	-
GAZP + BERT	59.1	59.2
BRIDGE v2 + BERT	70.0	68.3
NatSQL+GAP	73.7	75.0
SMBOP + GRAPPA	74.7	75.0
LGESQL + ELECTRA ♡	75.1	-
S^2SQL + ELECTRA ♡	76.4	-
T5-large	67.0	69.3
GRAPHIX-T5-large	72.7 (↑ 5.7)	75.9 (↑ 6.6)
T5-large + PICARD ♣	69.1	72.9
GRAPHIX-T5-large + PICARD ♣	76.6 (↑ 7.5)	80.5 (↑ 7.6)
T5-3B	71.5	74.4
GRAPHIX-T5-3B	75.6 (↑ 4.1)	78.2 (↑ 3.8)
T5-3B + PICARD ♣	75.5	79.3
GRAPHIX-T5-3B + PICARD ♣	**77.1** (↑ 1.6)	**81.0** (↑ 1.7)

Table 1: Exact match (EM) and execution (EX) accuracy (%) on SPIDER development set.

MODEL	TEMPLATE	LENGTH	TMCD
T5-base	59.3	49.0	60.9
T5-3B	64.8	56.7	69.6
NQG-T5-3B	64.7	56.7	69.5
GRAPHIX-T5-3B	**70.1** (↑ 5.4)	**60.6** (↑ 3.9)	**73.8** (↑ 4.3)

Table 3: Exact match (EM) accuracy (%) on compositional dataset SPIDER-SSP.

MODEL	SYN	DK	REALISTIC
GNN	23.6	26.0	-
IRNet	28.4	33.1	-
RAT-SQL	33.6	35.8	-
RAT-SQL + BERT	48.2	40.9	58.1
RAT-SQL + Grappa	49.1	38.5	59.3
LGESQL + ELECTRA	64.6	48.4	69.2
T5-large	53.6	40.0	58.5
GRAPHIX-T5-large	61.1 (↑ 7.5)	48.6 (↑ 8.6)	67.3 (↑ 8.8)
T5-3B	58.0	46.9	62.0
GRAPHIX-T5-3B	**66.9** (↑ 8.9)	**51.2** (↑ 4.3)	**72.4** (↑ 10.4)

Table 2: Exact match (EM) accuracy (%) on SYN, DK and REALISTIC benchmark.

Experiments:

- **Performance on 4 datasets and compositional generalization:**

Model	EM	EX
RAT-SQL + BERT ♡	69.7	-
RAT-SQL + Grappa ♡	73.9	-
GAZP + BERT	59.1	59.2
BRIDGE v2 + BERT	70.0	68.3
NatSQL+GAP	73.7	75.0
SMBOP + GRAPPA	74.7	75.0
LGESQL + ELECTRA ♡	75.1	-
S^2SQL + ELECTRA ♡	76.4	-
T5-large	67.0	69.3
Graphix-T5-large	72.7 (↑ 5.7)	75.9 (↑ 6.6)
T5-large + Picard ♣	69.1	72.9
Graphix-T5-large + Picard ♣	76.6 (↑ 7.5)	80.5 (↑ 7.6)
T5-3B	71.5	74.4
Graphix-T5-3B	75.6 (↑ 4.1)	78.2 (↑ 3.8)
T5-3B + Picard ♣	75.5	79.3
Graphix-T5-3B + Picard ♣	**77.1** (↑ 1.6)	**81.0** (↑ 1.7)

Table 1: Exact match (EM) and execution (EX) accuracy (%) on SPIDER development set.

Model	Template	Length	TMCD
T5-base	59.3	49.0	60.9
T5-3B	64.8	56.7	69.6
NQG-T5-3B	64.7	56.7	69.5
Graphix-T5-3B	**70.1** (↑ 5.4)	**60.6** (↑ 3.9)	**73.8** (↑ 4.3)

Table 3: Exact match (EM) accuracy (%) on compositional dataset SPIDER-SSP.

Observation:

- Graphix improves T5 a lot
- Graphix-T5-large > T5-3B

Model	Syn	DK	Realistic
GNN	23.6	26.0	-
IRNet	28.4	33.1	-
RAT-SQL	33.6	35.8	-
RAT-SQL + BERT	48.2	40.9	58.1
RAT-SQL + Grappa	49.1	38.5	59.3
LGESQL + ELECTRA	64.6	48.4	69.2
T5-large	53.6	40.0	58.5
Graphix-T5-large	61.1 (↑ 7.5)	48.6 (↑ 8.6)	67.3 (↑ 8.8)
T5-3B	58.0	46.9	62.0
Graphix-T5-3B	**66.9** (↑ 8.9)	**51.2** (↑ 4.3)	**72.4** (↑ 10.4)

Table 2: Exact match (EM) accuracy (%) on SYN, DK and REALISTIC benchmark.

Experiments:

- **Performance on 4 datasets and compositional generalization:**

MODEL	EM	EX
RAT-SQL + BERT ♡	69.7	-
RAT-SQL + Grappa ♡	73.9	-
GAZP + BERT	59.1	59.2
BRIDGE v2 + BERT	70.0	68.3
NatSQL+GAP	73.7	75.0
SMBOP + GRAPPA	74.7	75.0
LGESQL + ELECTRA ♡	75.1	-
S^2SQL + ELECTRA ♡	76.4	-
T5-large	67.0	69.3
GRAPHIX-T5-large	72.7 (↑ 5.7)	75.9 (↑ 6.6)
T5-large + PICARD ♣	69.1	72.9
GRAPHIX-T5-large + PICARD ♣	76.6 (↑ 7.5)	80.5 (↑ 7.6)
T5-3B	71.5	74.4
GRAPHIX-T5-3B	75.6 (↑ 4.1)	78.2 (↑ 3.8)
T5-3B + PICARD ♣	75.5	79.3
GRAPHIX-T5-3B + PICARD ♣	**77.1** (↑ 1.6)	**81.0** (↑ 1.7)

Table 1: Exact match (EM) and execution (EX) accuracy (%) on SPIDER development set.

MODEL	TEMPLATE	LENGTH	TMCD
T5-base	59.3	49.0	60.9
T5-3B	64.8	56.7	69.6
NQG-T5-3B	64.7	56.7	69.5
GRAPHIX-T5-3B	**70.1** (↑ 5.4)	**60.6** (↑ 3.9)	**73.8** (↑ 4.3)

Table 3: Exact match (EM) accuracy (%) on compositional dataset SPIDER-SSP.

MO			REALISTIC
GN			-
IR			-
RA			-
RA			58.1
RA			59.3
LG			69.2
T5-large	53.6	40.0	58.5
GRAPHIX-T5-large	61.1 (↑ 7.5)	48.6 (↑ 8.6)	67.3 (↑ 8.8)
T5-3B	58.0	46.9	62.0
GRAPHIX-T5-3B	**66.9** (↑ 8.9)	**51.2** (↑ 4.3)	**72.4** (↑ 10.4)

Table 2: Exact match (EM) accuracy (%) on SYN, DK and REALISTIC benchmark.

Observation:

- Graphix improves T5 a lot
- Graphix-T5-large (1B) > T5-3B

Experiments:

- **Performance on Low-Resource Setting:**

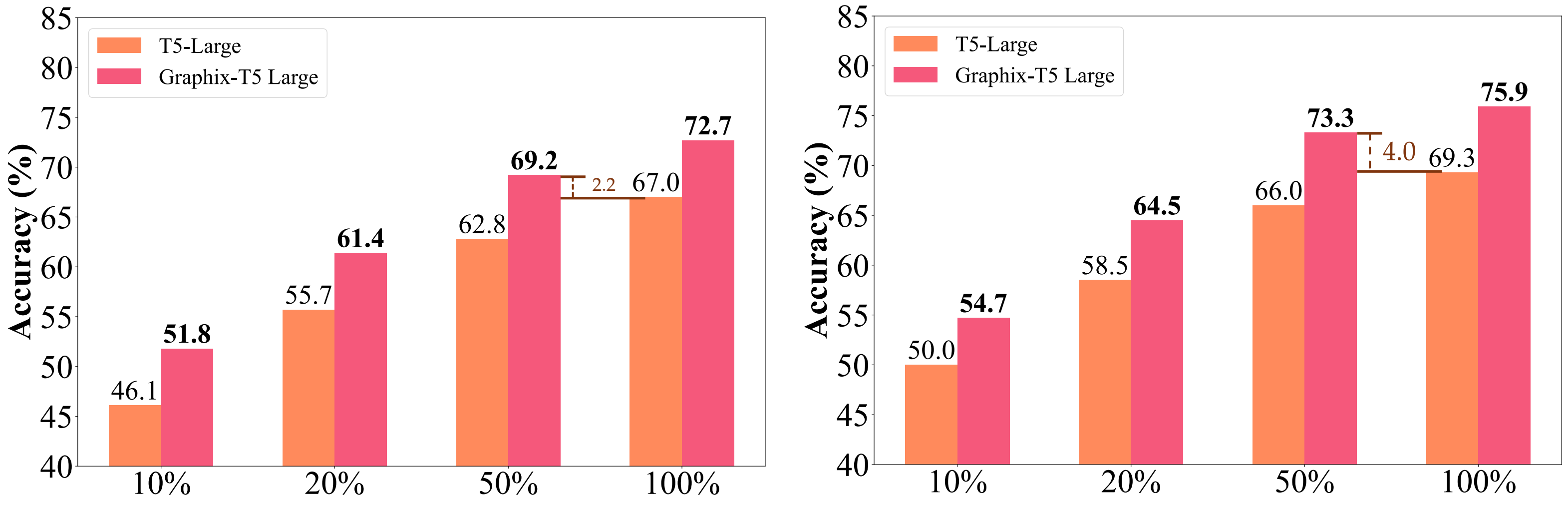


Figure 4: Exact match (EM) (left) and execution (EX) (right) accuracy (%) on SPIDER low-resource setting.

Observation:

- Graphix-T5-large w 50% data

> T5-large w 100% data

Experiments:

- **Performance on Low-Resource Setting:**

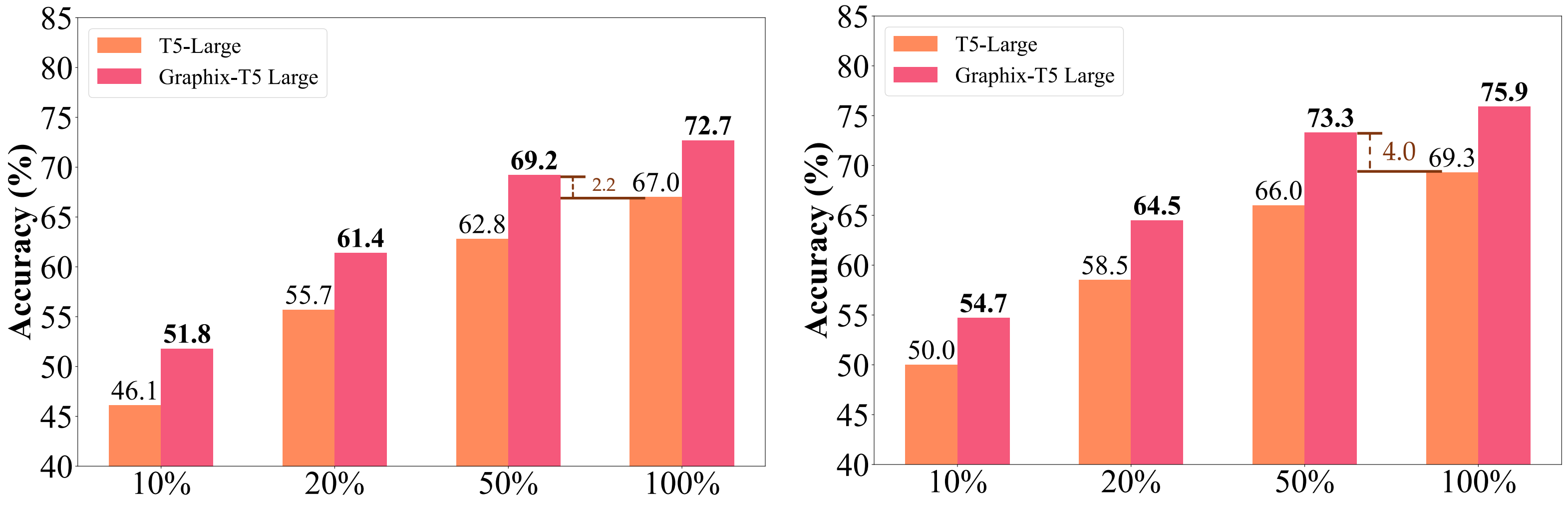


Figure 4: Exact match (EM) (left) and execution (EX) (right) accuracy (%) on SPIDER low-resource setting.

Take Away:

- structural knowledge created by humans can compensate for the inadequate learning due to low-resource data

Experiments:

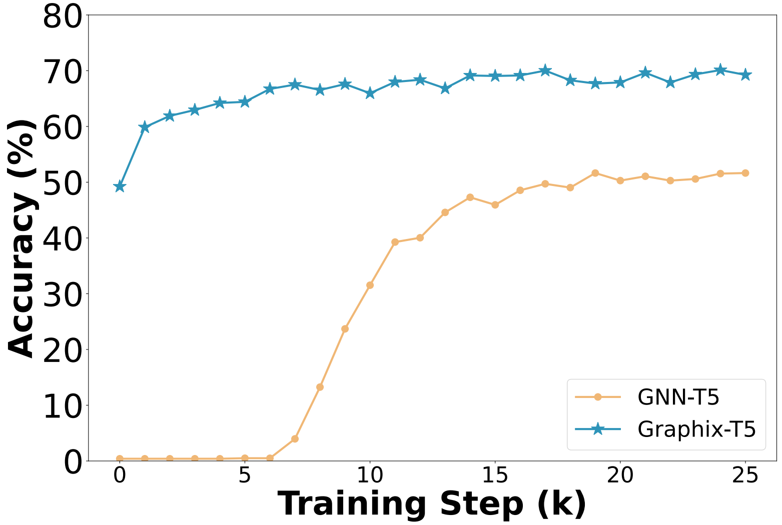
- **Ablation Study:**

Question:

- How effective is Bridge Mode?
- Could Graphix be incorporated into decoder?
- Is Graphix superior than other GNN variants ?

MODEL	EM	EX
(a) RAT-SQL + BERT	69.7	-
(b) T5-large	67.0	69.3
(c) GNN-T5-large	51.6	54.5
(d) GRAPHIX-T5-large		
w/ BRIDGE Mode	**72.7**	**75.9**
w/ NO-MATCH Mode	71.1	74.2
w/ DOUBLE-GRAPH	72.0	74.7

Table 5: Ablation Study of Graphix-T5



Experiments:

- **Ablation Study:**

Question:

- How effective is Bridge Mode?

  Bridge > No-Match

- Could Graphix be incorporated into decoder?

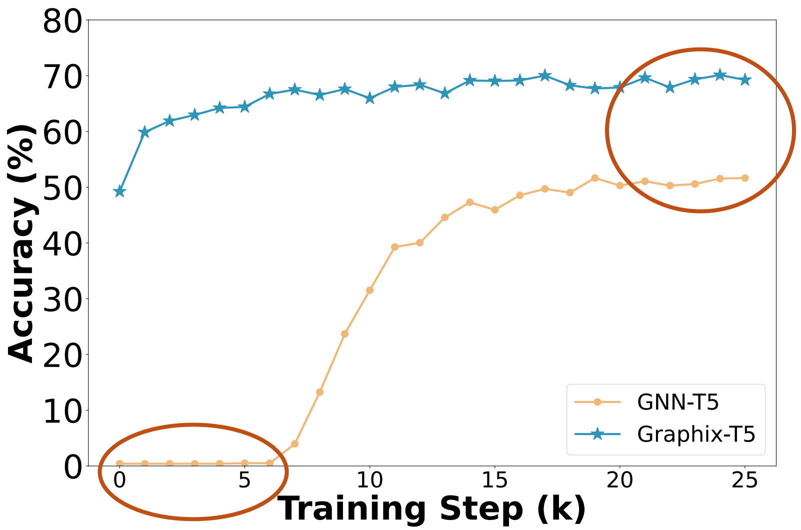
  No, it will break the generation capability

- Is Graphix superior to other GNN variants ?

  Yes, Graphix can inject structural bias w / o catastrophic forgetting

MODEL	EM	EX
(a) RAT-SQL + BERT	69.7	-
(b) T5-large	67.0	69.3
(c) GNN-T5-large	51.6	54.5
(d) GRAPHIX-T5-large		
w/ BRIDGE Mode	**72.7**	**75.9**
w/ NO-MATCH Mode	71.1	74.2
w/ DOUBLE-GRAPH	72.0	74.7

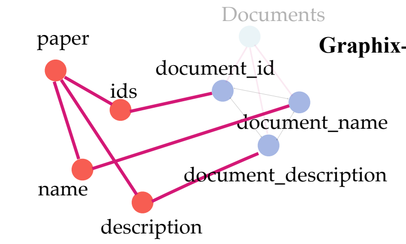
Table 5: Ablation Study of Graphix-T5



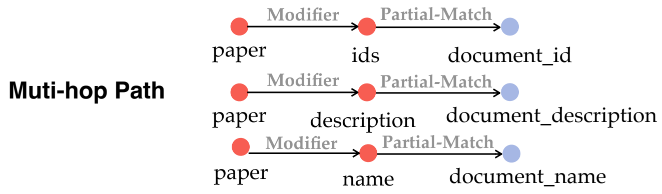
Catastrophic forgetting

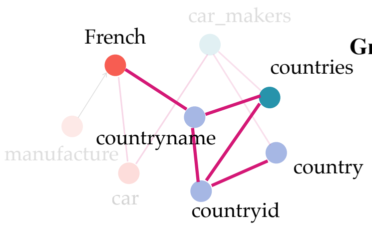
Experiments:

- **Qualitive & Difficulty Analysis:**

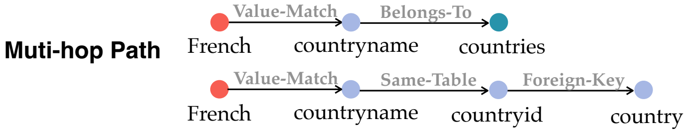


Question: *List paper IDs, paper names, and paper descriptions for all papers.*
T5-3B: SELECT paper_id, paper_name, paper_description FROM documents
Graphix-T5-3B: SELECT document_id, document_name, document_description FROM documents
Gold: SELECT **document_id**, **document_name**, **document_description** FROM documents





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';



Model	Spider					Syn					DK					Realistic				
	easy	medium	hard	extra	all	easy	medium	hard	extra	all	easy	medium	hard	extra	all	easy	medium	hard	extra	all
T5-large	85.5	70.9	55.2	41.6	67.0	69.0	56.8	46.3	30.2	53.6	**64.1**	44.3	22.9	18.1	40.0	79.8	68.0	44.4	28.9	58.5
Graphix-T5-large	**89.9**	**78.7**	**59.8**	**44.0**	**72.6**	**75.8**	**67.5**	**50.6**	**33.1**	**61.1**	63.6	**54.5**	**33.8**	**29.5**	**48.6**	**88.1**	**77.3**	**50.5**	**40.2**	**67.3**
T5-3B	89.5	78.3	58.6	40.4	71.6	74.2	64.5	48.0	27.8	58.0	**69.9**	53.5	24.3	24.8	46.9	85.3	73.4	46.5	27.8	62.0
Graphix-T5-3B	**91.9**	**81.6**	**61.5**	**50.0**	**75.6**	**80.6**	**73.1**	**52.9**	**44.6**	**66.9**	69.1	**55.3**	**39.2**	**31.4**	**51.2**	**93.6**	**85.7**	**52.5**	**41.2**	**72.4**

Observation:

- Graphix can make T5 aware of structure of databases to generate more structure-rich SQLs in terms of both semantics & structures.
- Graphix-T5 can deal with more **complicated** text-to-SQL scenarios than vanilla T5.
- **Structural Grounding** is beneficial to text-to-text PLM especially in the harder but real text-to-SQLs.

Summary of Graphix-T5:

- We proposed an effective architecture to boost the capability of **structural encoding** of T5 cohesively while keeping the pre-trained T5's potent contextual encoding ability.

- In order to achieve this goal, we designed a **Graph-Aware semi-pretrained** text-to-text PLM, namely **Graphix-T5** to augment the multi-hop reasoning for the challenging text-to-SQL tasks

- The results under the extensive experiments demonstrate the effectiveness of Graphix-T5, proving that **structural bias** is crucial for the current text-to-text PLMs for especially complicated text-to-SQL cases.

What's next?:

Spider **1.0**

Yale Semantic Parsing and Text-to-SQL Challenge

1 Aug 20, 2023	DAIL-SQL + GPT-4 + Self-Consistency *Alibaba Group* (Gao and Wang et al.,'2023) code	**86.6**
2 Aug 9, 2023	DAIL-SQL + GPT-4 *Alibaba Group* (Gao and Wang et al.,'2023) code	**86.2**
3 October 17, 2023	DPG-SQL + GPT-4 + Self-Correction *Anonymous* Code and paper coming soon	85.6
4 Apr 21, 2023	DIN-SQL + GPT-4 *University of Alberta* (Pourreza et al.,'2023) code	85.3
5 July 5, 2023	Hindsight Chain of Thought with GPT-4 *Anonymous* Code and paper coming soon	83.9
6 Jun 1, 2023	C3 + ChatGPT + Zero-Shot *Zhejiang University & Hundsun* (Dong et al.,'2023) code	**82.3**
7 July 5, 2023	Hindsight Chain of Thought with GPT-4 and Instructions *Anonymous* Code and paper coming soon	80.8

Recent SOTA models on previous benchmark are **dominated** by GPT-4

🤔

So, can LLM already serve as a database interface?

What's next?:

- The previous benchmarks have mostly focused on **database schema**, ignoring the importance of big / dirty database values (or records).

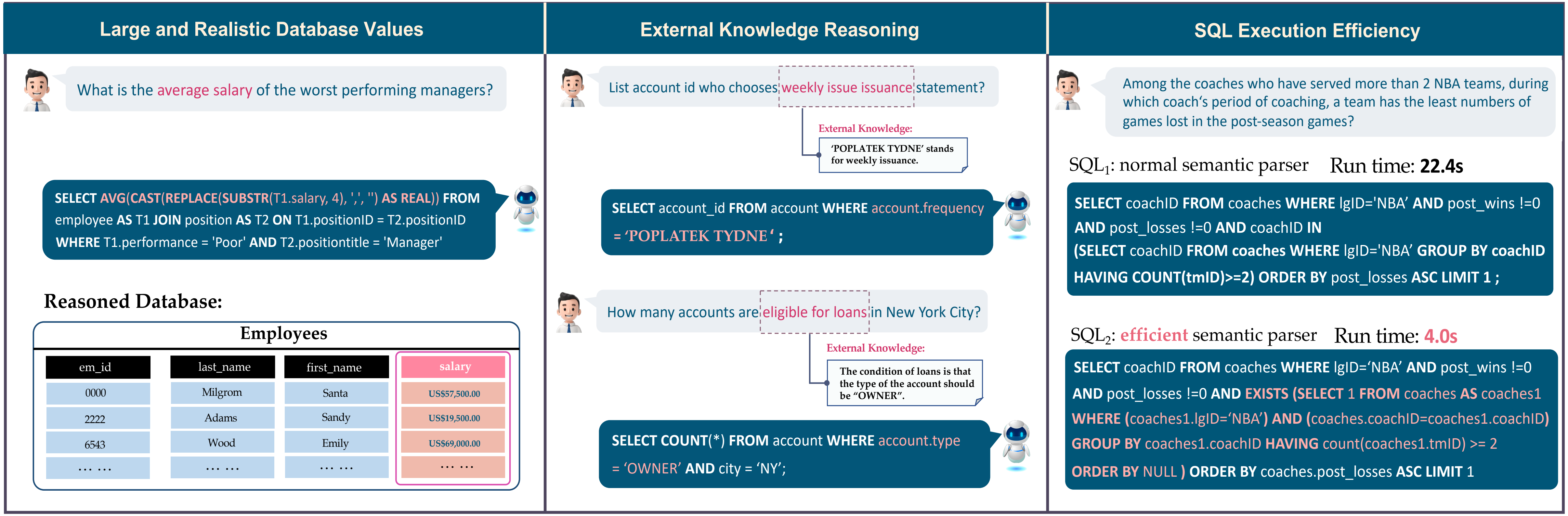
	Cinema_ID	Film_ID	Date	Show_times_per_day	Price
1	1	1	21 May	5	12.99
2	1	2	21 May	3	12.99
3	1	3	21 Jun	2	8.99
4	2	1	11 July	5	9.99
5	6	5	2 Aug	4	12.99
6	9	4	20 May	5	9.99
7	10	1	19 May	5	15.99

	Dname	Dnumber	Mgr_ssn	Mgr_start_date
1	Headquarters	1	888665555	1981-06-19
2	Administration	4	987654321	1995-01-01
3	Research	5	333445555	1988-05-22

As most database contents in the Spider are minimal and tidy, this produces a discrepancy between idealized and real-world scenarios.

Can LLM Already Serve as A Database Interface?

BIRD: A BIg Bench for Large-Scale Database Grounded Text-to-SQLs



https://bird-bench.github.io/

Can LLM Already Serve as A Database Interface? NeurIPS 2023 Spotlight

BIRD: A BIg Bench for Large-Scale Database Grounded Text-to-SQLs

Dev set reached 50K+ downloads

Mainly supported for Industries (20 +):

About BIRD

BIRD (**BI**g Bench for La**R**ge-scale **D**atabase Grounded Text-to-SQL Evaluation) represents a pioneering, cross-domain dataset that examines the impact of extensive database contents on text-to-SQL parsing. BIRD contains over **12,751** unique question-SQL pairs, **95** big databases with a total size of **33.4 GB**. It also covers more than **37** professional domains, such as blockchain, hockey, healthcare and education, etc.

Paper

Code

Train Set | 🔥 Dev Set

Leaderboard - Execution Accuracy (EX)

	Model	Code	Size	Orale Knowledge	Dev (%)	Test (%)
	Human Performance *Data Engineers + DB Students*			✓		**92.96**
🏆1 Aug 15, 2023	DIN-SQL + GPT-4 *University of Alberta* [Pourreza et al. 2023]	[link]	UNK	✓	50.72	**55.90**
2 Jul 01, 2023	GPT-4 *Baseline*	[link]	UNK	✓	46.35	54.89
3 Jul 16, 2023	Claude-2 *Baseline*	[link]	UNK	✓	42.70	49.02

https://bird-bench.github.io/

Can LLM Already Serve as A Database Interface?

BIRD: A BIg Bench for Large-Scale Database Grounded Text-to-SQLs

Mainly supported for Universities (10 +):

Stanford CS 224V SLIDES & HW

Summary

- Few-shot Chat-GPT parses SQL queries for Yelp
  - Restaurants: well-known domain to ChatGPT
  - Small table: 11 fields (incl. 2 Free-text, 1 small, 1 large ENUM)
  - Well-understood field names
- Open questions
  - BIRD: Can LLM serve as a DB interface? SOTA: 40%
    - HW2: Few-shot prompting of a single domain in BIRD
    - Students get experience and insight into an open question

BIRD: inyang Li et al. Can LLM already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. https://arxiv.org/pdf/2305.03111.pdf
HybridQA: https://aclanthology.org/2020.findings-emnlp.91/

LAM 40 STANFORD

MIT newest paper about code gen

SEED Components. We use this task to evaluate the **LLM query** component, in particular, our **tools usage** optimization.
Datasets. We used the Bird-SQL Benchmark [35] in the experiments, which is a comprehensive collection of well-annotated NL2SQL test cases, spanning across 37 distinct data domains. Each test case is associated with a single database and is supplemented with corresponding expert knowledge to facilitate the translation process. The training and dev dataset is open to public access, while the test dataset is held privately by the Bird-SQL Benchmark team. As the test set of Bird-SQL is held privately, we randomly selected 150 queries from the Dev dataset for evaluation.
Evaluation Metric. We measure the quality of the NL2SQL translation with two metrics officially recommended on Bird-SQL [35]: **Execution Accuracy (EX)** and **Valid Efficiency Score (VES)**. Execution Accuracy measures the number of SQL statements that are executable and yield correct responses. On the other hand, the Valid Efficiency Score assesses the efficiency of correctly executed SQL statements by comparing their execution time with a gold SQL reference.

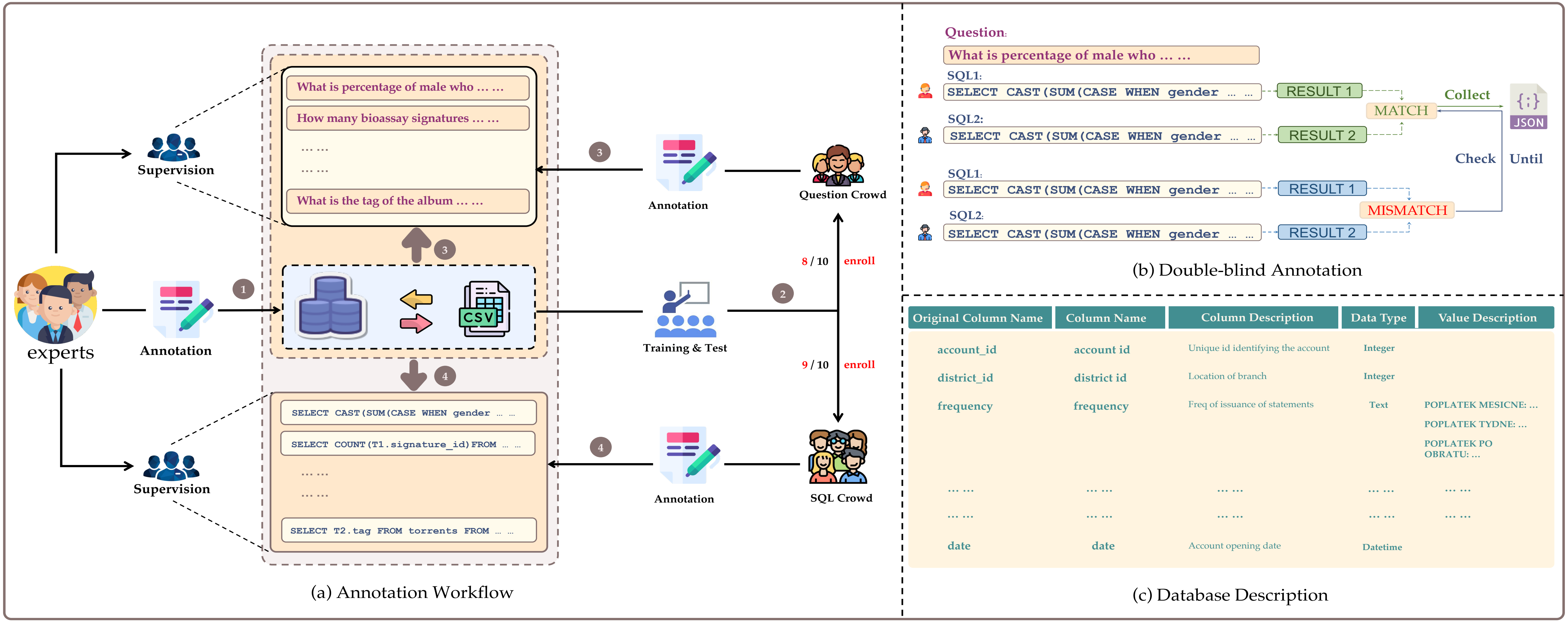
Tsinghua University (Prof. Jie Tang) → ChatGLM 3.0

Task Derivation For agent tasks associated with scenarios that have been widely studied, we can directly construct instructions from similar datasets. Thus to construct instructions on the Database (DB) task, we derive instructions from BIRD (Li et al., 2023), a SELECT-only database benchmark. We ran two types of task derivation. First, we construct a trajectory using the question and the reference SQL statement in each BIRD subtask. We then query the database using the reference SQL statement to obtain output of the database and serve it as the submitted answer of the agent. Finally, we ask GPT-4 to fill in the thoughts of the agent given the above information. In this way, we can generate correct trajectories directly from BIRD dataset.

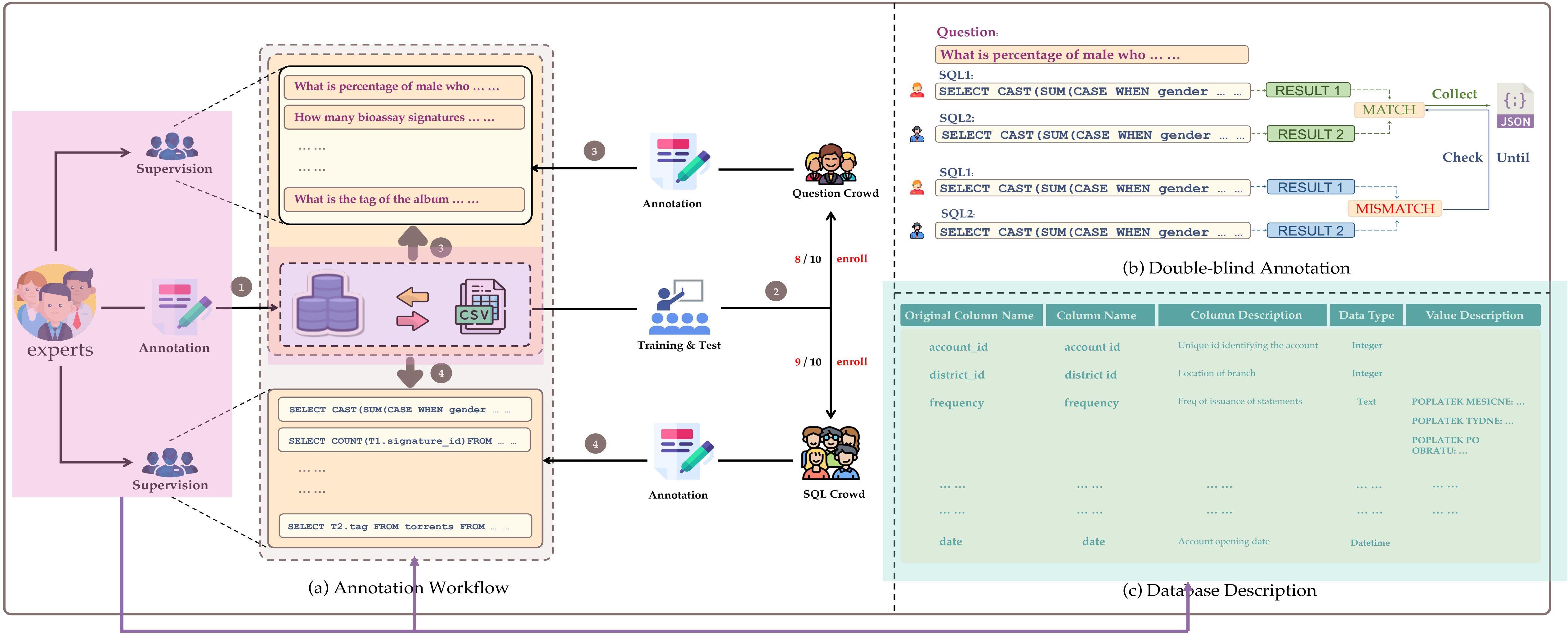
Self-Instruct For the Operating System (OS) task, due to the difficulty in obtaining instructions that involve manipulating OS in terminal, we employed the Self-Instruct method (Wang et al., 2023c) to construct the task. We first prompt GPT-4 to come up with some OS related tasks along with explanations to the task, a reference solution and an evaluation script. Then, we prompt another GPT-4 instance (the solver) with the task and collect its trajectory. After the task is completed, we run the reference solution and compare its result to the one from the solver GPT-4 using the evaluation script. We collect the trajectories where the reference solution and the solver's solution give the same answer. For the DB task, since BIRD only contains SELECT data, we construct other types of database operations (INSERT, UPDATE and DELETE) in a similar self-instruct approach.

https://bird-bench.github.io/

Dataset Construction

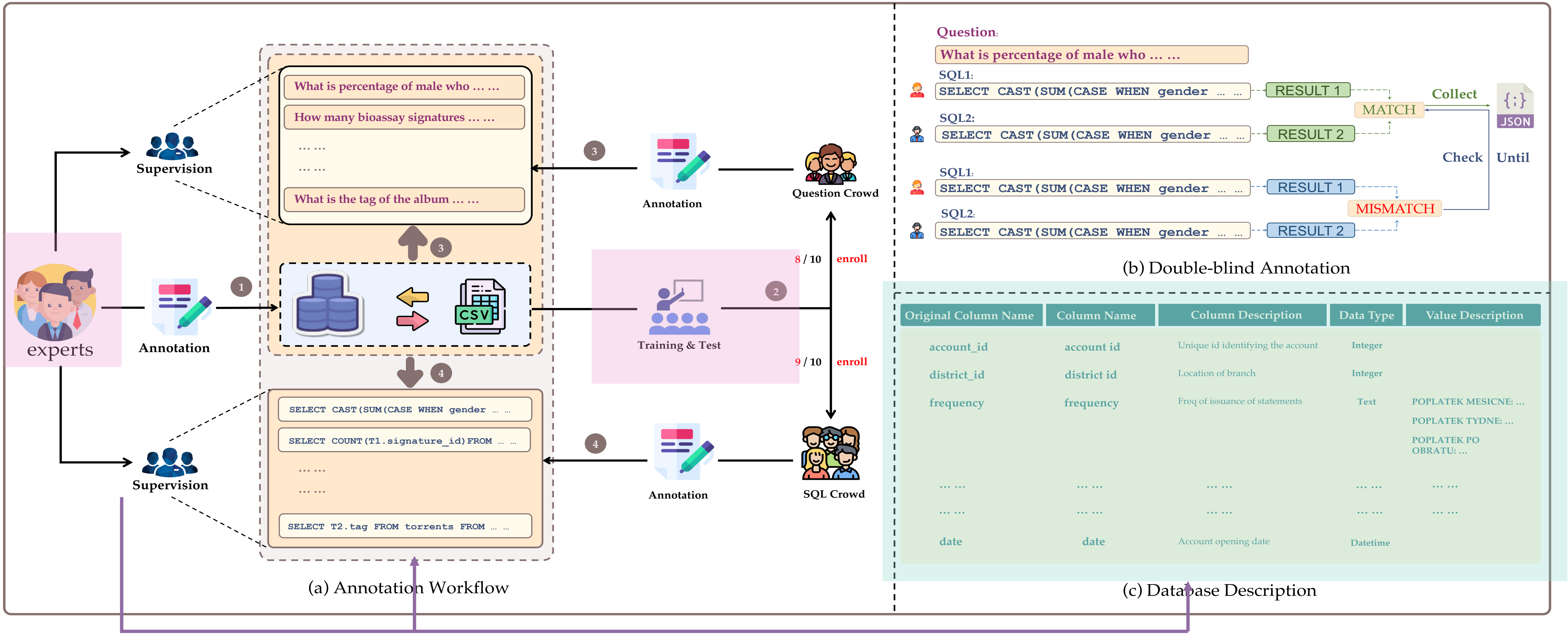


Dataset Construction



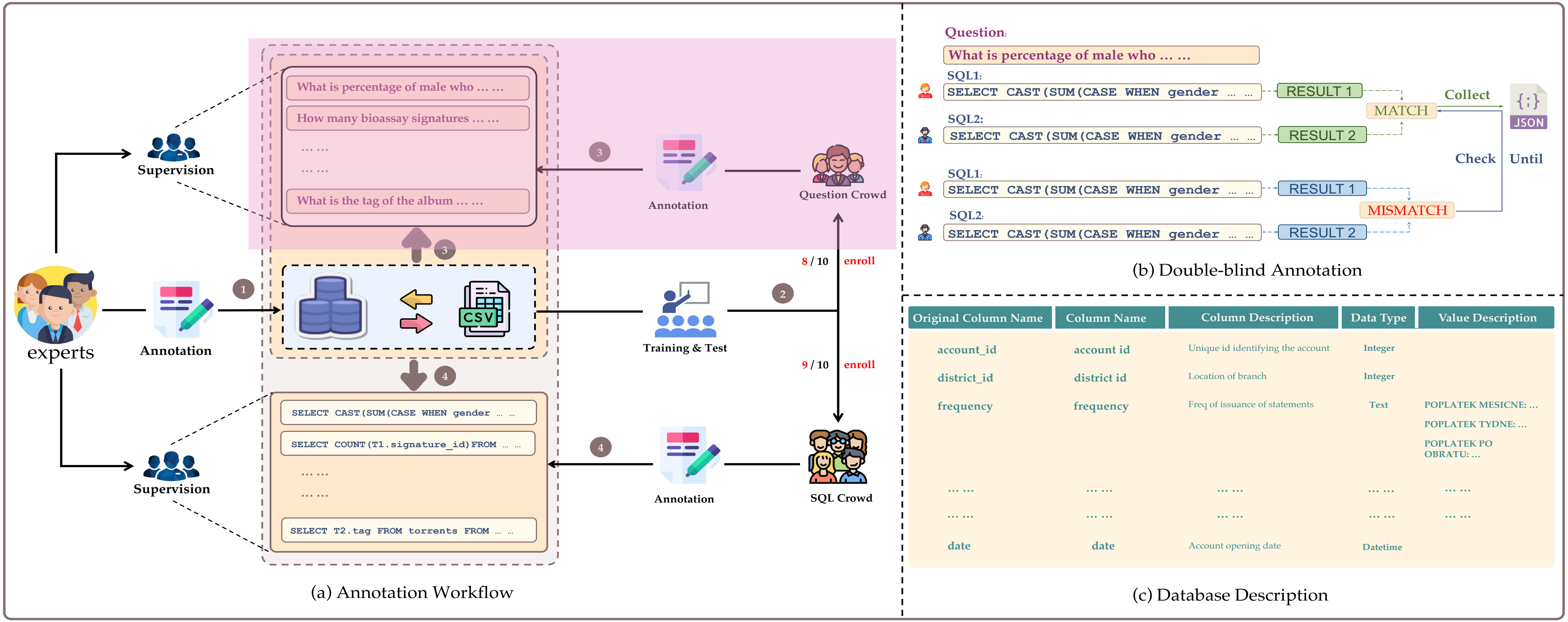
Step 1: Experts assemble and produce databases and description files.

Dataset Construction



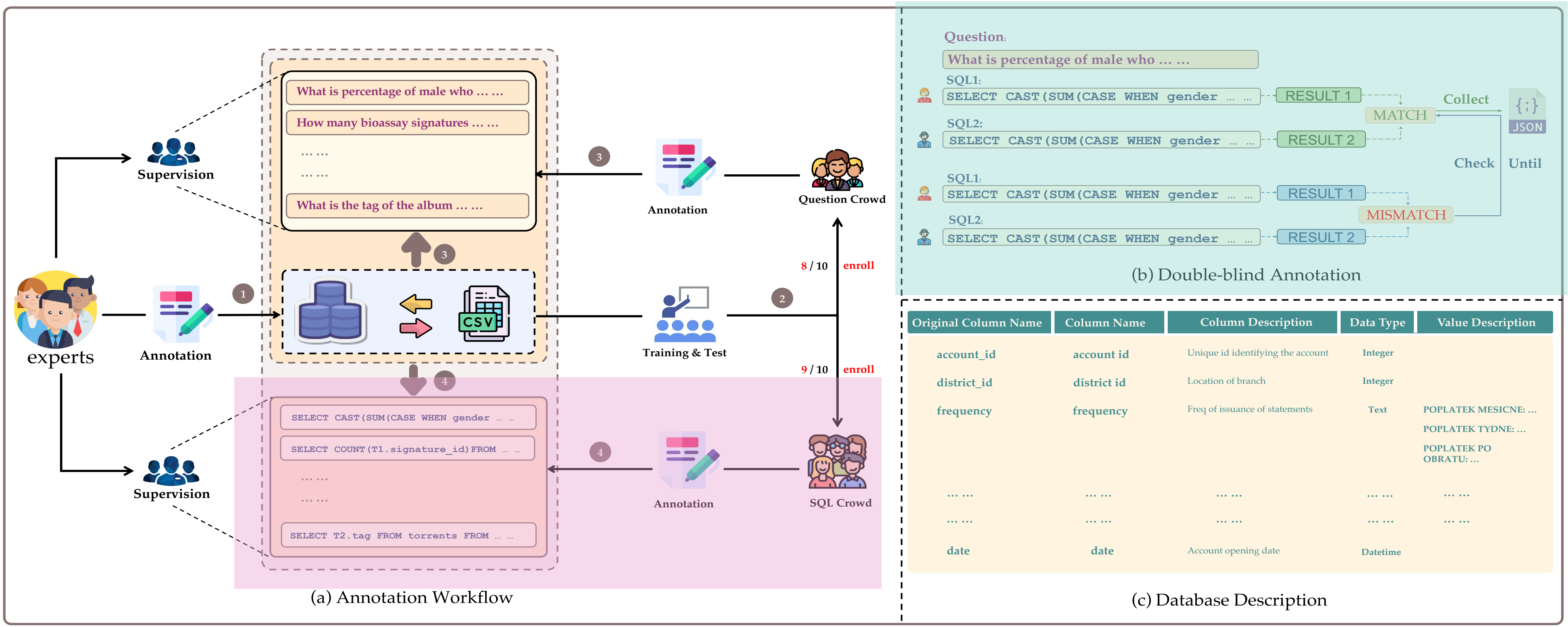
Step 2: Experts teach and evaluate crowdsourcing people.

Dataset Construction



Step 3: Question annotators create a corpus of questions using databases and their corresponding description files.

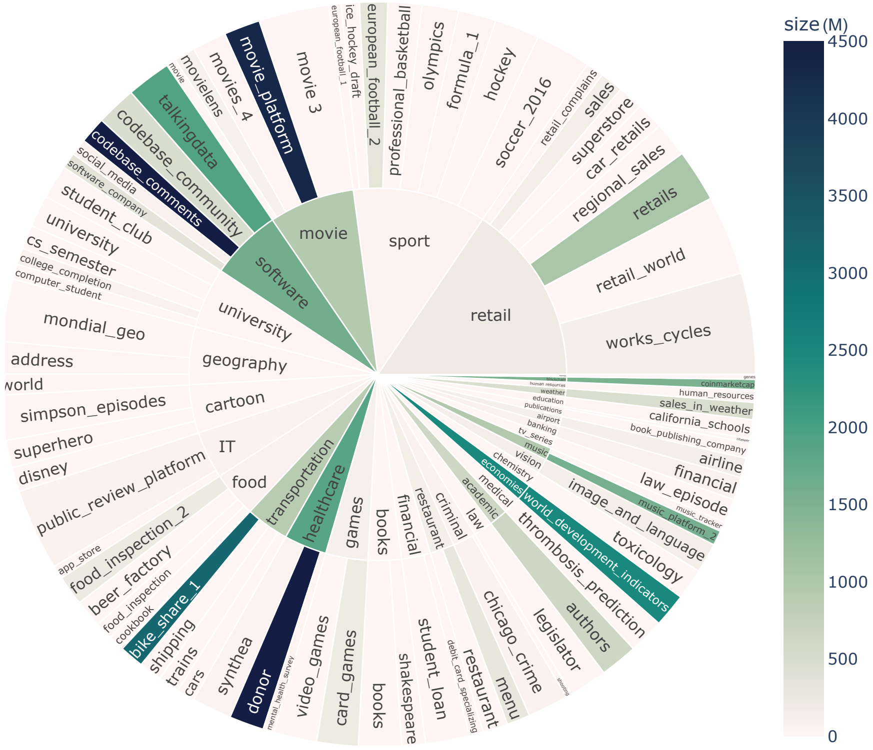
Dataset Construction



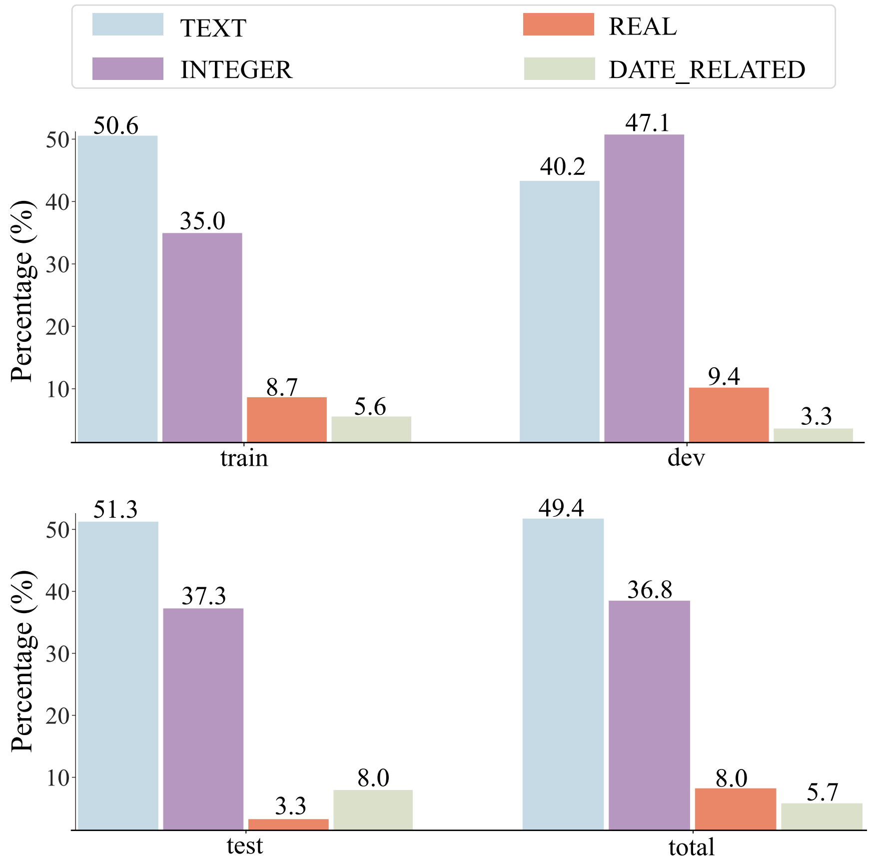
Step 4: SQL annotators produce SQL files, equipped with databases, descriptions, and questions

Can LLM Already Serve as A Database Interface?

BIRD: A BIg Bench for Large-Scale Database Grounded Text-to-SQLs



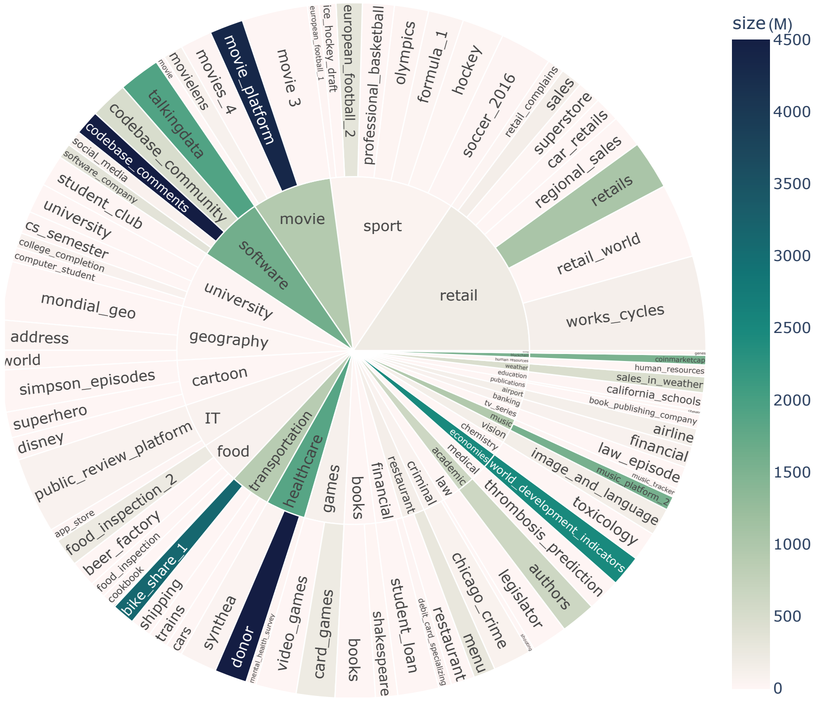
a) Database domain distribution w/ size



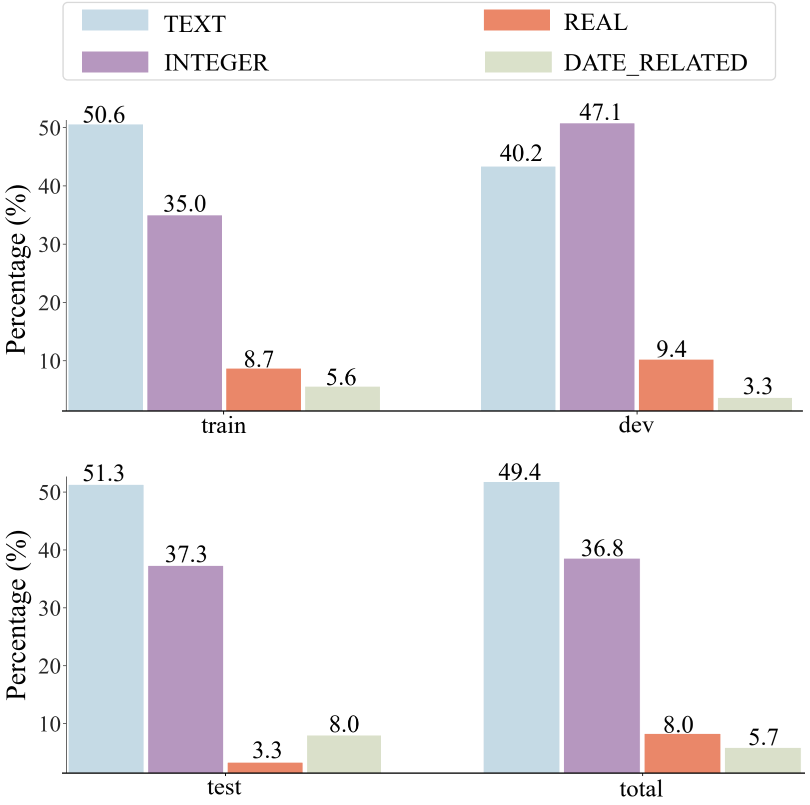
b) Database value type distribution

Can LLM Already Serve as A Database Interface?

BIRD: A BIg Bench for Large-Scale Database Grounded Text-to-SQLs



a) Database domain distribution w/ size



b) Database value type distribution

12,751 text-to-SQL pairs over **95** big databases with a total size of **33.4** GB spanning **37** domains

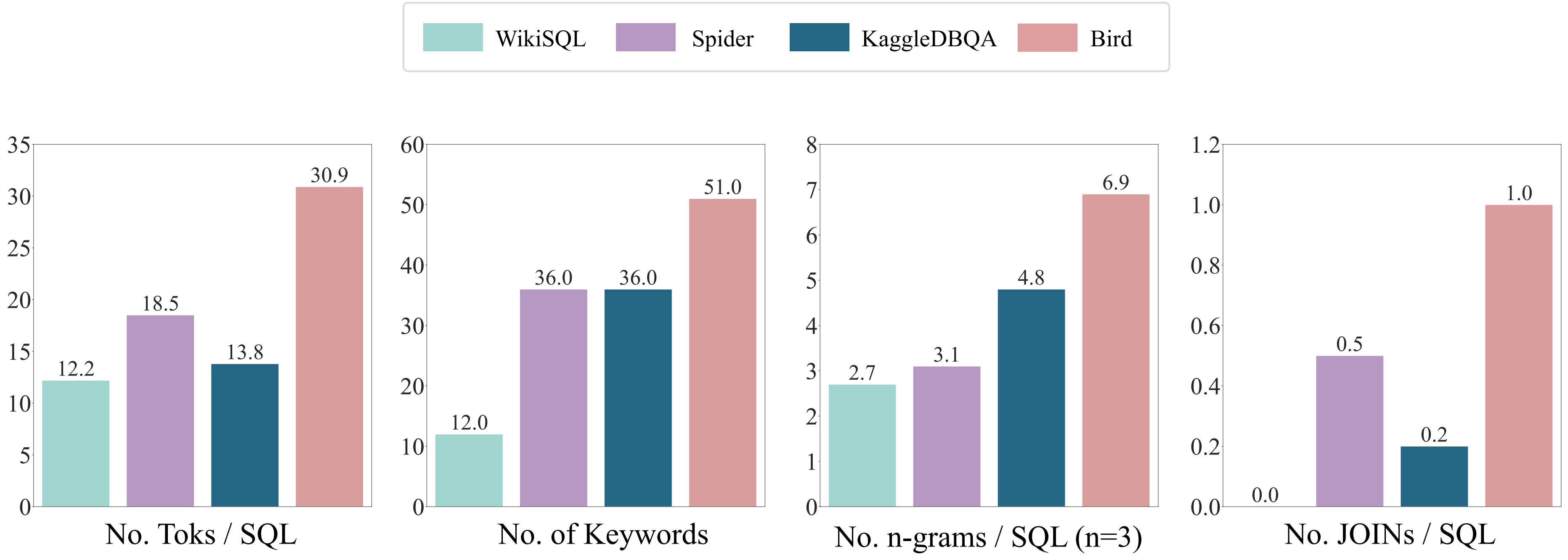
80 open-source relational databases for training

15 additional relational databases for evaluation

Data Statistics

Dataset	# Example	# DB	# Table/DB	# Row/DB	Function	Knowledge	Efficiency
WikiSQL [60]	80,654	26,521	1	17	✘	✘	✘
Spider [55]	10,181	200	5.1	2K	✘	✘	✘
KaggleDBQA [25]	272	8	2.3	280K	✘	✓	✘
BIRD	12,751	95	7.3	549K	✓	✓	✓

An overview comparison between BIRD and other cross-domain text-to-SQL benchmarks.



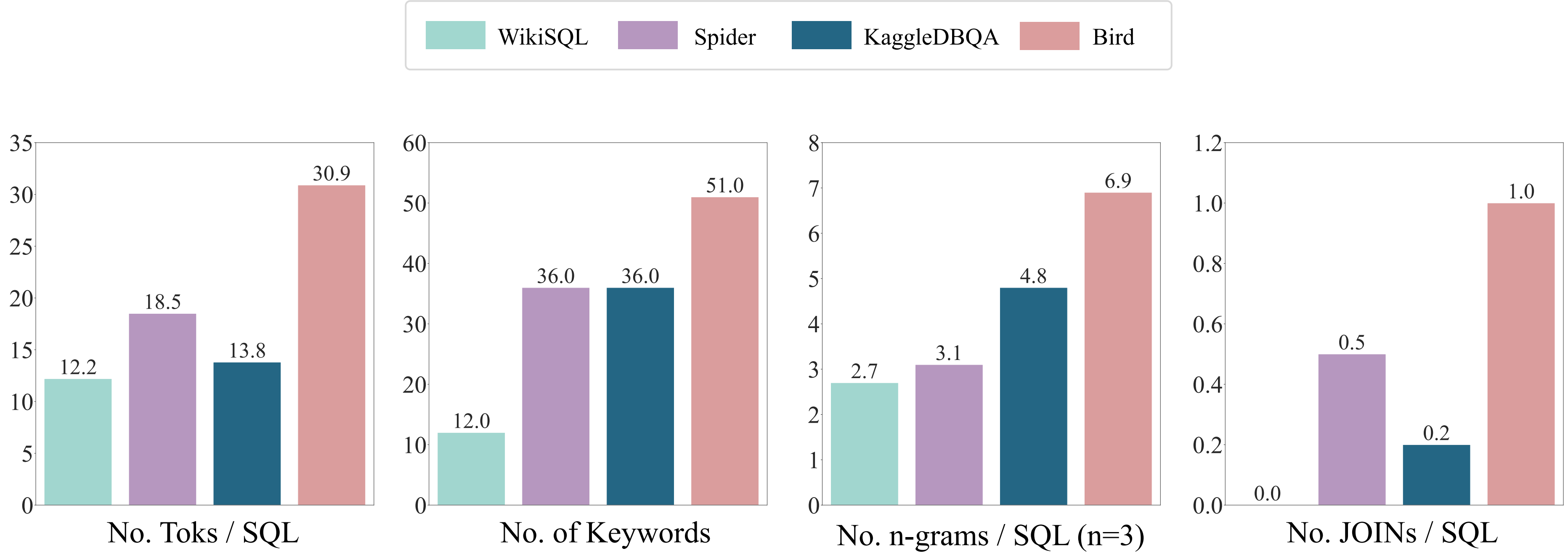
A comparative statistical analysis of SQL queries in the BIRD dataset and other benchmarks

Data Statistics

Dataset	# Example	# DB	# Table/DB	# Row/DB	Function
WikiSQL [60]	80,654	26,521	1	17	✗
Spider [55]	10,181	200	5.1	2K	✗
KaggleDBQA [25]	272	8	2.3	280K	✗
BIRD	12,751	95	7.3	549K	✓

- Window Functions, i.e., OVER()
- Date Functions, i.e., JULIANDAY()
- Conversion Functions, i.e., CAST()
- Math Functions, i.e., ROUND()
- String Functions, i.e., SUBSTR()

An overview comparison between BIRD and other cross-domain text-to-SQ



A comparative statistical analysis of SQL queries in the BIRD dataset and other benchmarks

Data Statistics

Dataset	# Example	# DB	# Table/DB	# Row/DB	Function	Knowledge
WikiSQL [60]	80,654	26,521	1	17	✘	✘
Spider [55]	10,181	200	5.1	2K	✘	✘
KaggleDBQA [25]	272	8	2.3	280K	✘	✓
BIRD	12,751	95	7.3	549K	✓	✓

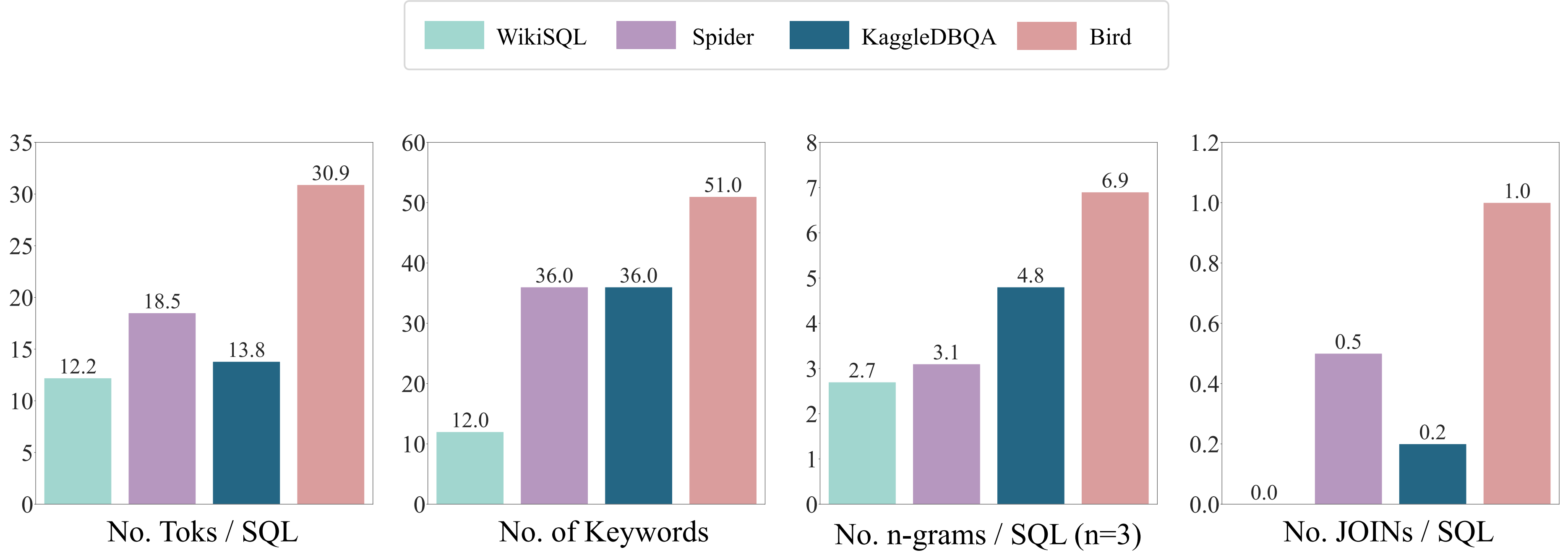
An overview comparison between BIRD and other cross-domain text-to-SQL benchmarks.

- External Knowledge

  winning rate = #won / #games

- Self-contained Value Knowledge

  POPLAKE TYDNE refers to weekly issuance



A comparative statistical analysis of SQL queries in the BIRD dataset and other benchmarks

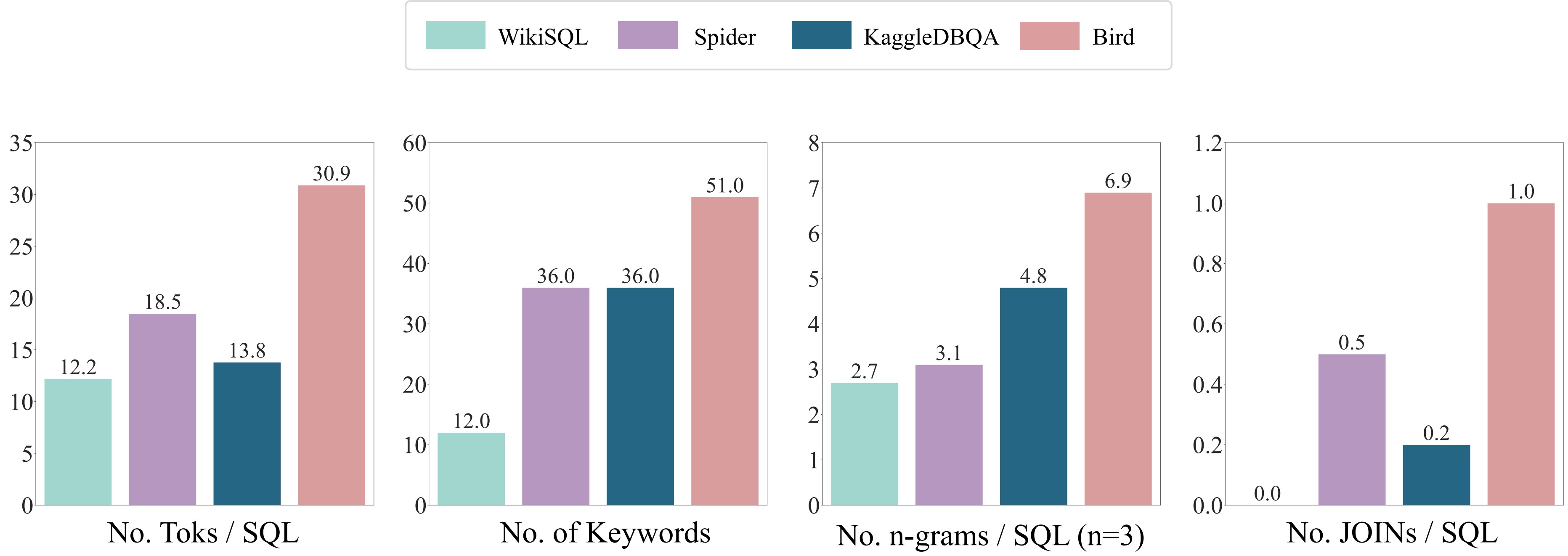
Data Statistics

Dataset	# Example	# DB	# Table/DB	# Row/DB	Function	Knowledge	Efficiency
WikiSQL [60]	80,654	26,521	1	17	✗	✗	✗
Spider [55]	10,181	200	5.1	2K	✗	✗	✗
KaggleDBQA [25]	272	8	2.3	280K	✗	✓	✗
BIRD	12,751	95	7.3	549K	✓	✓	✓

SQL Execution Efficiency:

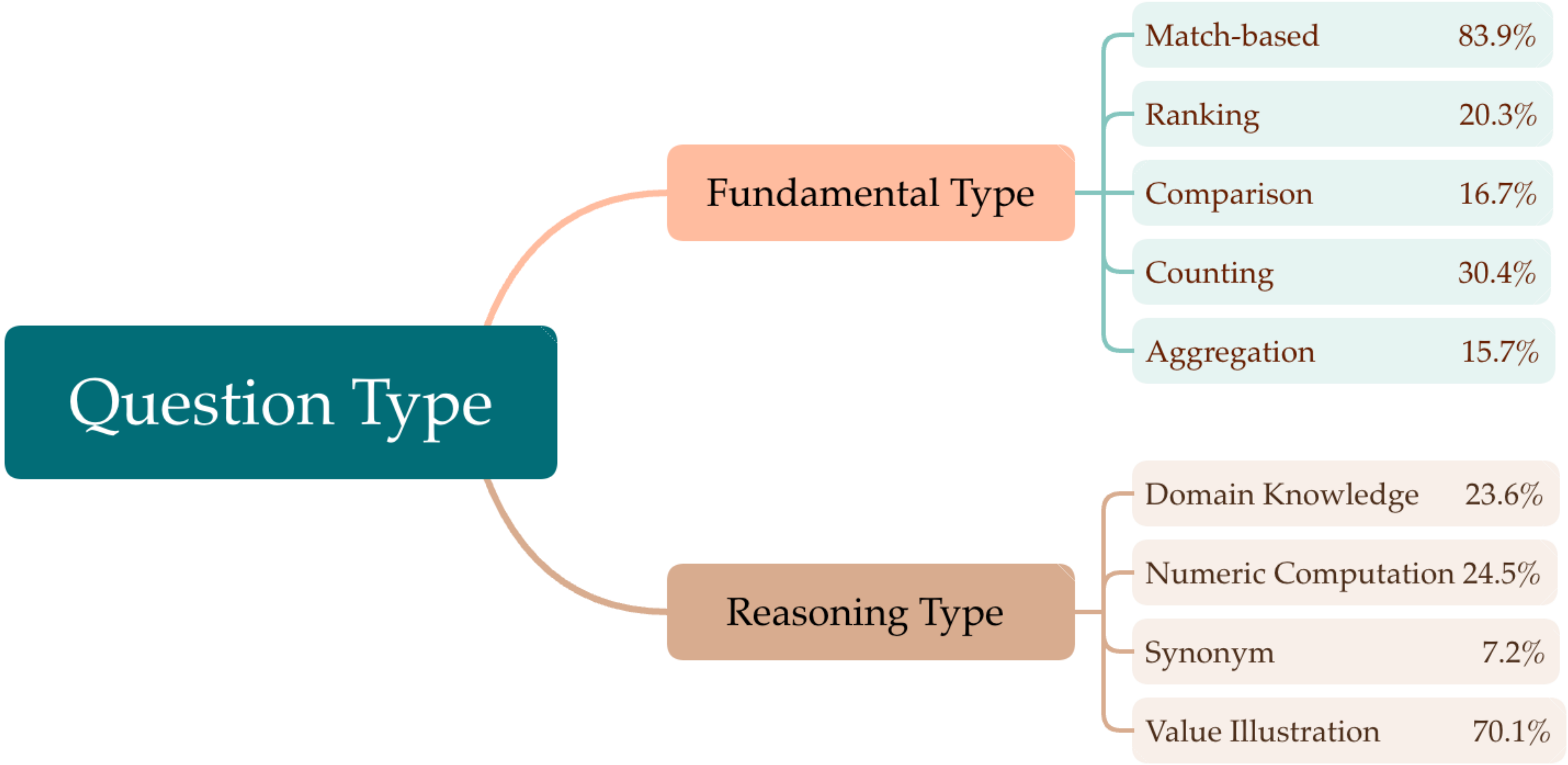
24s vs 4s

An overview comparison between BIRD and other cross-domain text-to-SQL benchmarks.



A comparative statistical analysis of SQL queries in the BIRD dataset and other benchmarks

Question Statistics



Question Statistics

Question Type	Sub Type	Question / SQL
Fundamental Type	Match-based	How many gas stations in **CZE** has **Premium** gas? `SELECT COUNT(GasStationID) FROM gasstations WHERE Country = 'CZE' AND Segment = 'Premium'`
	Ranking	What are the titles of the **top 5** posts with the highest popularity? `SELECT Title FROM posts ORDER BY ViewCount DESC LIMIT 5`
	Comparison	How many color cards with no borders have been ranked **higher than 12000** on EDHRec? `SELECT COUNT(id) FROM cards WHERE edhrecRank > 12000 AND borderColor = 'borderless'`
	Counting	**How many** of the members' hometowns are from Maryland state? `SELECT COUNT(T2.member_id) FROM zip_code AS T1 INNER JOIN member AS T2 ON T1.zip_code = T2.zip WHERE T1.state = 'Maryland'`
	Aggregation	What is the **average height** of the superheroes from Marvel Comics? `SELECT AVG(T1.height_cm) FROM superhero AS T1 INNER JOIN publisher AS T2 ON T1.publisher_id = T2.id WHERE T2.publisher_name = 'Marvel Comics'`

Question Type	Sub Type	Question / SQL
Reasoning Type	Domain Knowledge	Name the ID and age of patient with two or more laboratory examinations which show their **hematoclit level exceeded the normal range.** `SELECT T1.ID, STRFTIME('%Y', CURRENT_TIMESTAMP) - STRFTIME('%Y', T1.Birthday) FROM Patient AS T1 INNER JOIN Laboratory AS T2 ON T1.ID = T2.ID WHERE T1.ID IN ( SELECT ID FROM Laboratory WHERE HCT > 52 GROUP BY ID HAVING COUNT(ID) >= 2 )`
	Numeric Computation	Among the posts with a score of over 20, what is the **percentage** of them being owned by an elder user? `SELECT CAST(SUM(IIF(T2.Age > 65, 1, 0)) AS REAL) * 100 / count(T1.Id) FROM posts AS T1 INNER JOIN users AS T2 ON T1.OwnerUserId = T2.Id WHERE T1.Score > 20`
	Synonym	How many clients opened their accounts in Jesenik branch were **women** ? **(female)** `SELECT COUNT(T1.client_id) FROM client AS T1 INNER JOIN district AS T2 ON T1.district_id = T2.district_id WHERE T1.gender = 'F' AND T2.A2 = 'Jesenik'`
	Value Illustration	Among the **weekly issuance** accounts, how many have a loan of under 200000? `SELECT COUNT(T1.account_id) FROM loan AS T1 INNER JOIN account AS T2 ON T1.account_id = T2.account_id WHERE T2.frequency = 'POPLATEK TYDNE' AND T1.amount < 200000`

Examples of two main question types in the BIRD

Question Statistics

Leaderboard - Execution Accuracy (EX)

	Model	Code	Size	Orale Knowledge	Dev (%)	Test (%)
	Human Performance *Data Engineers + DB Students*			✓		**92.96**
1 Aug 15, 2023	DIN-SQL + GPT-4 *University of Alberta* [Pourreza et al. 2023]	[link]	UNK	✓	50.72	**55.90**
2 Jul 01, 2023	GPT-4 *Baseline*	[link]	UNK	✓	46.35	54.89
3 Jul 16, 2023	Claude-2 *Baseline*	[link]	UNK	✓	42.70	49.02
4 Mar 17, 2023	ChatGPT + CoT *HKU & DAMO* [Li et al. 2023]	[link]	UNK	✓	36.64	40.08
5 Mar 17, 2023	ChatGPT *Baseline*		UNK	✓	37.22	39.30
6 Feb 17, 2023	Codex *Baseline*		175B	✓	34.35	36.47
7 Jul 16, 2023	Palm-2 *Baseline*	[link]	UNK	✓	27.38	33.04
8 Mar 17, 2023	ChatGPT + CoT *HKU & DAMO* [Li et al. 2023]	[link]	UNK		25.88	28.95
9 Mar 17, 2023	ChatGPT *Baseline*		UNK		24.05	26.77
10 Feb 17, 2023	Codex *Baseline*		175B		25.42	24.86

Leaderboard - Valid Efficiency Score (VES)

	Model	Code	Size	Oracle Knowledge	Dev	Test
	Human Performance *Data Engineers + DB Students*			✓		**90.27**
1 Jul 01, 2023	GPT-4 *Baseline*	[link]	UNK	✓	49.77	**60.77**
2 Aug 15, 2023	DIN-SQL + GPT-4 *University of Alberta* [Pourreza et al. 2023]	[link]	UNK	✓	58.79	59.44
3 Mar 17, 2023	ChatGPT + CoT *HKU & DAMO* [Li et al. 2023]	[link]	UNK	✓	42.30	56.56
4 Mar 17, 2023	ChatGPT *Baseline*		UNK	✓	43.81	51.40
5 Mar 17, 2023	ChatGPT + CoT *HKU & DAMO* [Li et al. 2023]	[link]	UNK		32.33	49.69
6 Feb 17, 2023	Codex *Baseline*		175B	✓	43.41	41.60
7 Mar 17, 2023	ChatGPT *Baseline*		UNK		27.97	36.68
8 Feb 17, 2023	Codex *Baseline*		175B		33.37	35.40
9 Feb 5, 2023	T5-3B *Baseline*		3B	✓	25.57	27.80
10 Feb 3, 2023	T5-Large *Baseline*		770M	✓	22.74	25.00

Execution Accuracy (EX) is defined as the proportion of examples in the evaluation set for which the executed results of both the predicted and ground truth SQLs are identical, relative to the overall number of SQLs

Valid Efficiency Score (VES) is designed to measure the efficiency of valid SQLs generated by models

https://bird-bench.github.io/

Experimental Results

Models	Development Data		Testing Data	
	w/o knowledge	w/ knowledge	w/o knowledge	w/ knowledge
FT-based				
T5-Base	6.32	11.54 (+5.22)	7.06	12.89 (+5.83)
T5-Large	9.71	19.75 (+10.04)	10.38	20.94 (+10.56)
T5-3B	10.37	23.34 (+12.97)	11.17	24.05 (+12.88)
ICL-based				
Codex	25.42	34.35 (+8.93)	24.86	36.47 (+11.61)
ChatGPT	24.05	**37.22** (+13.17)	26.77	39.30 (+12.53)
ChatGPT + COT	**25.88**	36.64 (+10.76)	28.95	40.08 (+11.24)
Human Performance	-	-	**72.37**	**92.96** (+20.59)

The Execution Accuracy (EX)
of SOTA text-to-SQL models in BIRD

Models	Development Data		Testing Data	
	w/o knowledge	w/ knowledge	w/o knowledge	w/ knowledge
FT-based				
T5-Base	7.78	12.90 (+5.12)	8.97	14.71 (+5.74)
T5-Large	9.90	22.74 (+12.84)	12.25	25.00 (+12.75)
T5-3B	13.62	25.57 (+11.95)	15.17	27.80 (+12.63)
ICL-based				
Codex	**33.37**	43.41 (+10.04)	35.40	41.60 (+6.20)
ChatGPT	27.97	**43.81** (+15.84)	36.68	51.40 (+14.72)
ChatGPT + COT	32.33	42.30 (+9.97)	49.69	56.56 (+6.87)
Human Performance	-	-	**70.36**	**90.27** (+19.91)

The Valid Efficiency Score (VES)
of SOTA text-to-SQL models in BIRD

Models	Development Data w/o knowledge	Development Data w/ knowledge	Testing Data w/o knowledge	Testing Data w/ knowledge
Palm-2	18.77	27.38	24.71	33.04
Claude-2	28.29	42.70	34.60	49.02
GPT-4	30.90	46.35	34.88	54.89
GPT-4 + DIN-SQL	-	50.72	-	55.90
Human Performance	-	-	72.37	92.96

The Execution Accuracy (EX) of other powerful LLMs in BIRD

Experimental Results

Models	Development Data		Testing Data	
	w/o knowledge	w/ knowledge	w/o knowledge	w/ knowledge
FT-based				
T5-Base	6.32	11.54 (+5.22)	7.06	12.89 (+5.83)
T5-Large	9.71	19.75 (+10.04)	10.38	20.94 (+10.56)
T5-3B	10.37	23.34 (+12.97)	11.17	24.05 (+12.88)
ICL-based				
Codex	25.42	34.35 (+8.93)	24.86	36.47 (+11.61)
ChatGPT	24.05	**37.22** (+13.17)	26.77	39.30 (+12.53)
ChatGPT + COT	**25.88**	36.64 (+10.76)	28.95	40.08 (+11.24)
Human Performance	-	-	**72.37**	**92.96** (+20.59)

The Execution Accuracy (EX)
of SOTA text-to-SQL models in BIRD

Models	Development Data		Testing Data	
	w/o knowledge	w/ knowledge	w/o knowledge	w/ knowledge
FT-based				
T5-Base	7.78	12.90 (+5.12)	8.97	14.71 (+5.74)
T5-Large	9.90	22.74 (+12.84)	12.25	25.00 (+12.75)
T5-3B	13.62	25.57 (+11.95)	15.17	27.80 (+12.63)
ICL-based				
Codex	**33.37**	43.41 (+10.04)	35.40	41.60 (+6.20)
ChatGPT	27.97	**43.81** (+15.84)	36.68	51.40 (+14.72)
ChatGPT + COT	32.33	42.30 (+9.97)	49.69	56.56 (+6.87)
Human Performance	-	-	**70.36**	**90.27** (+19.91)

The Valid Efficiency Score (VES)
of SOTA text-to-SQL models in BIRD

Models	Development Data w/o knowledge	Development Data w/ knowledge	Testing Data w/o knowledge	Testing Data w/ knowledge
Palm-2	18.77	27.38	24.71	33.04
Claude-2	28.29	42.70	34.60	49.02
GPT-4	30.90	46.35	34.88	54.89
GPT-4 + DIN-SQL	-	50.72	-	55.90
Human Performance	-	-	72.37	92.96

The Execution Accuracy (EX) of other powerful LLMs in BIRD

Experimental Results

Wrong Schema Linking (41.6%)

Question:
List the names of schools with more than 30 differences in enrollments between K-12 and ages 5-17. Please also give the full street address of the schools.
Evidence:
Difference in enrollment = `Enrollment (K-12)` - `Enrollment (Ages 5-17)`
Ground Truth:
```
SELECT T1.School, T1.StreetAbr FROM
schools AS T1 INNER JOIN frpm AS T2 ON
T1.CDSCode = T2.CDSCode WHERE
T2.`Enrollment (K-12)` -
T2.`Enrollment (Ages 5-17)` > 30
```
ChatGPT SQL:
```
SELECT s.School, s.Street, s.City,
s.Zip FROM schools s JOIN frpm f ON
s.CDSCode = f.CDSCode WHERE
f.`Enrollment (K-12)` - f.`Enrollment
(Ages 5-17)` > 30
```

Question:
List out the accounts who have the earliest trading date in 1995.
Evidence:
False
Ground Truth:
```
SELECT account_id FROM trans WHERE
STRFTIME('%Y', date) = '1995' ORDER
BY date ASC LIMIT 1
```
ChatGPT SQL:
```
SELECT account_id FROM account WHERE
date = (SELECT MIN(date) FROM
account WHERE date LIKE '1995%')
```

Misunderstanding Knowledge Evidence (17.6%)

Question:
On average, how much did the Student_Club spend on food for the typical event in the past?
Evidence:
DIVIDE(SUM(spent), COUNT(spent)) where category = 'Food'; 'event in the past' means event_status = 'Closed'
Ground Truth:
```
SELECT SUM(spent) / COUNT(spent) FROM
budget WHERE category = 'Food' AND
event_status = 'Closed'
```
ChatGPT SQL:
```
SELECT DIVIDE(SUM(spent), COUNT(spent))
FROM budget WHERE category = 'Food' AND
event_status = 'Closed'
```

Question :
Name elementary schools in Riverside which the average of average math score for SAT is grater than 400, what is the funding type of these schools?
Evidence:
Average of average math = sum(average math scores) / count(schools)
Ground Truth:
```
SELECT T1.sname, T2.`Charter Funding
Type` FROM satscores AS T1 INNER JOIN
frpm … HAVING CAST(SUM(T1.AvgScrMath)
AS REAL) / COUNT(T1.cds) > 400
```
ChatGPT SQL:
```
SELECT s.School, s.FundingType FROM
schools s … AND sc.rtype = 'E' AND
sc.AvgScrMath > 400 GROUP BY s.School,
s.FundingType
```

Misunderstanding Database Content (40.8%)

Question:
What is the average math score of the school with the lowest average score for all subjects, and in which county is it located?
Evidence:
Average score for all subjects can be computed by AvgScrMath + AvgScrRead + AvgScrWrite
Ground Truth:
```
SELECT T1.AvgScrMath, T2.County FROM
satscores AS T1 INNER JOIN schools AS
T2 ON T1.cds = T2.CDSCode WHERE
T1.AvgScrMath IS NOT NULL ORDER BY
T1.AvgScrMath + T1.AvgScrRead +
T1.AvgScrWrite ASC LIMIT 1
```
ChatGPT SQL:
```
SELECT cname, AvgScrMath FROM satscores
JOIN schools ON satscores.cds =
schools.CDSCode WHERE satscores.rtype =
'S' ORDER BY (AvgScrMath + AvgScrRead +
AvgScrWrite) ASC LIMIT 1
```

Question:
What is the average lap time in milliseconds of all the lap records set on the various circuits in Italy?
Evidence:
average = AVG(milliseconds)
Ground Truth:
```
SELECT CAST(SUM(T1.milliseconds) AS
REAL) / COUNT(T1.lap) FROM lapTimes AS
T1 INNER JOIN races AS T2 on T1.raceId
= T2.raceId INNER JOIN circuits AS T3
on T2.circuitId = T3.circuitId WHERE
T3.country = 'Italy'
```
ChatGPT SQL:
```
SELECT AVG(milliseconds) FROM
lap_records WHERE circuit IN ('Monza',
'Imola', 'Mugello', 'Misano')
```

Syntax Error (3.0%)

Question:
How many meeting events were held in 2020?
Evidence:
meeting events refers to type = 'Meeting'; held in 2020 refers to YEAR(event_date) = 2020
Ground Truth:
```
SELECT COUNT(event_id) FROM event
WHERE type = 'Meeting' AND
STRFTIME('%Y', COLUMN) = '2020'
```
ChatGPT SQL:
```
SELECT COUNT(*) FROM event WHERE type
= 'Meeting' AND YEAR(event_date) =
2020
```

Question:
How much more was customer 7 consuming in April 2013 than customer 5?
Evidence:
April 2013 refers to '201304' in the yearmonth.date
Ground Truth:
```
SELECT SUM(IIF(CustomerID = 7,
Consumption, 0)) - SUM(IIF(CustomerID =
5, Consumption, 0)) FROM yearmonth
WHERE Date = '201304'
```
ChatGPT SQL:
```
SELECT7, (SELECT Consumption FROM
yearmonth WHERE CustomerID = 7 AND Date
= 201304) - (SELECT Consumption FROM
yearmonth WHERE CustomerID = 5 AND Date
= 201304)
```

Error Analysis: 4 major types of error cases are presented

Experimental Results

Category	Simple	Moderate	Challenging	Total
Overall	54.34	34.64	31.70	46.35
		Fundamental Type		
Match-based	60.64	37.37	34.52	51.44
Ranking	32.97	24.76	30.00	30.00
Comparison	58.44	26.09	26.67	40.34
Counting	58.58	37.50	20.51	48.28
Aggregation	44.75	28.41	25.00	34.82
		Reason Type		
Domain knowledge	54.60	35.17	20.41	42.02
Numeric computation	34.78	18.89	25.00	24.47
Synonym	53.19	43.84	25.00	46.52
Value illustration	55.13	35.40	26.00	44.19

Fine-grained dev EX results of GPT-4 w/ knowledge

Interesting Story About Values

Interaction with GPT4-32K

- GPT4-32k fails to consider the tied results in a joined tables correctly

```
SELECT T1.first_name, T1.last_name, T2.source
FROM member AS T1
INNER JOIN income AS T2 ON T1.member_id = T2.link_to_member
WHERE T2.amount = (
    SELECT MAX(amount)
    FROM income
)
ORDER BY T2.amount DESC
```

```
SELECT T1.first_name, T1.last_name, T2.source
FROM member AS T1
INNER JOIN income AS T2 ON T1.member_id = T2.link_to_member
WHERE T2.amount = (
    SELECT MAX(T4.amount)
    FROM member AS T3
    INNER JOIN income AS T4
    ON T3.member_id = T4.link_to_member
    )
```

Experimental Results

Category	Simple	Moderate	Challenging	Total
Overall	54.34	34.64	31.70	46.35
		Fundamental Type		
Match-based	60.64	37.37	34.52	51.44
Ranking	32.97	24.76	30.00	30.00
Comparison	58.44	26.09	26.67	40.34
Counting	58.58	37.50	20.51	48.28
Aggregation	44.75	28.41	25.00	34.82
		Reason Type		
Domain knowledge	54.60	35.17	20.41	42.02
Numeric computation	34.78	18.89	25.00	24.47
Synonym	53.19	43.84	25.00	46.52
Value illustration	55.13	35.40	26.00	44.19

Fine-grained dev EX results of GPT-4 w/ knowledge

Interesting Story About Values

Interaction with GPT4-32K

- GPT4-32k fails to consider the tied results in a joined tables correctly
- GPT4 struggles to perform well in addressing numeric computation problems in text-to-SQL

Experimental Results

Category	Simple	Moderate	Challenging	Total
Overall	54.34	34.64	31.70	46.35
		Fundamental Type		
Match-based	60.64	37.37	34.52	51.44
Ranking	32.97	24.76	30.00	30.00
Comparison	58.44	26.09	26.67	40.34
Counting	58.58	37.50	20.51	48.28
Aggregation	44.75	28.41	25.00	34.82
		Reason Type		
Domain knowledge	54.60	35.17	20.41	42.02
Numeric computation	34.78	18.89	25.00	24.47
Synonym	53.19	43.84	25.00	46.52
Value illustration	55.13	35.40	26.00	44.19

Fine-grained dev EX results of GPT-4 w/ knowledge

Interesting Story About Values

Interaction with GPT4-32K

- GPT4-32k fails to consider the tied results in a joined tables correctly
- GPT4 struggles to perform well in addressing numeric computation problems in text-to-SQL
- GPT4 still lacks the capacity to comprehend complicated values and suffers hallucinations.

We hypothesize that GPT-4 is pre-trained based on semantic parsing objectiveness, losing the enough attention on values.

Conclusion:

- We introduce BIRD, an English large-scale cross-domain, text-to-SQL benchmark with a particular focus on large database contents.

- BIRD mitigates the gap between text-to-SQL research and **real-world** applications by exploring three additional challenges:
    - Handling large and dirty database values
    - External knowledge reasoning
    - Optimizing SQL execution efficiency

- Our experimental results demonstrate that BIRD presents a more **daunting** challenge and leaves plenty of room for improvement and innovation in the text-to-SQL tasks.

- Our thorough efficiency and error analyses provide valuable insights and directions for future research.

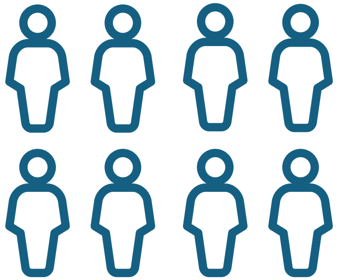
High-Quality Benchmark Construction Suggestions:

- Recruit Reliable People directly!

Bachelor degree

- Correct value
- Good understanding
- Knowledgeable

Much Better Than



Normal or Unknown People

High-Quality Benchmark Construction Suggestions:

- Recruit Reliable People directly!

- Taxonomy Before Annotations!

2. **Collection Strategy:** tagging staff can generate questions according to but not limited to the following categories of questions.

   a. **Match-based questions:** how many teams come from 'EA'?

   b. **Span-based questions:** Please list the top three teams with the most shots in the year:

   c. **Comparison question:** how many team has more than or equal to (not less than) 200 attempts in a single year?

   d. **Counting question:** how many teams in the NBL scored more than 400 points in 1937?

   e. **Addition question:** from 1945 to 1947, what was the total number of shots made by NYK team? (486 + 647 + 251)

   f. **Subtraction (or negative meaning) question:** 1) how many NBA teams won no more than 10 home games in 2000? 2) Among the teams from 'EA', how many teams won no more than 10 home games: (20350 – 14777)

   g. **Aggregation questions:** involving the largest (max), smallest (min) and average questions. For example, in 1945, which team took the most / least attempts? What was the average number of field goal mades by all teams in 1945?

   h. **Division questions(difficult, please give the formula if involved, for example):** in 1946,

how many teams whose winning rate are there more than 70%? Calculation: winning rate = won / won + lost

   i. **Combinatorial questions (it is difficult, please give a certain formula, for example).** Please list the full names of the teams with the fastest growth in winning rate from 1960 to 1961. Calculation: increase of winning rate = [won_1961 / (won_1961 + lost_1961)] – [won_1960 / (won_1960 + lost_1960)]

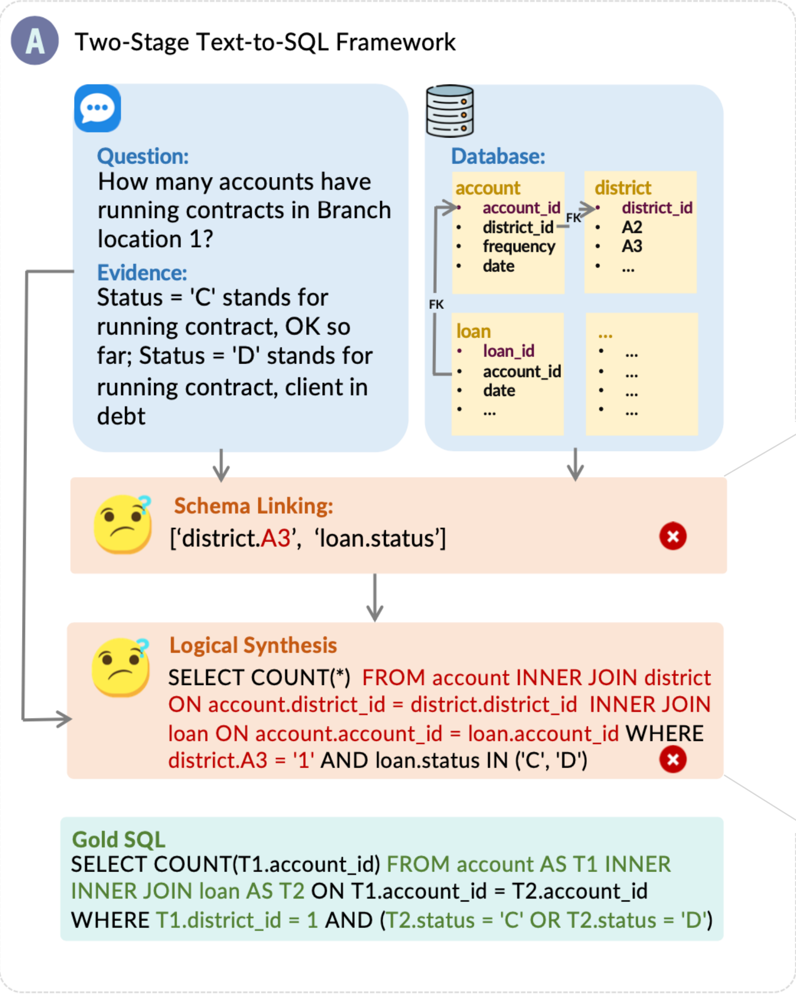
   j. **Inference question:** this question needs to be inferred by describing the information content. How many accounts are eligible for loans? (only when the account type is "owner" can the account information have the loan qualification, which is stated in the disp_id table.)

Question Type	Sub Type	Question / SQL	Percentage
Fundamental Type	Match-based	How many gas stations in CZE has Premium gas? `SELECT COUNT(GasStationID) FROM gasstations WHERE Country = 'CZE' AND Segment = 'Premium'`	83.9 %
	Ranking	What are the titles of the top 5 posts with the highest popularity? `SELECT Title FROM posts ORDER BY ViewCount DESC LIMIT 5`	20.3 %
	Comparison	How many color cards with no borders have been ranked higher than 12000 on EDHRec? `SELECT COUNT(id) FROM cards WHERE edhrecRank > 12000 AND borderColor = 'borderless'`	16.7 %
	Counting	How many of the members' hometowns are from Maryland state? `SELECT COUNT(T2.member_id) FROM zip_code AS T1 INNER JOIN member AS T2 ON T1.zip_code = T2.zip WHERE T1.state = 'Maryland'`	30.4 %
	Aggregation	What is the average height of the superheroes from Marvel Comics? `SELECT AVG(T1.height_cm) FROM superhero AS T1 INNER JOIN publisher AS T2 ON T1.publisher_id = T2.id WHERE T2.publisher_name = 'Marvel Comics'`	15.7 %
Reasoning Type	Domain Knowledge	Name the ID and age of patient with two or more laboratory examinations which show their hematoclit level exceeded the normal range. `SELECT T1.ID, STRFTIME('%Y', CURRENT_TIMESTAMP) - STRFTIME('%Y', T1.Birthday) FROM Patient AS T1 INNER JOIN Laboratory AS T2 ON T1.ID = T2.ID WHERE T1.ID IN ( SELECT ID FROM Laboratory WHERE HCT > 52 GROUP BY ID HAVING COUNT(ID) >= 2 )`	23.6 %
	Numeric Computation	Among the posts with a score of over 20, what is the percentage of them being owned by an elder user? `SELECT CAST(SUM(IIF(T2.Age > 65, 1, 0)) AS REAL) * 100 / count(T1.Id) FROM posts AS T1 INNER JOIN users AS T2 ON T1.OwnerUserId = T2.Id WHERE T1.Score > 20`	24.5 %
	Synonym	How many clients opened their accounts in Jesenik branch were women ? (female) `SELECT COUNT(T1.client_id) FROM client AS T1 INNER JOIN district AS T2 ON T1.district_id = T2.district_id WHERE T1.gender = 'F' AND T2.A2 = 'Jesenik'`	7.2 %
	Value Illustration	Among the weekly issuance accounts, how many have a loan of under 200000? `SELECT COUNT(T1.account_id) FROM loan AS T1 INNER JOIN account AS T2 ON T1.account_id = T2.account_id WHERE T2.frequency = 'POPLATEK TYDNE' AND T1.amount < 200000`	70.1 %

High-Quality Benchmark Construction Suggestions:

- Recruit reliable people directly!
- Taxonomy Before Annotations!
- First Annotation w/o Fixing can be considered as human performance
- Can Double-Blind Annotations be cheaper?
- Interactive Environment Setting is quite realistic!

Task Alignment: A Novel and Effective Strategy for Mitigating Hallucinations in Text-to-SQL Generation



Primary Hallucinations in Current Text-to-SQL Framework

Hallucination: The generation of content that is irrelevant, erroneous, or inconsistent with user intents.

Schema-Based	**Example**
Schema Contradiction (30%)	**Question:** *What language is the set of 180 cards that belongs to the Ravnica block translated into?* **Gold:** SELECT T2.language FROM sets AS T1 INNER JOIN set_translations AS T2 ON WHERE T1.block = 'Ravnica' AND T1.baseSetSize = 180 **Wrong SQL:** SELECT language FROM sets WHERE baseSetSize = 180 AND block = 'Ravnica'
Attribute Overanalysis (49%)	**Question:** *Which player is the tallest?* **Gold:** SELECT player_name FROM Player ORDER BY height DESC LIMIT 1 **Wrong SQL:** SELECT player_name, height FROM Player ORDER BY height DESC LIMIT 1
Value Misrepresentation (24%)	**Question:** *Give the race of the blue-haired men superhero.* **Gold:** SELECT ... WHERE colour.colour = 'Blue' AND gender.gender = 'Male' **Wrong SQL:** SELECT ... WHERE colour.colour = 'blue' AND gender.gender = 'M'
Logic-Based	**Example**
Join Redundancy (15%)	**Question:** *Determine the bond type formed in the chemical compound containing element Tellurium.* **Gold:** SELECT T2.bond_type FROM atom AS T1 INNER JOIN bond AS T2 ON WHERE T1.element = 'te' **Wrong SQL:** SELECT bond_type FROM bond INNER JOIN connected ON ... INNER JOIN atom ON ... WHERE atom.element = 'te'
Clause Abuse (25%)	**Question:** *Among the posts that were voted by user 14, what is the id of the most valuable post?* **Gold:** SELECT post.Id ... WHERE votes.UserId = 14 ORDER BY post.FavoriteCount DESC LIMIT 1 **Wrong SQL:** SELECT post.Id FROM votes INNER JOIN posts ON ... WHERE votes.UserId = 14 GROUP BY post.Id ORDER BY post.FavoriteCount DESC LIMIT 1
Mathematical Delusion (17%)	**Question:** *What is the percentage of the amount 50 received by the Student_Club among members?* **Gold:** SELECT CAST(SUM(CASE WHEN income.amount = 50 THEN 1.0 ELSE 0 END) AS REAL) * 100 / COUNT(income.income_id) FROM ... WHERE member.position = 'Member' **Wrong SQL:** SELECT DIVIDE(SUM(CASE WHEN income.amount = 50 THEN 1 ELSE 0 END), COUNT(member.member_id)) FROM ... WHERE member.position = 'Member'

Why Hallucinations?

- Insufficient generalization capabilities of LLM
- Arises when models misinterpret tasks as entirely new challenges in which they lack prior training

How do humans deal with it?

Draw on familiar situations

↓

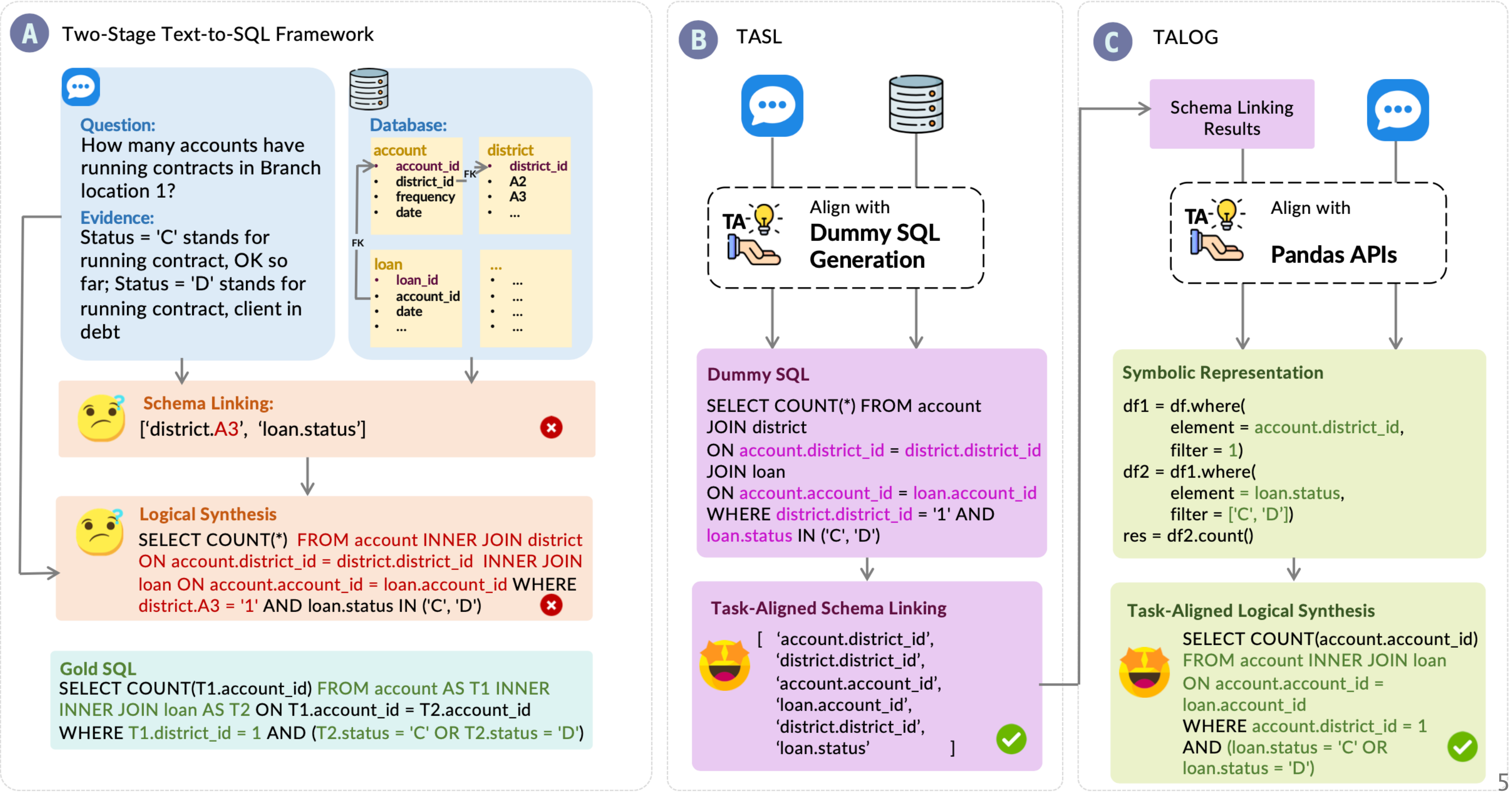
Analogy

Task Alignment

- Align novel tasks to pretrained tasks
- Explicitly guides LLMs to approach unfamiliar tasks from the perspective of more familiar ones, alleviating the burden of from-scratch generalization

TA-SQL

TASQL: Task-Aligned Schema Linking Module (TASL) **(B)** + Task-Aligned Logical Synthesis Module (TALOG) **(C)**



Experimental Results

Results on BIRD

METHOD	DEV	TEST
w/o knowledge		
Palm-2	18.77	24.71
Codex	25.42	24.86
ChatGPT	24.05	26.77
ChatGPT+COT	25.88	28.95
Claude-2	28.29	34.60
GPT-4	30.90	34.88
TA-SQL+GPT-4	**50.58** (↑ 63.68)	**54.38** (↑ 55.90)
w/ knowledge		
Palm-2	27.38	33.04
Codex	34.35	36.47
ChatGPT	37.22	39.30
ChatGPT+COT	36.64	40.08
Claude-2	42.70	49.02
DIN-SQL+GPT-4 ♣	50.72	55.90
DAIL-SQL+GPT-4 ♣	54.76	56.08
GPT-4	46.35	54.89
TA-SQL+GPT-4	**56.19** (↑ 21.23)	**59.14** (↑ 7.74)

Table 2: Execution Accuracy (EX) (%) on BIRD. ♣ means the model uses self-consistency or re-modification mechanisms. ↑ is a relative improvement.

In the setting with oracle knowledge

- TA-SQL effectively mitigates hallucinations in the GPT4 baseline, resulting in a relative improvement of 21.23% in EX on the development set and 7.74% on the test set.
- Surprisingly, TA-SQL equipped with GPT4 outperforms the SOTA ICL-based method by 2.61% even without the application of self-consistency or re-modification mechanisms

In the setting without oracle knowledge

- TA-SQL achieves performance comparable to the GPT4 baseline equipped with oracle external knowledge
- addressing hallucinations within the existing knowledge

VS

the addition of manually extracted external knowledge

New Updates & Next

- **Mini-dev** (Lite version of development dataset)
- 500 high-quality text2sql pairs derived from 11 distinct databases
- Available in MySQL and PostgreSQL

New Updates & Next

- New evaluation metrics (beta versions~) for the Mini-Dev dataset:
  - the **Reward-based Valid Efficiency Score (R-VES)**

Valid Efficiency Score (VES) VES is designed to measure the efficiency of valid SQLs generated by models. It is worth noting that the term "valid SQLs" refers to predicted SQL queries whose result sets align with those of the ground-truth SQLs. Any SQL queries that fail to fetch the correct values will be declared invalid since they are totally useless if they cannot fulfill the user requests, regardless of their efficiency. In this case, the VES metric considers both the efficiency and accuracy of execution results, providing a comprehensive evaluation of a model's performance. Formally, the VES can be expressed as:

$$\text{VES} = \frac{\Sigma_{n=1}^{N} \mathbb{1}(V_n, \hat{V}_n) \cdot \mathbf{R}(Y_n, \hat{Y}_n)}{N}, \quad \boxed{\mathbf{R}(Y_n, \hat{Y}_n) = \sqrt{\frac{\mathbf{E}(Y_n)}{\mathbf{E}(\hat{Y}_n)}}} \tag{4}$$

$$\boxed{\text{R-VES} = \begin{cases} 1.25 & \text{if } \hat{y} \text{ is correct and } \tau \geq 2 \\ 1 & \text{if } \hat{y} \text{ is correct and } 1 \leq \tau < 2 \\ 0.75 & \text{if } \hat{y} \text{ is correct and } 0.5 \leq \tau < 1 \\ 0.5 & \text{if } \hat{y} \text{ is correct and } 0.25 \leq \tau < 0.5 \\ 0.25 & \text{if } \hat{y} \text{ is correct and } \tau < 0.25 \\ 0 & \text{if } \hat{y} \text{ is incorrect} \end{cases}}$$

Where:

- $\hat{y}$ represents the predicted SQL.
- $\tau = \frac{\text{Ground truth SQL run time}}{\text{Predicted SQL run time}}$ represents the time ratio. τ is calculated by running the SQL 100 times, taking the average, and dropping any outliers.

New Updates & Next

- New evaluation metrics (beta versions~) for the Mini-Dev dataset:
  - the **Reward-based Valid Efficiency Score (R-VES)**
  - the **Soft F1-Score**
    - measuring the similarity between the tables produced by predicted SQL queries and those from the ground truth.

Row		
1	'Apple'	325
2	'Orange'	
3	'Banana'	119

Ground truth

Row		
1	325	'Apple'
2	191	'Orange'
3		'Banana'

Predicted

	Matched	Pred_only	Gold_only
Row 1	2	0	0
Row 2	1	1	0
Row 3	1	0	1

- tp = SUM(Matched) = 4
- fp = SUM(Pred_only) = 1
- fn = SUM(Gold_only) = 1
- Precision = tp / (tp + fp) = 4 / 5 = 0.8
- Recall = tp / (tp + fn) = 4 / 5 = 0.8
- F1 = 0.8

Thank you!

More details and updates at
https://bird-bench.github.io/

Any suggestions or feedback are welcome~