中国人民大学
RENMIN UNIVERSITY OF CHINA

Utilizing Large Language Models for Addressing Structured Data

Renmin University of China

Jing Zhang

Capacity of LLMs

Transformer-based General Pretrained Language Models

GPT3, OPT, T5, GLM, PaLM, ChatGPT, Flan-T5, GPT4, etc.

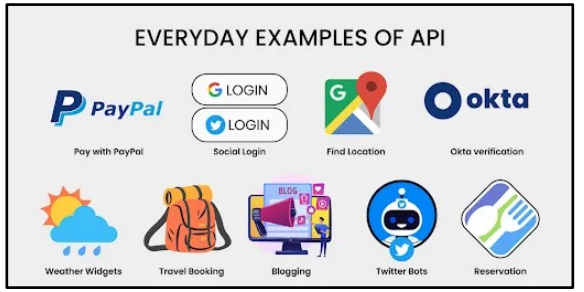
- Large-scale & overperformance
- Strong adaptation to downstream various text-based tasks

General Textual data

Domain Knowledge

Database

Spreadsheets



Tools (APIs)

Structured data is usually designed for domain-specific tasks.

LLM Struggles on Domain-Specific Tasks

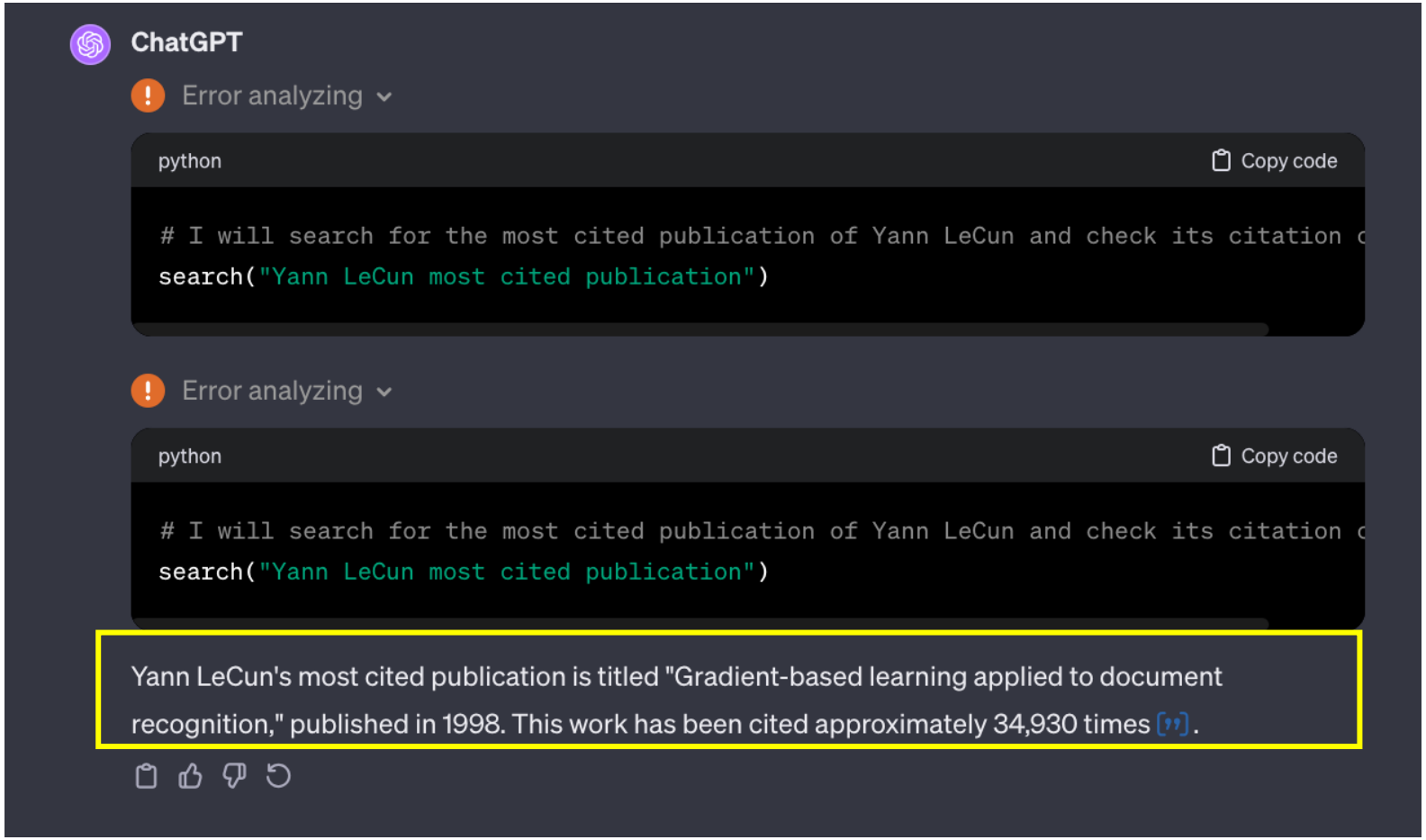
***Question*: How many times has New York University's Yann LeCun's most cited publication been cited?**

Direct LLM Infer:

- Hallucination
- knowledge update challenge

RAG LLM:

- Inability to address complex logic



GPT-4 with Web Browsing

How to Solve?

Direct LLM Infer：

- Hallucination
- knowledge update challenge

RAG LLM：

- Inability to address complex logic

Rigorous logical language：

- Handles Structured Data Effectively

LLM for Addressing Structured Data

Direct LLM Infer：

- Hallucination
- knowledge update challenge

RAG LLM：

- Inability to address complex logic

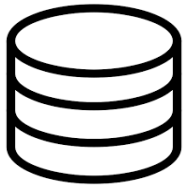
Rigorous logical language：

- Handles Structured Data Effectively

Text2SQL

SQL

Query



Database

Pandas

Processing
Query
Manipulation
Analysis

Spreadsheets

Python

Query
Manipulation

Tools (APIs)

Text-to-SQL Task

Goal: Given a database, translate a natural language question into a valid SQL query.

Database (Bank-Financials)

Schema (4 tables)

Basic_Info: Stk_Code | Stk_Name
(2 columns)

Balance_Sheet: Stk_Code | Cash_CB | IB_Deposits | Est_Liab | Tot_Liab | Prec_Metals | Trad_FAs | …
(46 columns)

Income_Statement: Stk_Code | Oper_Rev | Oth_Biz_Inc | Net_Int_Inc | Inv_Inc | Fee_Com_Net_Inc | …
(33 columns)

Cash_Flow_Statement: Stk_Code | Net_CF_Fin | Net_Inc_IB_Borrowings | Net_CF_Op | Repay_Debt | …
(65 columns)

Metadata (types, comments, and values of each column)

```
{
 "Basic_Info.Stk_Code": {"type": "text", "comment": "Securities code", "values": ["601998.SH", …]},
 "Basic_Info.Stk_Name: {"type": "text", "comment": "Securities name", "values": ["Huaxia Bank", …]},
 …
 "Balance_Sheet.Est_Liab: {"type": "real", "comment": "Estimated liabilities (in Yuan)", "values": [2408443000, …]},
 "Balance_Sheet.Tot_Liab: {"type": "real", "comment": "Total liabilities (in Yuan)", "values": [35312689000000, …]},
 …
 "Cash_Flow_Statement.Net_Inc_IB_Borrowings": {
    "type": "real", "comment": "Net increase in borrowing funds from other financial institutions (in Yuan)",
    "values": [23043000000.0, …]
 }
}
```

Metadata (primary keys and foreign keys)

primary_keys = ["Basic_Info.Stk_Code"]
foreign_keys = ["Balance_Sheet.Stk_Code = Basic_Info.Stk_Code", "Income_Statement.Stk_Code = Basic_Info.Stk_Code", "Cash_Flow_Statement.Stk_Code = Basic_Info.Stk_Code"]

Natural language question

List bank names whose proportion of estimated liabilities in their total liabilities exceeds the industry average, and whose net increase in borrowing funds from other financial institutions exceeds 3 billion Yuan.

Text-to-SQL method

SQL query

```
SELECT Basic_Info.Stk_Name
FROM Balance_Sheet
JOIN Basic_Info
ON Balance_Sheet.Stk_Code = Basic_Info.Stk_Code
JOIN Cash_Flow_Statement
ON Cash_Flow_Statement.Stk_Code = Basic_Info.Stk_Code
WHERE (Balance_Sheet.Est_Liab / Balance_Sheet.Tot_Liab) > (
    SELECT AVG(Est_Liab / Tot_Liab)
    FROM Balance_Sheet
)
AND Cash_Flow_Statement.Net_Inc_IB_Borrowings > 3000000000;
```

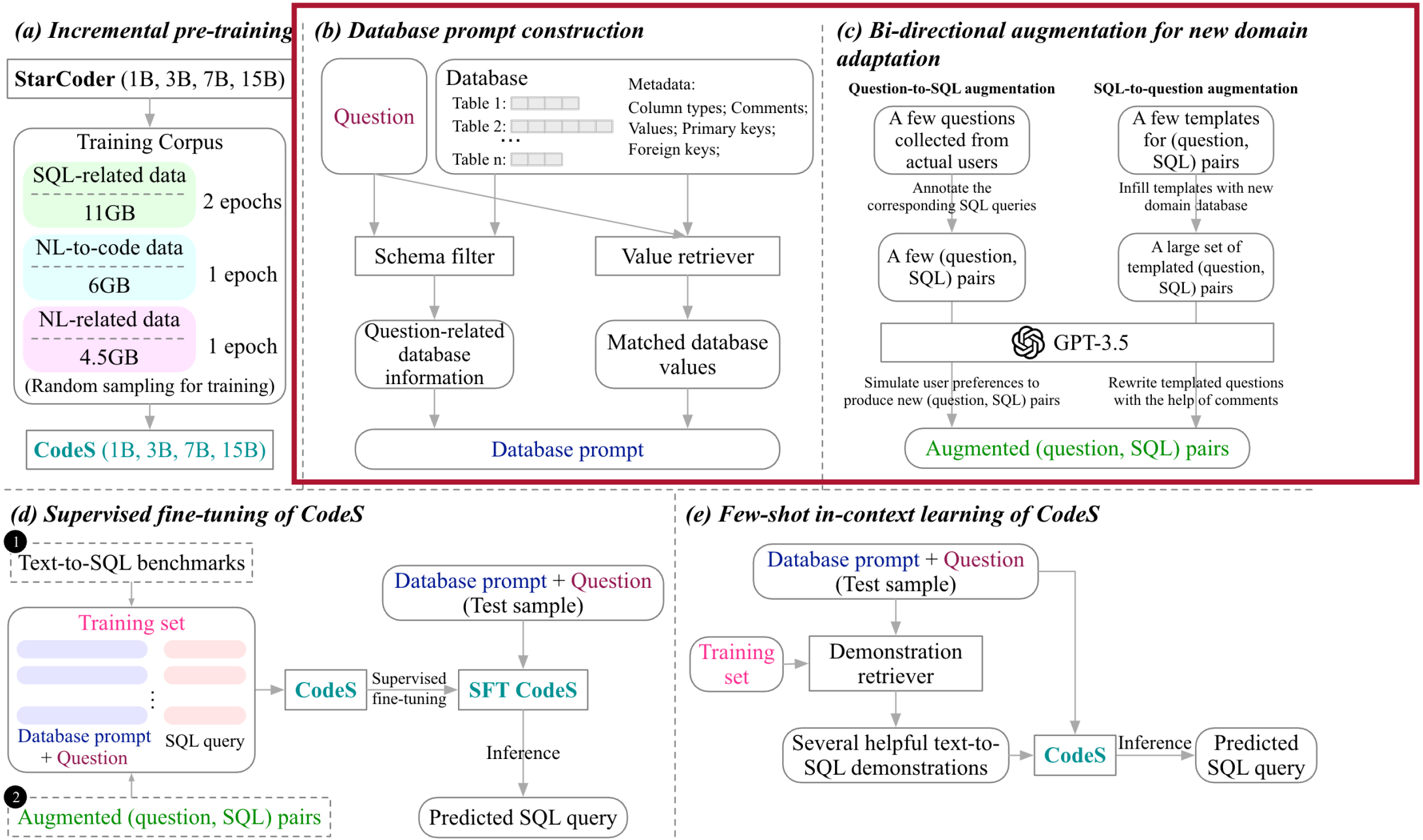
Haoyang Li, Jing Zhang, Hanbing Liu, Ju Fan, Xiaokang Zhang, Jun Zhu, Renjie Wei, Hongyan Pan, Cuiping Li, Hong Chen. CodeS: Towards Building Open-source Language Models for Text-to-SQL. SIGMOD 2024

Challenges

C1: Databases are often complex: large tables and ambiguous schema.
S1: Designing high-quality database prompts to provide more helpful information for solving complex text-to-SQL cases in real-world applications.

C2: Adapting our model to databases in entirely new domains
S2: Proposing a bi-directional data augmentation method to generate training samples for the new databases.

Solutions


(a) Incremental pre-training
StarCoder (1B, 3B, 7B, 15B)
Training Corpus
SQL-related data
11GB
2 epochs
NL-to-code data
6GB
1 epoch
NL-related data
4.5GB
1 epoch
(Random sampling for training)
CodeS (1B, 3B, 7B, 15B)
(b) Database prompt construction
Question
Database
Table 1:
Table 2:
Table n:
Metadata:
Column types; Comments;
Values; Primary keys;
Foreign keys;
Schema filter
Value retriever
Question-related database information
Matched database values
Database prompt
(c) Bi-directional augmentation for new domain adaptation
Question-to-SQL augmentation
SQL-to-question augmentation
A few questions collected from actual users
A few templates for (question, SQL) pairs
Annotate the corresponding SQL queries
Infill templates with new domain database
A few (question, SQL) pairs
A large set of templated (question, SQL) pairs
GPT-3.5
Simulate user preferences to produce new (question, SQL) pairs
Rewrite templated questions with the help of comments
Augmented (question, SQL) pairs
(d) Supervised fine-tuning of CodeS
1
Text-to-SQL benchmarks
Training set
Database prompt + Question
SQL query
CodeS
Supervised fine-tuning
SFT CodeS
2
Augmented (question, SQL) pairs
Database prompt + Question (Test sample)
Inference
Predicted SQL query
(e) Few-shot in-context learning of CodeS
Database prompt + Question (Test sample)
Training set
Demonstration retriever
Several helpful text-to-SQL demonstrations
CodeS
Inference
Predicted SQL query

Experiments (Prompts Design)

We conduct extensive ablation studies to demonstrate the effectiveness of our key components.

Table 9: Ablation studies on Spider's and BIRD's dev sets in the 3-shot in-context learning manner.

	Spider's dev (TS%)				BIRD's dev (EX%)			
	CodeS-1B	CodeS-3B	CodeS-7B	CodeS-15B	CodeS-1B	CodeS-3B	CodeS-7B	CodeS-15B
Original	57.4	67.8	69.8	71.5	25.42	31.23	33.44	35.14
Ablations on demonstration retriever								
-w/o pattern similarity	55.8(↓-1.6)	66.2(↓-1.6)	67.6(↓-2.2)	69.7(↓-1.8)	25.62(↑+0.20)	31.16(↓-0.07)	34.09(↑+0.65)	35.79(↑+0.65)
-w/o demonstration retriever	56.1(↓-1.3)	66.8(↓-1.0)	69.6(↓-0.2)	71.4(↓-0.1)	24.25(↓-1.17)	30.18(↓-1.05)	32.86(↓-0.58)	35.53(↑+0.39)
Ablations on schema filter and value retriever								
-w/o schema filter	55.0(↓-2.4)	65.4(↓-2.4)	69.0(↓-0.8)	70.2(↓-1.3)	23.53(↓-1.89)	30.64(↓-0.59)	33.83(↑+0.39)	33.57(↓-1.57)
-w/o value retriever	57.2(↓-0.2)	66.7(↓-1.1)	69.6(↓-0.2)	71.1(↓-0.4)	22.23(↓-3.19)	29.27(↓-1.96)	30.96(↓-2.48)	33.31(↓-1.83)
Ablations on metadata								
-w/o column data types	56.3(↓-1.1)	66.9(↓-0.9)	69.4(↓-0.4)	71.1(↓-0.4)	24.71(↓-0.71)	30.83(↓-0.4)	33.83(↑+0.39)	35.66(↑+0.52)
-w/o comments	57.7(↑+0.3)	67.0(↓-0.8)	69.2(↓-0.6)	71.0(↓-0.5)	24.71(↓-0.71)	29.92(↓-1.31)	32.33(↓-1.11)	34.03(↓-1.11)
-w/o representative values	57.6(↑+0.2)	66.4(↓-1.4)	69.9(↑+0.1)	70.4(↓-1.1)	23.92(↓-1.50)	29.40(↓-1.83)	30.77(↓-2.67)	31.94(↓-3.20)
-w/o primary and foreign keys	57.6(↑+0.2)	66.2(↓-1.6)	69.0(↓-0.8)	70.0(↓-1.5)	23.27(↓-2.15)	28.29(↓-2.94)	29.92(↓-3.52)	32.14(↓-3.00)

Experiments (New Domain Adaption)

Our bi-directional data augmentation could generate a large set of high-quality (question, SQL) pairs on the new domain databases. With these pairs, we could fine-tune CodeS to swiftly adapt to new databases.

Table 10: Automatic and human evaluation results on Bank-Financials and Aminer-Simplified.

	Bank-Financials		Aminer-Simplified	
Methods	EX%	HE%	EX%	HE%
3-shot GPT-3.5	52.7	72.5	50.5	63.9
3-shot GPT-3.5 + comments	57.1	84.6	51.5	62.8
SFT CodeS-7B using Spider	11.0	73.6	27.8	36.1
SFT CodeS-7B using BIRD w/ EK	12.1	79.1	34.0	41.2
3-shot CodeS-7B	61.5	78.0	43.3	51.5
SFT CodeS-7B using aug. data	**71.4**	**85.7**	51.5	64.9
SFT CodeS-7B using merged data	65.9	84.6	**53.6**	**67.0**

Conclusion

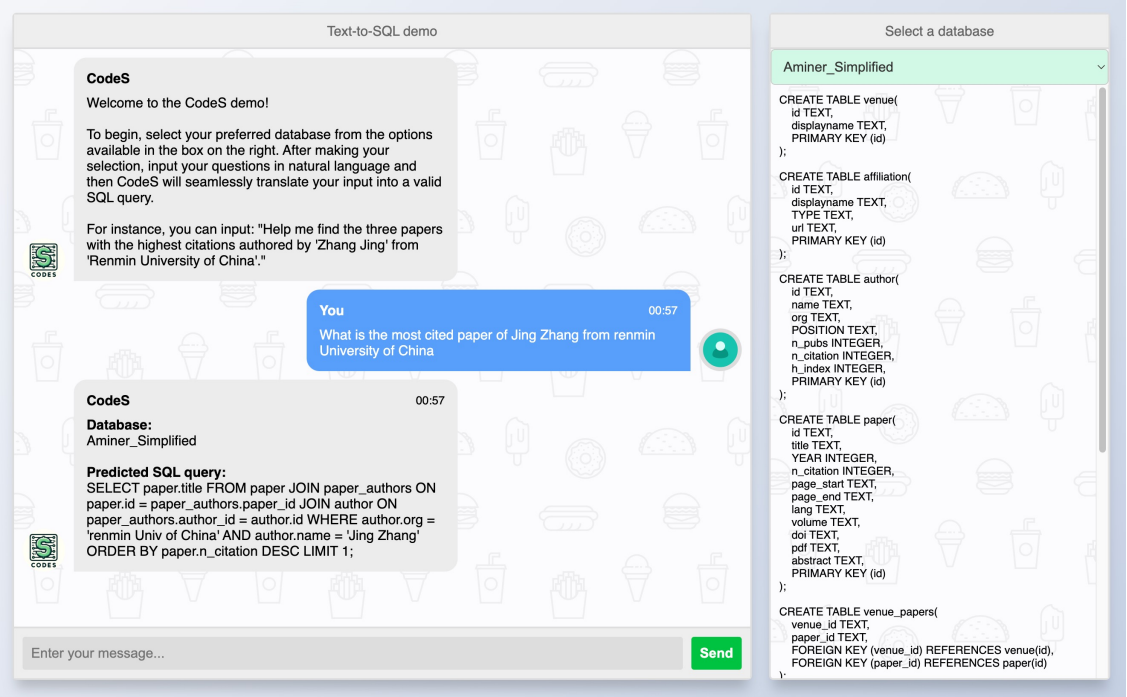
Contributions

- We have taken a significant strides toward enhancing the landscape of open-source text-to-SQL models.
- With the introduction of CodeS, developers now have access to a range of specialized pre-trained language models to develop their text-to-SQL applications.
- We also propose a **comprehensive database prompt construction strategy and a novel bi-directional data augmentation method**.

Github Link:
https://github.com/RUCKBReasoning/codes

Demo Link:
https://github.com/RUCKBReasoning/text2sql-demo

Paper Link:
https://arxiv.org/abs/2402.16347



LLM for Addressing Structured Data

Direct LLM Infer:

- Hallucination
- knowledge update challenge

RAG LLM:

- Inability to address complex logic

Rigorous logical language:

- Handles Structured Data Effectively

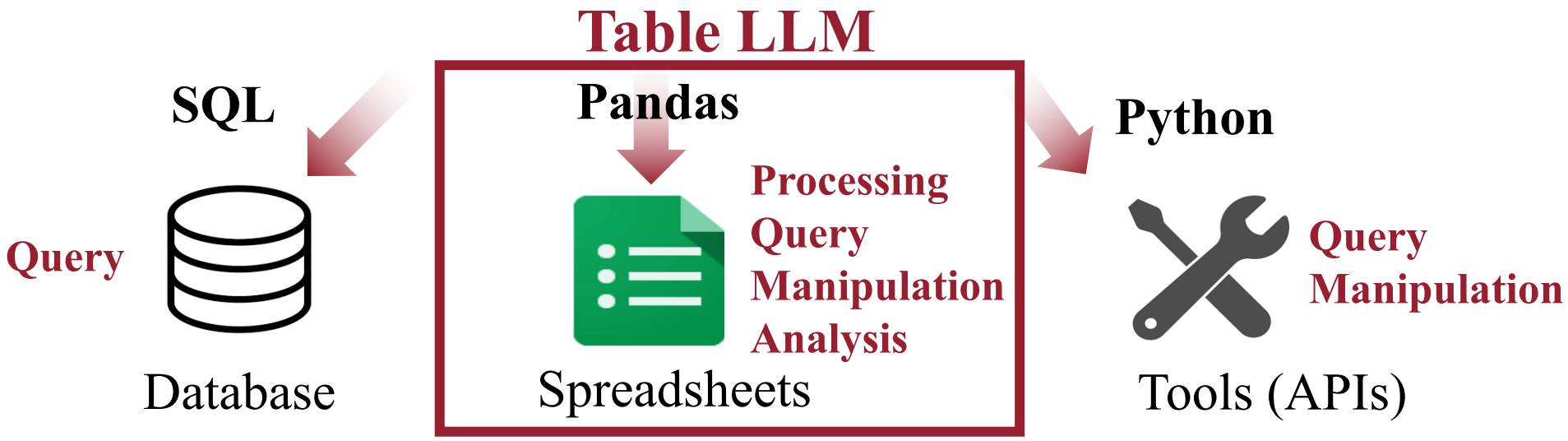


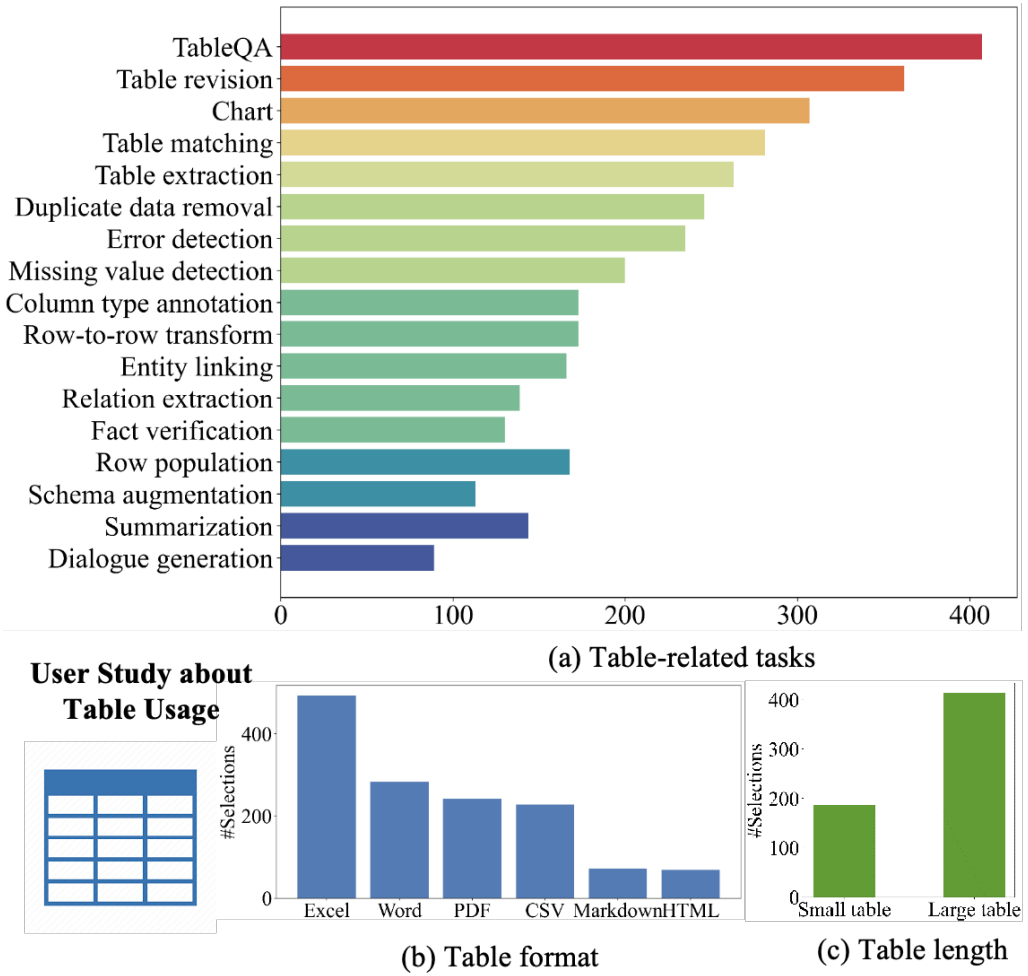
Table Task

Real-world office use of tabular data challenges:

(1) **Diverse Operations**: user preferred tasks involve a wide range of operations, including query, update, merge, and chart besides tableQA.

(2) **Unique Processing Approaches for Different Formats**:

- **Word/PDF documents:** textual data alongside tabular information, allowing for hybrid querying.
- **Excel/CSV spreadsheets:** regular and long tables, enabling intricate operations like update and merge.



An extensive user study related to tables

Xiaokang Zhang, Jing Zhang, Zeyao Ma, etc. https://arxiv.org/pdf/2403.19318

SFT

- Three main distinct methodologies for obtaining SFT data:
  - Crowdsourced Collection: This method involves leveraging the collective intelligence of a large group of individuals, typically sourced from online platforms or communities, to gather diverse SFT data through various tasks or interactions.
  - Modification of Existing Benchmark: This approach entails adapting and enhancing pre-existing datasets to better suit the requirements of SFT training. It involves techniques such as data augmentation, annotation, or domain-specific tailoring to enrich the dataset's diversity and relevance.
  - LLM Self-Iteration: Here, we capitalize on the ability of LLMs like ChatGPT to continuously improve themselves through self-training iterations.

SFT TableLLM

We augment existing benchmarks by enriching their reasoning processes to facilitate the training of LLMs.

ID	Employer	#Employees
1	Medline	1,200
2	MPD	422
3	Amcor	350
4	FSD 79	287
5	Univ. of SML	220
6	ME School D.	213
7	M. High School	211
8	Village of M	183

Benchmark

Question: how many employers have at least 300 employees?

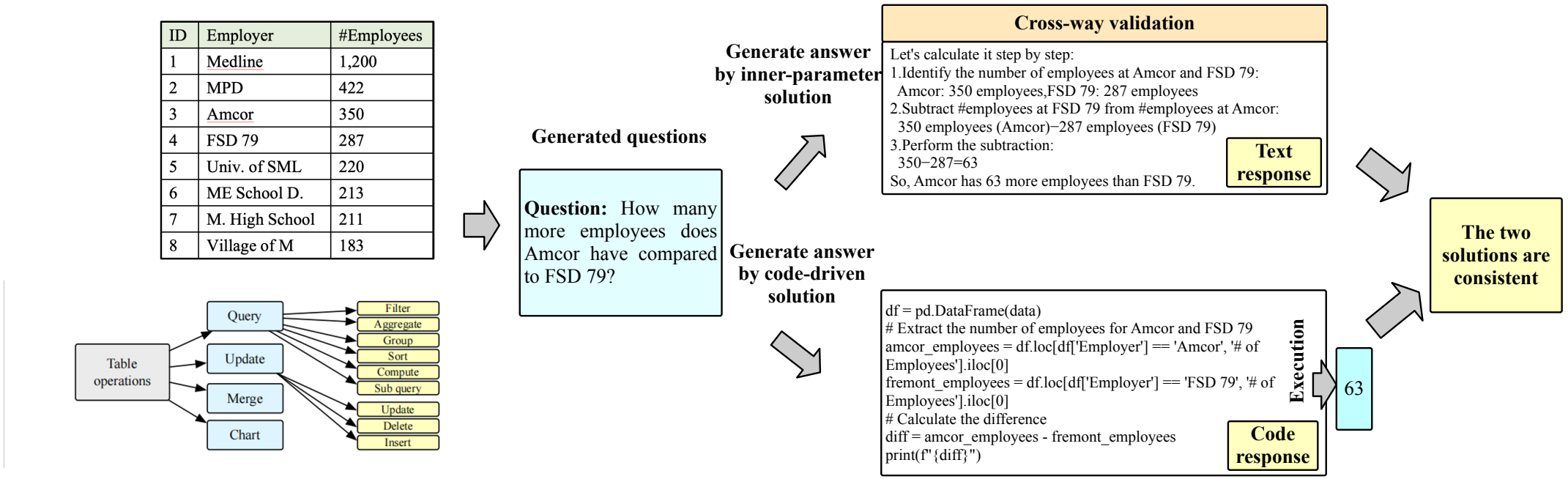
Answer: 3

Reasoning process extending

Let's go through each row of the table:
1.Medline: #employees is 1,200, which is greater than 300.
2.MPD: #employees is 422, which is greater than 300.
3.Amcor: #employees is 350, which is greater than 300.
4.FSD 79: #employees is 287, which is less than 300.
5.Univ. of SML: #employees is 220, which is less than 300.
6.ME School D.: #employees is 213, which is less than 300.
7.M. High School: #employees is 211, which is less than 300.
8.Village of M: #employees is 183, which is less than 300.
Based on this analysis, we can conclude that there are 3 employers with at least 300 employees: Medline, MPD, and Amcor.

SFT TableLLM

To increase the diversity and coverage of the training data, we provide table atom operator list and use a cross-validation strategy to automatically generate new questions and answers from the provided tabular data



Theorem：

(1) If A and B are drawn from the same distribution such that $P(Y_a) = P(Y_b) = p > ½$, then consistency checking outperforms single inference, i.e., $P(Y|E) >= P(Y_a)$.

(2) If A and B are further drawn from independent distributions, the effect will be superior (in terms of expectation)

Experiments (Supervised Fine-tuning)

TableLLM generally surpasses others in the spreadsheet-embedded scenario and is on par with GPT-3.5 in the document-embedded scenario.

Table 2: Overall evaluation in both document-embedded and spreadsheet-embedded tabular data scenarios (%)

Model	Document-embedded tabular data				Spreadsheet-embedded tabular data			Average accuracy	Inference times
	WikiTQ	TAT-QA	FeTaQA	OTT-QA	WikiSQL	Spider	Our created		
TaPEX	38.55	–	–	–	83.90	15.04	–	45.83	1
TaPas	31.60	–	–	–	74.20	23.05	–	42.95	1
TableLlama	24.01	22.25	20.47	6.39	43.70	–	–	23.36	1
Llama2-Chat (13B)	48.82	49.63	67.73	61.50	–	–	–	56.92	1
GPT-3.5	58.45	72.13	71.18	60.80	81.70	67.38	77.08	69.82	1
GPT-4	**74.09**	**77.13**	**78.35**	**69.50**	84.00	69.53	77.83	**75.78**	1
CodeLlama (13B)	43.44	47.25	57.24	49.72	38.30	21.88	47.58	43.63	1
Deepseek-Coder (33B)	6.48	11.00	7.12	7.44	72.50	58.40	73.92	33.84	1
StructGPT (GPT-3.5)	52.45	27.53	11.80	13.96	67.80	**84.80**	–	43.06	3
Binder (GPT-3.5)	61.61	12.77	6.85	5.13	78.60	52.55	–	36.25	50
DATER (GPT-3.5)	53.40	28.45	18.26	13.03	58.20	26.52	–	32.98	100
TableLLM (7B)	58.77	66.88	72.64	63.11	86.60	82.62	78.83	72.68	1
TableLLM (13B)	62.40	68.25	74.50	62.51	**90.70**	83.40	**80.83**	74.66	1

* Underline represents the runner up.

Experiments (Ablation Studies)

We conduct extensive ablation studies to analyze the influence of different training datasets and demonstrate the effectiveness proposed cross-way validation method.

Table 3: Effect of diverse training data sources (%)

Train data	Document-embedded		Spreadsheet-embedded	
	WikiTQ	TAT-QA	Spider	Our created
CodeLlama (13B)	43.4	47.3	21.9	47.6
Original train data	49.9	53.4	–	–
Extended train data	53.7	62.6	82.0	52.2
Generated train data	51.5	59.8	63.7	80.1
Mixed data	54.7	63.5	84.2	80.9

Table 4: Effect of cross-way validation strategy (%)

Validation strategy	WikiTQ	TAT-QA
Self-check validation	49.4	55.8
Same-way validation	49.6	58.2
Cross-way validation	51.5	59.8

Conclusion

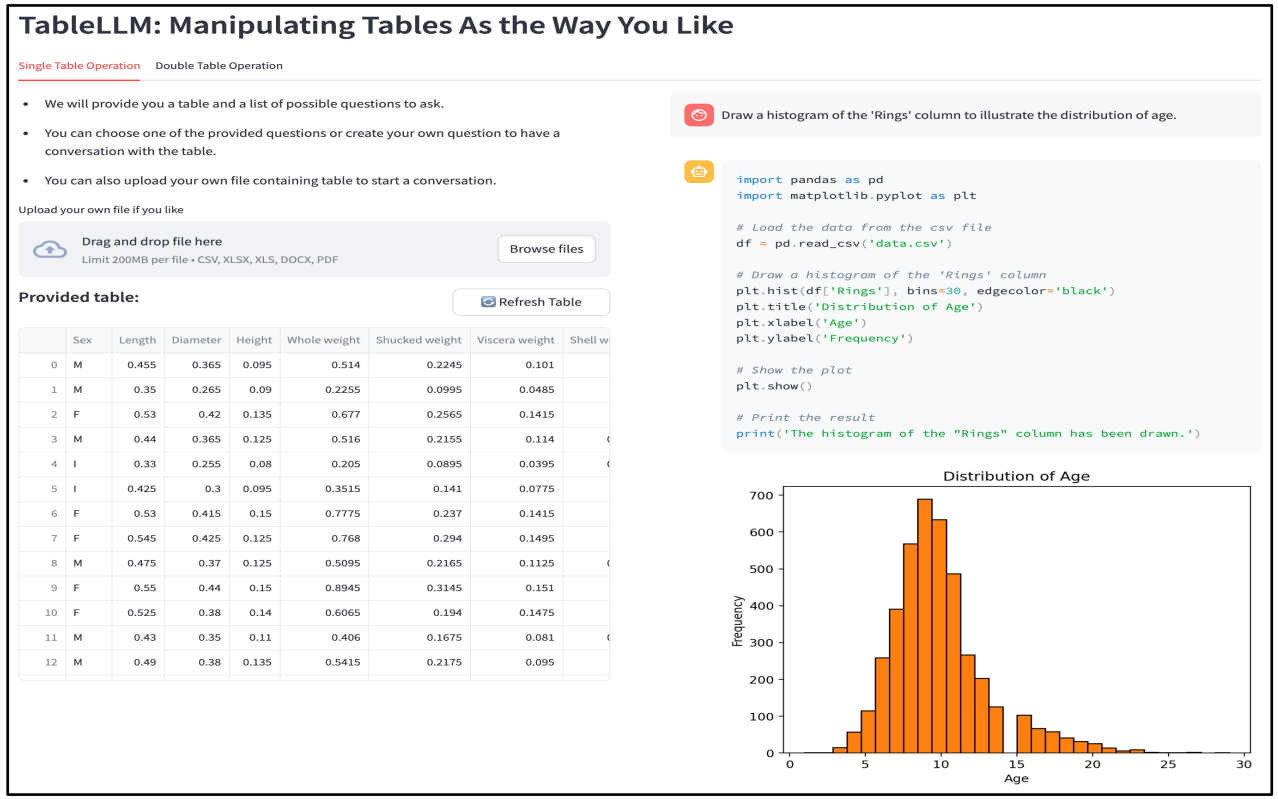
Contributions

- We address a practical problem of tabular data manipulation in real-world office usage scenarios.
- We present techniques that extend reasoning processing and integrate a cross-way validation strategy to enhance the quality of distant supervision training data.
- We deliver a high-quality opensource LLM tailored for tabular data manipulation in both 7B and 13B, thereby enhancing accessibility and fostering collaboration within the community.

Github Link:
https://github.com/TableLLM/TableLLM

Paper Link:
https://arxiv.org/abs/2403.19318

Platform Link:
https://tablellm.github.io/



LLM for Addressing Structured Data

Direct LLM Infer:

- Hallucination
- knowledge update challenge

RAG LLM:

- Inability to address complex logic

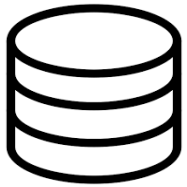
Rigorous logical language:

- Handles Structured Data Effectively

Tool LLM

SQL

Query



Database

Pandas

Processing
Query
Manipulation
Analysis

Spreadsheets

Python

Query
Manipulation

Tools (APIs)

× AMiner

SoAy: A Solution-based LLM API-using Methodology for Academic Information Seeking

Yuanchun Wang[†*], Jifan Yu[§*], Zijun Yao[§], Jing Zhang[†], Yuyang Xie[§], Shangqing Tu[§]
Yiyang Fu[†], Youhe Feng[†], Jinkai Zhang[†], Jingyao Zhang[◇], Bowen Huang[◇], Yuanyao Li[◇]
Huihui Yuan[◇], Lei Hou[§], Juanzi Li[§] and **Jie Tang[§]**
[†]Renmin University of China [§]Tsinghua University [◇]Zhipu AI

[Paper] https://arxiv.org/pdf/2405.15165

[Code] https://github.com/RUCKBReasoning/SoAy

[System] https://soay.aminer.cn/

[Model] https://huggingface.co/frederickwang99/soayllama_v2_7b

[Benchmark & Dataset] https://huggingface.co/datasets/frederickwang99/SoAyBench

LLM Tool-using Process

1. Task Planning

- Tuning-free Methods
- Tuning-based Methods

2. Tool Selection

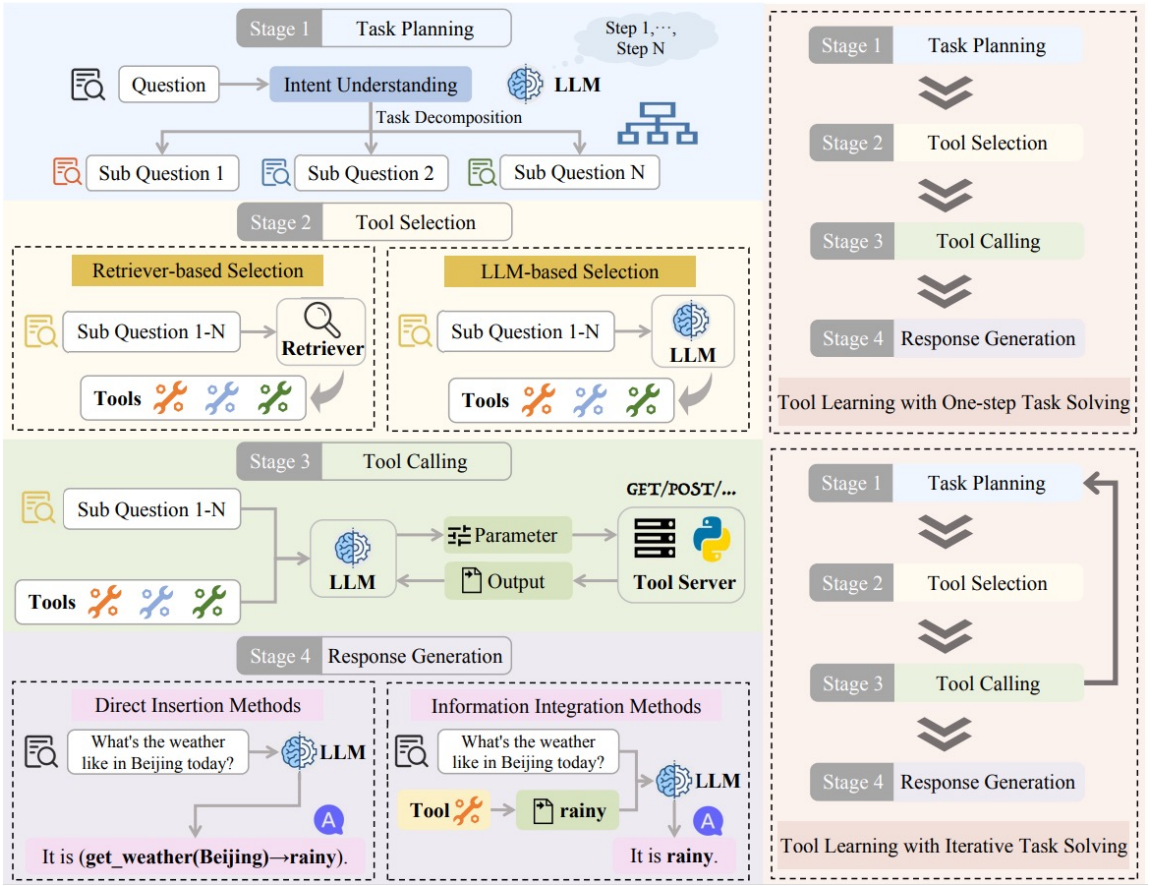
- Retriever-based Tool Selection
- LLM-based Tool Selection

3. Tool Calling

- Tuning-free Methods
- Tuning-based Methods

4. Response Generation

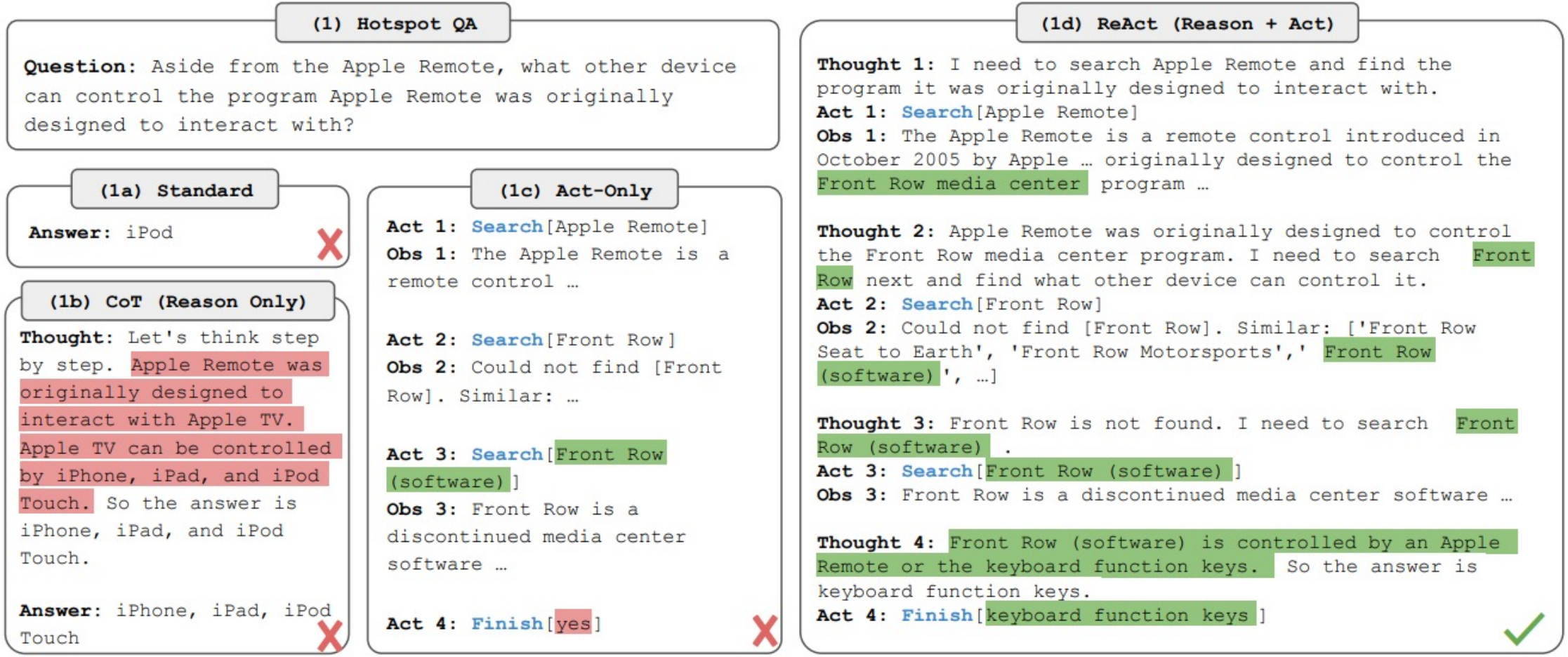
- Direct Insertion Methods
- Information Integration Methods



Core steps in tool learning.

Tool Learning with Large Language Models: A Survey. Changle Qu, etc. Arxiv.

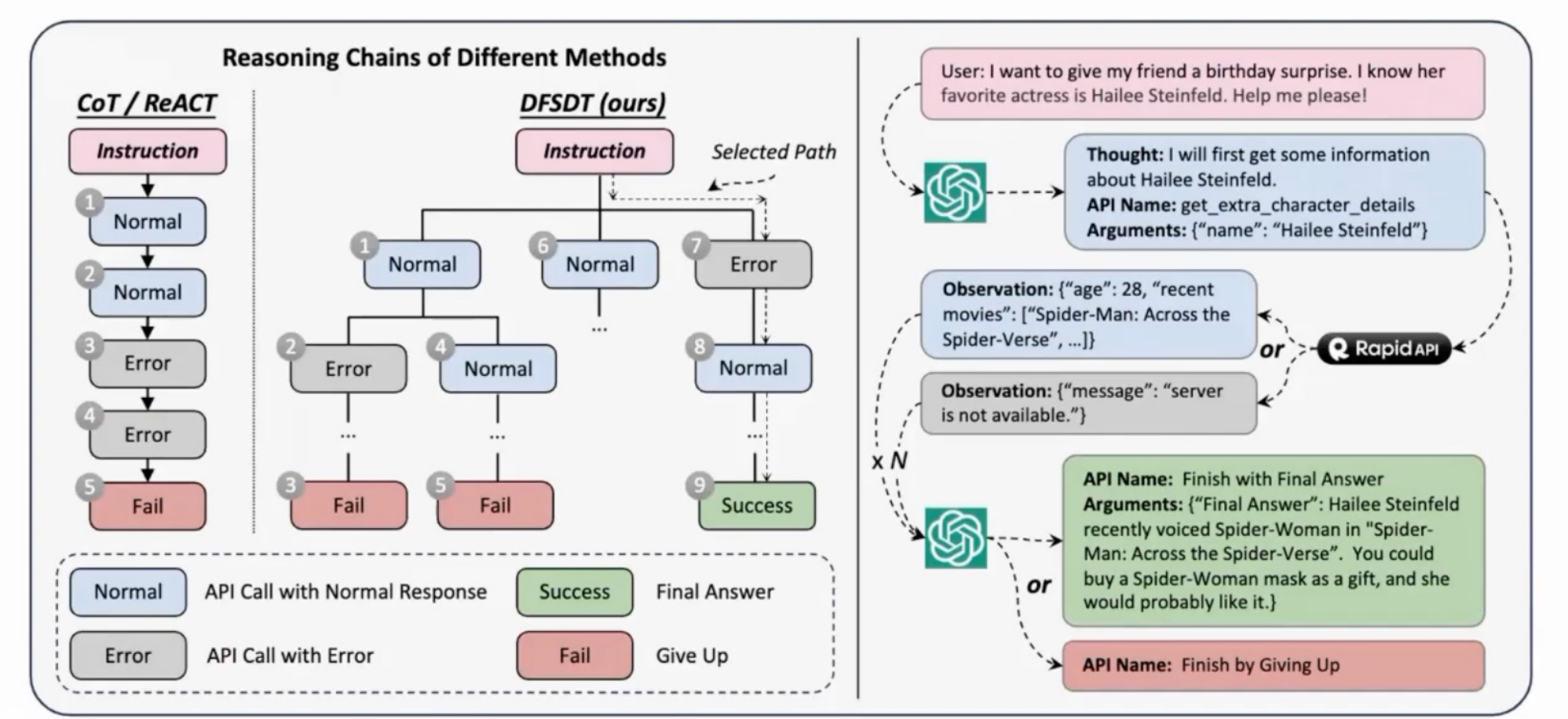
LLM Tool-using Planning



ReAct planning on a Hotspot QA case.

ReAct: Synergizing Reasoning and Acting in Language Models. Shunyu Yao, etc. ICLR'23.

LLM Tool-using Planning



DFSDT planning on a ToolBench case.

ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. Yujia Qin, etc. ICLR'24 spotlight.

Academic Information Systems API Calling

Query: How many times has New York University's Yann LeCun's most cited publication been cited?

ID	API name	Type	Parameter(s)	Return
1	searchPerson	fuzzy	name, organization, interest	[person_id, name, num_citation, interest, num_pubs, organization]
2	searchPublication	fuzzy	publication_info	[pub_id, title, year]
3	getCoauthors	exact	person_id	[id, name, relation]
4	getPersonInterest	exact	person_id	list of interests
5	getPublication	exact	pub_id	abstract, author_list, num_citation
6	getPersonBasicInfo	exact	pub_id	person_id, name, gender, organization, position, bio, education_experience, email
7	getPersonPubs	exact	person_id	[authors_name_list, pub_id, title, num_citation, year]

Yann LeCun, NYU ➤ searchPerson ➤ [{ person_id: ec0f***jsk, person_name: Yann LeCun, ... }]

ec0f***jsk ➤ getPersonPubs ➤ [{ pub_id : al4k***8fa, ...}, { pub_id : 79pa***rjk, ...}, { pub_id : q2f4***n3c, ...}...]

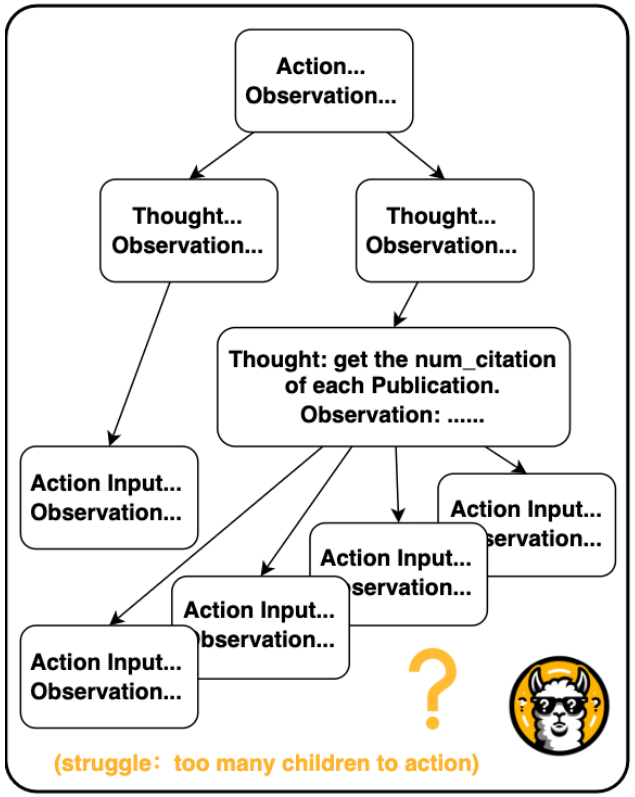
al4k***8fa
79pa***rjk
q2f4***n3c
➤ getPublication ➤ [{ title: Efficient Backprop, citation: 7145}, { title: Deeplearning, citation: 79904}, { title: The Minist Database, citation: 7592}...]

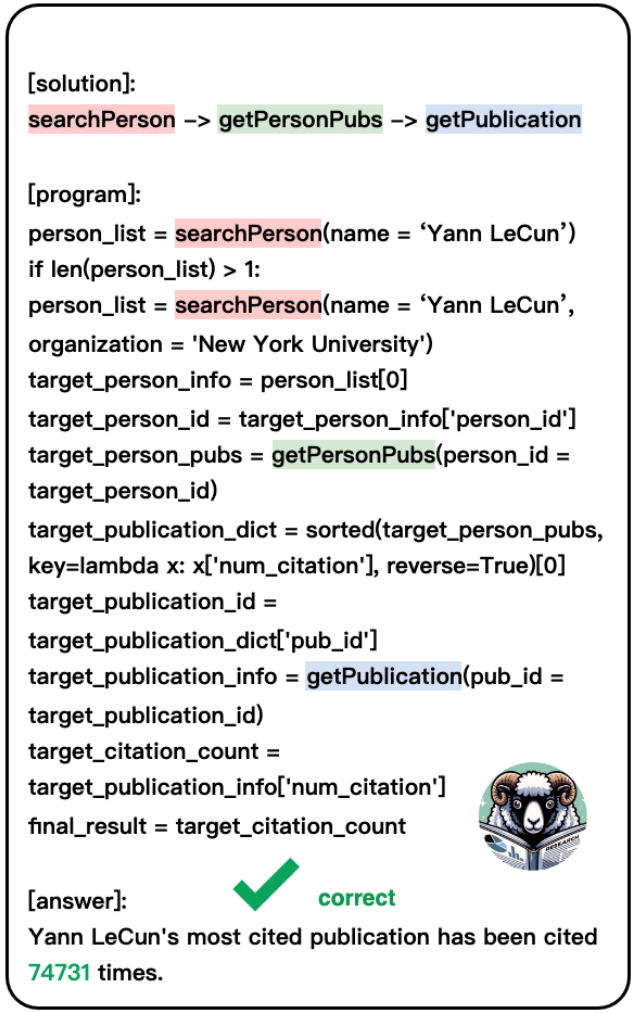
LLM API-Using

Query: How many times has New York University's Yann LeCun's most cited publication been cited?

Retrieval & Execution:
Failed to handle
API Coupling

DFSDT Reasoning:
Could not meet
the Efficiency needs





SoAy:

Pre-defined Solution
&
Solution-based Program
Generation

Different API-using sturctures facing the same academic question.

SoAy: SoAPIs Applying Framework

Query: How many times has New York University's Yann LeCun's most cited publication been cited?

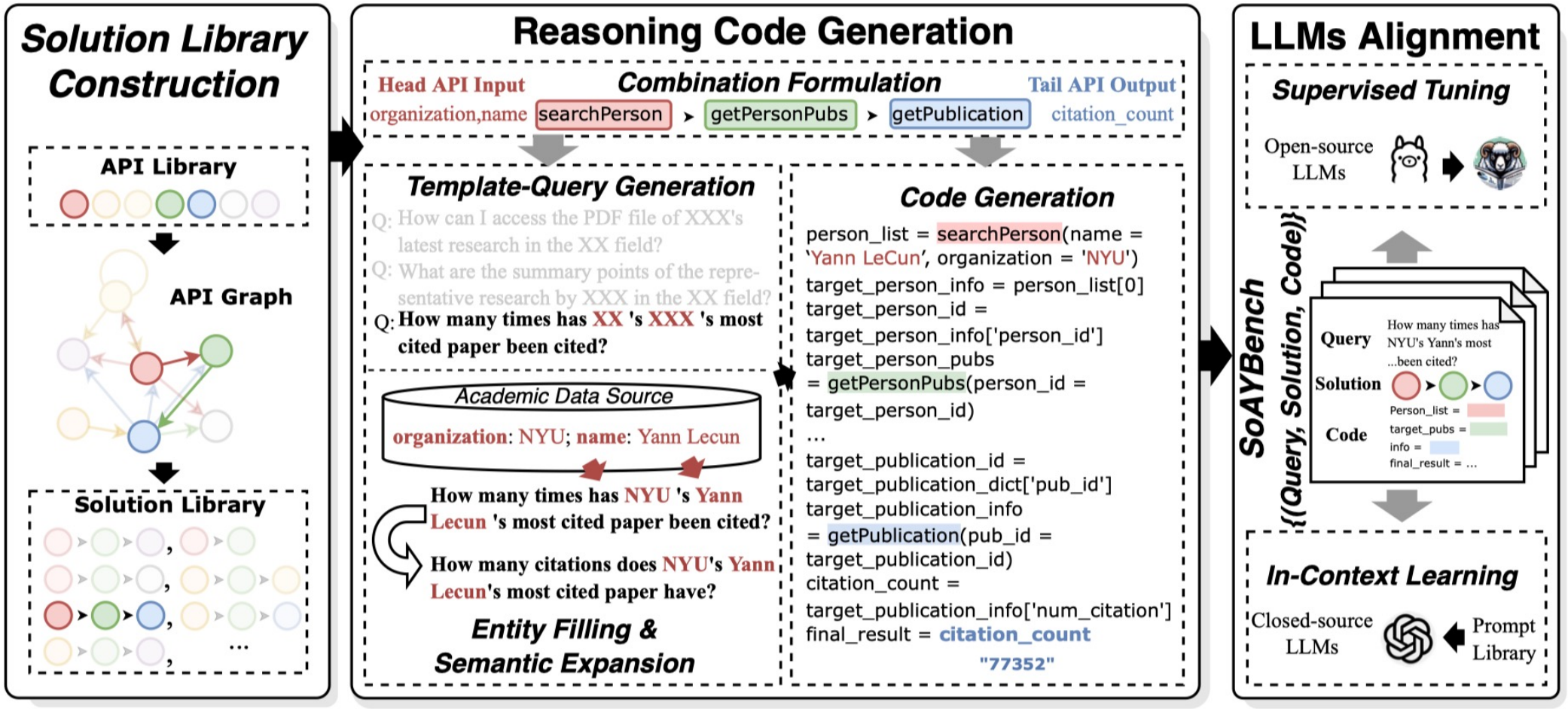


Fig.13 SoAy Framework.

SoAyBench

To assess API utilization capabilities, it is essential to publish the foundational APIs of AMiner for LLMs to invoke and provide a test set composed of academic {Query, Code, Answer} triplets for evaluation.

However, given the dynamic nature of academic data, with scholar and publication information rapidly changing, maintaining a test set with static answers proves challenging.

To address this challenge, **we clone AMiner's SoAPIs at a specific point in time** to create a static service, from which we generate a corresponding static test set.

SoAyBench now are open-sourced at : 🤗 Hugging Face:
https://huggingface.co/datasets/frederickwang99/SoAyBench

Question statistics in SoAyBench.

Question Type	One-hop	Two-hop	Three-hop	**Total**
Scholar	540	1,800	540	2,980
Publication	180	180	720	1,080
Total	720	1,980	1,230	**3,960**

Results on SoAyBench - Part I

Results on SoAyBench. DS, WS, WP and EE are differenct types of error, ACC denotes a accurate answer, EM means exact match, not only the answer but also the solution. Score is a weighted sum of the ACC score on different question types.

Method	Version	Question Type	Error Rate↓				EM(%)	ACC(%)	Score
			DS(%)	WS(%)	WP(%)	EE(%)			
ToolLLaMA	7B	one-hop	12.50±8.00	24.31±13.26	1.39±0.00	54.17±16.01	7.64±5.20	20.14	
		two-hop	10.10±4.10	47.22±12.28	0.76±2.27	38.13±9.62	3.79±2.92	13.89	16.72
		three-hop	11.51±6.53	38.10±14.27	1.19±3.57	43.25±13.07	5.95±4.59	17.46	
GPT-DFSDT	3.5	one-hop	55.56±21.06	15.28±7.80	4.86±0.00	21.53±10.67	2.78±0.00	58.33	
		two-hop	29.55±11.47	34.34±9.23	4.29±3.64	25.76±8.65	6.06±4.11	35.61	43.22
		three-hop	38.10±15.09	28.57±11.35	3.17±2.50	25.00±8.87	5.16±6.19	43.25	
	3.5-16k	one-hop	25.69±10.91	9.72±5.00	2.78±0.00	22.92±9.47	38.89±15.60	64.58	
		two-hop	16.92±7.76	15.91±6.05	3.28±1.31	46.97±7.13	16.92±4.99	33.84	43.67
		three-hop	18.65±7.37	15.48±5.63	2.78±0.00	38.49±10.43	24.60±8.53	43.25	
	4	one-hop	27.78±9.60	2.08±0.00	4.17±5.00	28.47±6.82	37.50±10.91	65.28	
		two-hop	26.26±8.89	9.60±4.88	17.93±5.40	15.15±5.39	31.06±9.12	57.32	58.16
		three-hop	22.22±8.65	7.54±4.46	17.06±6.96	19.05±6.45	34.13±9.87	56.35	
GPT-SoAY	3.5	one-hop	27.78±8.70	15.97±7.73	3.47±0.00	13.19±7.80	39.58±9.12	67.36	
		two-hop	33.84±4.94	9.60±4.75	6.06±2.81	13.13±7.12	37.37±5.06	71.21	67.30
		three-hop	22.22±6.43	12.70±5.91	9.52±4.42	13.10±6.72	42.46±6.00	64.68	
	3.5-16k	one-hop	28.47±11.67	15.28±6.12	1.39±0.00	17.36±7.78	37.50±9.07	65.97	
		two-hop	35.86±6.01	7.32±3.41	5.30±2.18	15.91±7.16	35.61±4.65	71.46	66.76
		three-hop	23.02±7.16	10.32±4.99	8.33±3.42	17.46±7.37	40.87±6.26	63.89	
	4	one-hop	0.00±0.00	0.00±0.00	1.39±0.00	2.78±0.00	95.83±5.70	95.83	
		two-hop	15.91±4.71	1.26±0.00	9.34±1.07	2.02±1.69	71.46±3.74	87.37	86.57
		three-hop	6.75±0.00	0.40±0.00	14.68±1.68	1.98±0.00	76.19±3.25	82.94	

Results on SoAyBench - Part II

Results on SoAyBench. DS, WS, WP and EE are differenct types of error, ACC denotes a accurate answer, EM means exact match, not only the answer but also the solution. Score is a weighted sum of the ACC score on different question types.

GPT-SoAY	3.5	one-hop	27.78±8.70	15.97±7.73	3.47±0.00	13.19±7.80	39.58±9.12	67.36	
		two-hop	33.84±4.94	9.60±4.75	6.06±2.81	13.13±7.12	37.37±5.06	71.21	67.30
		three-hop	22.22±6.43	12.70±5.91	9.52±4.42	13.10±6.72	42.46±6.00	64.68	
	3.5-16k	one-hop	28.47±11.67	15.28±6.12	1.39±0.00	17.36±7.78	37.50±9.07	65.97	
		two-hop	35.86±6.01	7.32±3.41	5.30±2.18	15.91±7.16	35.61±4.65	71.46	66.76
		three-hop	23.02±7.16	10.32±4.99	8.33±3.42	17.46±7.37	40.87±6.26	63.89	
	4	one-hop	0.00±0.00	0.00±0.00	1.39±0.00	2.78±0.00	95.83±5.70	95.83	
		two-hop	15.91±4.71	1.26±0.00	9.34±1.07	2.02±1.69	71.46±3.74	87.37	86.57
		three-hop	6.75±0.00	0.40±0.00	14.68±1.68	1.98±0.00	76.19±3.25	82.94	
SoAYLLaMA	Chat-7B	one-hop	0.00±0.00	0.00±0.00	0.00±0.00	0.69±0.00	99.31±2.94	99.31	
		two-hop	0.00±0.00	0.00±0.00	20.20±3.84	2.53±1.97	77.27±2.70	77.27	85.76
		three-hop	0.00±0.00	0.00±0.00	9.92±3.56	3.17±2.50	86.90±2.72	86.90	
	Code-7B	one-hop	0.69±0.00	0.00±0.00	0.69±0.00	5.56±4.37	93.06±7.50	93.75	
		two-hop	0.25±0.00	3.28±0.00	7.07±2.75	4.80±3.69	84.60±5.18	84.85	88.95
		three-hop	0.40±0.00	0.00±0.00	4.76±2.14	5.16±4.57	89.68±6.54	90.08	
	Code-13B	one-hop	0.00±0.00	0.00±0.00	1.39±0.00	0.00±0.00	98.61±4.03	98.61	
		two-hop	0.00±0.00	2.27±0.00	14.14±2.14	0.51±0.00	83.08±3.32	83.08	**92.74**
		three-hop	0.00±0.00	0.00±0.00	2.38±2.86	0.40±0.00	97.22±4.28	97.22	

Results on SoAyBench - Efficiency

A Small Model trained on Data generated by GPT-4 can outperform GPT-4, and also more efficient.

SoAyGPT	3.5	one-hop	27.78±8.70	15.97±7.73	3.47±0.00	13.19±7.80	39.58±9.12	67.36	67.30
		two-hop	33.84±4.94	9.60±4.75	6.06±2.81	13.13±7.12	37.37±5.06	71.21	
		three-hop	22.22±6.43	12.70±5.91	9.52±4.42	13.10±6.72	42.46±6.00	64.68	
	3.5-16k	one-hop	28.47±11.67	15.28±6.12	1.39±0.00	17.36±7.78	37.50±9.07	65.97	66.76
		two-hop	35.86±6.01	7.32±3.41	5.30±2.18	15.91±7.16	35.61±4.65	71.46	
		three-hop	23.02±7.16	10.32±4.99	8.33±3.42	17.46±7.37	40.87±6.26	63.89	
	4	one-hop	0.00±0.00	0.00±0.00	1.39±0.00	2.78±0.00	95.83±5.70	95.83	86.57
		two-hop	15.91±4.71	1.26±0.00	9.34±1.07	2.02±1.69	71.46±3.74	87.37	
		three-hop	6.75±0.00	0.40±0.00	14.68±1.68	1.98±0.00	76.19±3.25	82.94	
SoAyLLaMA	Chat-7B	one-hop	0.00±0.00	0.00±0.00	0.00±0.00	0.69±0.00	99.31±2.94	99.31	85.76
		two-hop	0.00±0.00	0.00±0.00	20.20±3.84	2.53±1.97	77.27±2.70	77.27	
		three-hop	0.00±0.00	0.00±0.00	9.92±3.56	3.17±2.50	86.90±2.72	86.90	
	Code-7B	one-hop	0.69±0.00	0.00±0.00	0.69±0.00	5.56±4.37	93.06±7.50	93.75	88.95
		two-hop	0.25±0.00	3.28±0.00	7.07±2.75	4.80±3.69	84.60±5.18	84.85	
		three-hop	0.40±0.00	0.00±0.00	4.76±2.14	5.16±4.57	89.68±6.54	90.08	
	Code-13B	one-hop	0.00±0.00	0.00±0.00	1.39±0.00	0.00±0.00	98.61±4.03	98.61	**92.74**
		two-hop	0.00±0.00	2.27±0.00	14.14±2.14	0.51±0.00	83.08±3.32	83.08	
		three-hop	0.00±0.00	0.00±0.00	2.38±2.86	0.40±0.00	97.22±4.28	97.22	

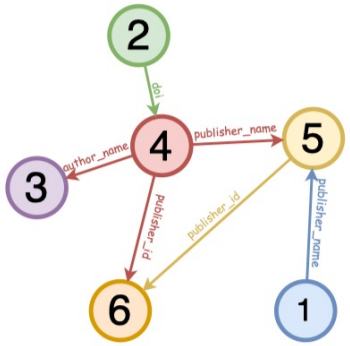
Method	7B	13B	3.5	3.5-16k	4
ToolLLaMA	45.10	/	/	/	/
GPT-DFSDT	/	/	39.12	53.73	70.92
SoAyGPT	/	/	6.15	6.40	26.05
SoAyLLaMA-Code	1.12	1.35	/	/	/

Deployment on other Academic Platforms

AMiner APIs are NOT the only that face the coupling challenges.
We also deployment SoAy on two other open-sourced scenarios: OpenLibrary and CrossRef

ID	API name	Type	Parameter(s)	Return
1	searchPublisherBySubject	fuzzy	subject	[publisher_name, doi_count]
2	searchWorksByTitle	fuzzy	work_title	[type, author, doi, publisher]
3	searchWorksByAuthor	fuzzy	author_name	[works_title, works_doi]
4	getWorksByDoi	exact	doi	[author_name, work_title, publisher_name, type, reference_count]
5	getPublisherBasicInfo	exact	publisher_name	[publisher_id, current_dois, backfile_dois, total_dois, doi_prefix]
6	getPublisherWorks	exact	publisher_id	[works_title, doi, works_author]

(a) CrossrefAPI Library



(b) CrossrefAPI Graph

Solution	Parameter(s)	Return	Question Template
searchPublisherBySubject	subject	publisher_name	Please list some publishers in the XXX field.
searchPublisherBySubject → getPublisherBasicInfo	subject	publisher_id	Please give me some publishers' id of crossref about the field of XXX.
seachPublisherBySubject → getPublisherBasicInfo → getPublisherWorks	subject	doi	Can you list some articles' DOI numbers in the field of XXX?
searchWorksByTitle	work_title	type	I want to know the type of work XXX.

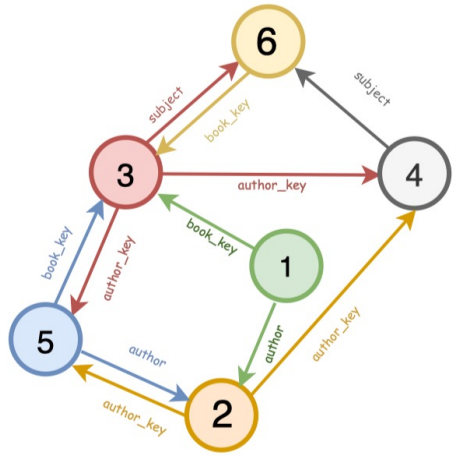
(c) Solution Library (partly shown)

ID	API name	Type	Parameter(s)	Return
1	searchBook	fuzzy	book_info	[book_key, title, author_name, year]
2	searchAuthor	fuzzy	author_info	[author_key, name, list of alternate_names]
3	getBook	exact	book_key	description, list of author, title, first_publish, list of subjects
4	getAuthorBasicInfo	exact	author_key	name, list of alternate_names, birth_date, work_count, top_work, top_subjects
5	getAuthorWorks	exact	author_key, amount	[book_key, title, subjects]
6	searchSubject	fuzzy	subject	[book_key, title]

(a) SoAPI Library

Solution	Parameter(s)	Return	Question Template
searchSubject	subject	list of books	Please list some books on XXX topic.
searchAuthor→getAuthorWorks	author_info	list of books	Which works were written by XXX?
searchBook→getBook	book_info	book_description	Introduce some information about XXX.
searchBook→getBook→getAuthorWorks	book_info	list of books	What other books has the author of XXX written?

(c) Solution Library (partly shown)



(b) SoAPI Graph

Challenges of the SoAyBench & SoAyEval

There's still some challenges on the evaluation part of specific-domain tool using.

- The benchmark or evaluation set is limited on the Academic domain.
- The complexity of testing on the combination of the **LLMs**, Tool-using **Workflows** and the **domains**.

Method	7B	13B	3.5	3.5-16k	4
ToolLLaMA	45.10	/	/	/	/
GPT-DFSDT	/	/	39.12	53.73	70.92
SoAyGPT	/	/	6.15	6.40	26.05
SoAyLLaMA-Code	1.12	1.35	/	/	/

R-Eval: A Unified Toolkit for Evaluating Domain Knowledge of Retrieval Augmented Large Language Models

Shangqing Tu*
DCST, Tsinghua Univerisity
Beijing 100084, China
tsq22@mails.tsinghua.edu.cn

Yuanchun Wang*
SoI, Renmin University of China
Beijing 100084, China
wangyuanchun@ruc.edu.cn

Jifan Yu
DCST, Tsinghua Univerisity
Beijing 100084, China
yujf21@mails.tsinghua.edu.cn

Yuyang Xie
DCST, Tsinghua Univerisity
Beijing 100084, China
xieyy21@mails.tsinghua.edu.cn

Yaran Shi
SIOE, Beihang Univerisity
Beijing 100084, China
syr2021@buaa.edu.cn

Xiaozhi Wang
DCST, Tsinghua Univerisity
Beijing 100084, China
wangxz20@mails.tsinghua.edu.cn

Jing Zhang
SoI, Renmin University of China
Beijing 100084, China
zhang-jing@ruc.edu.cn

Lei Hou
BNRist, DCST, Tsinghua Univerisity
Beijing 100084, China
houlei@tsinghua.edu.cn

Juanzi Li
BNRist, DCST, Tsinghua Univerisity
Beijing 100084, China
lijuanzi@tsinghua.edu.cn

[Paper] Accepted by KDD'24 (ADS track)

[Code & Toolkit] https://github.com/THU-KEG/R-Eval

Background - Component Selection

Given a Specific Domain, which LLM and which RAG Workflow to choose?

Shortcomings of exsisting evaluations:

- Insufficient exploration of combinations between LLMs and RAG workflows.
- Lack comprehensive mining of the domain knowledge.

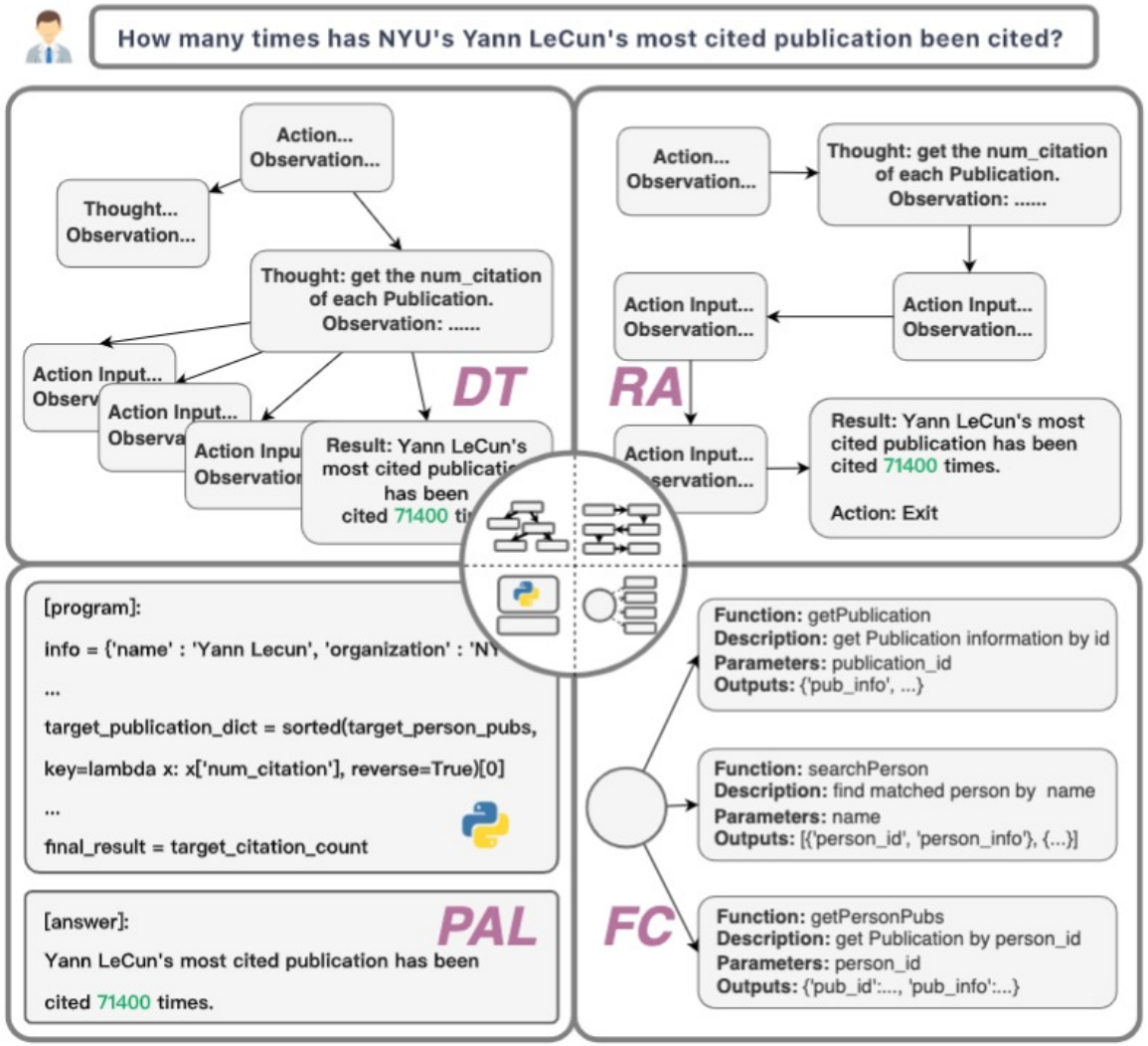
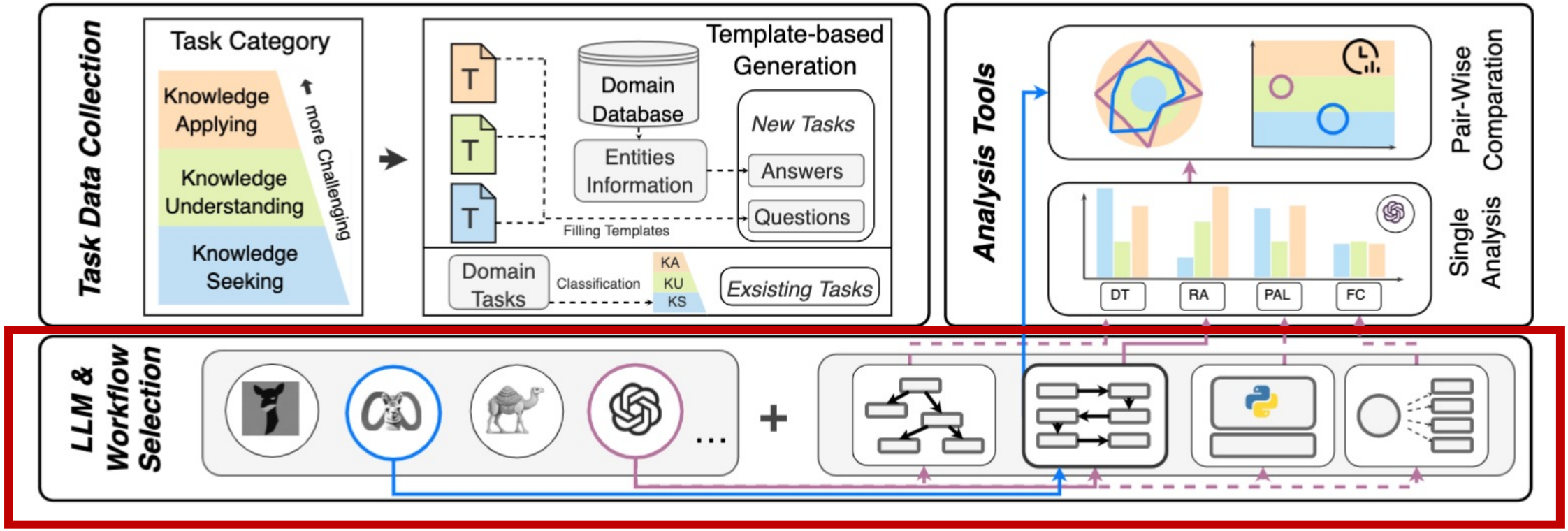


Fig.16 Four Popular RAG Workflows.

Evaluation Framework

We propose R-Eval, a Python toolkit designed to streamline the evaluation of different RAG workflows in conjunction with LLMs on a specific domain's task.

- **A easy-to-use evaluation of the combination between RAG Workflows and LLMs**
- Customized testing data in specific domains through template-based question generation



Evaluation Result - Performance Ranking

The same Workflow + LLM,

Same Domain Task

Differenct Level

Workflow	LLM	aminer KS		aminer KU		aminer KA		Overall Average (Level 1, 2, 3)					
		1-3	Rank	2-4	Rank	3-5	Rank	wiki	Rank	aminer	Rank	all	Rank
ReAct	gpt-4-1106	89.7	1st	46.7	3rd	57.7	1st	38.8	1st	64.7	1st	45.3	1st
PAL	gpt-3.5-turbo	80.1	3rd	50.7	2nd	54.9	2nd	19.9	6th	61.9	2nd	30.4	2nd
PAL	gpt-4-1106	59.3	4th	56.8	1st	52.7	3rd	20.3	5th	56.2	3rd	29.2	3rd
ReAct	llama2-7b-chat	45.2	5th	36.5	6th	21.5	6th	23.8	3rd	34.4	5th	26.4	4th
PAL	llama2-13b	25.3	6th	36.4	7th	20.3	7th	25.2	2nd	27.3	6th	25.7	5th
ReAct	gpt-3.5-turbo	84.6	2nd	4.0	14th	33.0	4th	19.6	7th	40.6	4th	24.9	6th
ReAct	vicuna-13b	19.9	10th	6.0	13th	7.1	16th	20.7	4th	11.0	17th	18.2	7th
PAL	tulu-7b	9.1	15th	26.8	9th	11.5	12th	18.9	8th	15.8	9th	18.1	8th
PAL	vicuna-13b	4.5	17th	40.9	4th	2.3	20th	16.7	9th	15.9	8th	16.5	9th
ReAct	llama2-13b	16.7	13th	0.7	19th	23.2	5th	15.0	10th	13.5	12th	14.6	10th
PAL	llama2-7b-chat	18.7	12th	2.8	15th	16.1	8th	12.4	11th	12.5	14th	12.4	11th
PAL	codellama-13b	4.4	18th	38.3	5th	8.1	14th	10.0	14th	16.9	7th	11.7	12th
PAL	toolllama2-7b	1.6	20th	24.4	10th	4.6	18th	12.2	12th	10.2	18th	11.7	13th
ReAct	tulu-7b	4.0	19th	27.8	8th	7.9	15th	10.3	13th	13.2	13th	11.0	14th
DFSDT	gpt-4-1106	20.6	9th	9.6	12th	11.8	11th	9.9	15th	14.0	11th	10.9	15th
FC	gpt-4-1106	24.7	7th	10.9	11th	10.2	13th	8.2	18th	15.3	10th	9.9	16th
FC	gpt-3.5-turbo	19.0	11th	1.0	17th	15.9	9th	8.8	16th	12.0	15th	9.6	17th
ReAct	toolllama2-7b	15.0	14th	2.2	16th	5.7	17th	8.3	17th	7.6	19th	8.1	18th
DFSDT	gpt-3.5-turbo	20.7	8th	0.2	20th	13.8	10th	4.8	20th	11.6	16th	6.5	19th
ReAct	codellama-13b	0.2	21th	0.8	18th	0.7	21th	7.0	19th	0.6	21th	5.4	20th
DFSDT	toolllama2-7b	7.1	16th	0.0	21th	2.3	19th	3.5	21th	3.1	20th	3.4	21th

Fig.18 Evaluation Results of R-Eval on AMiner, wiki and overall ranking.

Evaluation Result - Performance Ranking

The same LLM & Domain

Different Workflow

Workflow	LLM	aminer KS		aminer KU		aminer KA		Overall Average (Level 1, 2, 3)					
		1-3	Rank	2-4	Rank	3-5	Rank	wiki	Rank	aminer	Rank	all	Rank
ReAct	gpt-4-1106	89.7	1st	46.7	3rd	57.7	1st	38.8	1st	64.7	1st	45.3	1st
PAL	gpt-3.5-turbo	80.1	3rd	50.7	2nd	54.9	2nd	19.9	6th	61.9	2nd	30.4	2nd
PAL	gpt-4-1106	59.3	4th	56.8	1st	52.7	3rd	20.3	5th	56.2	3rd	29.2	3rd
ReAct	llama2-7b-chat	45.2	5th	36.5	6th	21.5	6th	23.8	3rd	34.4	5th	26.4	4th
PAL	llama2-13b	25.3	6th	36.4	7th	20.3	7th	25.2	2nd	27.3	6th	25.7	5th
ReAct	gpt-3.5-turbo	84.6	2nd	4.0	14th	33.0	4th	19.6	7th	40.6	4th	24.9	6th
ReAct	vicuna-13b	19.9	10th	6.0	13th	7.1	16th	20.7	4th	11.0	17th	18.2	7th
PAL	tulu-7b	9.1	15th	26.8	9th	11.5	12th	18.9	8th	15.8	9th	18.1	8th
PAL	vicuna-13b	4.5	17th	40.9	4th	2.3	20th	16.7	9th	15.9	8th	16.5	9th
ReAct	llama2-13b	16.7	13th	0.7	19th	23.2	5th	15.0	10th	13.5	12th	14.6	10th
PAL	llama2-7b-chat	18.7	12th	2.8	15th	16.1	8th	12.4	11th	12.5	14th	12.4	11th
PAL	codellama-13b	4.4	18th	38.3	5th	8.1	14th	10.0	14th	16.9	7th	11.7	12th
PAL	toolllama2-7b	1.6	20th	24.4	10th	4.6	18th	12.2	12th	10.2	18th	11.7	13th
ReAct	tulu-7b	4.0	19th	27.8	8th	7.9	15th	10.3	13th	13.2	13th	11.0	14th
DFSDT	gpt-4-1106	20.6	9th	9.6	12th	11.8	11th	9.9	15th	14.0	11th	10.9	15th
FC	gpt-4-1106	24.7	7th	10.9	11th	10.2	13th	8.2	18th	15.3	10th	9.9	16th
FC	gpt-3.5-turbo	19.0	11th	1.0	17th	15.9	9th	8.8	16th	12.0	15th	9.6	17th
ReAct	toolllama2-7b	15.0	14th	2.2	16th	5.7	17th	8.3	17th	7.6	19th	8.1	18th
DFSDT	gpt-3.5-turbo	20.7	8th	0.2	20th	13.8	10th	4.8	20th	11.6	16th	6.5	19th
ReAct	codellama-13b	0.2	21th	0.8	18th	0.7	21th	7.0	19th	0.6	21th	5.4	20th
DFSDT	toolllama2-7b	7.1	16th	0.0	21th	2.3	19th	3.5	21th	3.1	20th	3.4	21th

Fig.18 Evaluation Results of R-Eval on AMiner, wiki and overall ranking.

Visilization of the performance

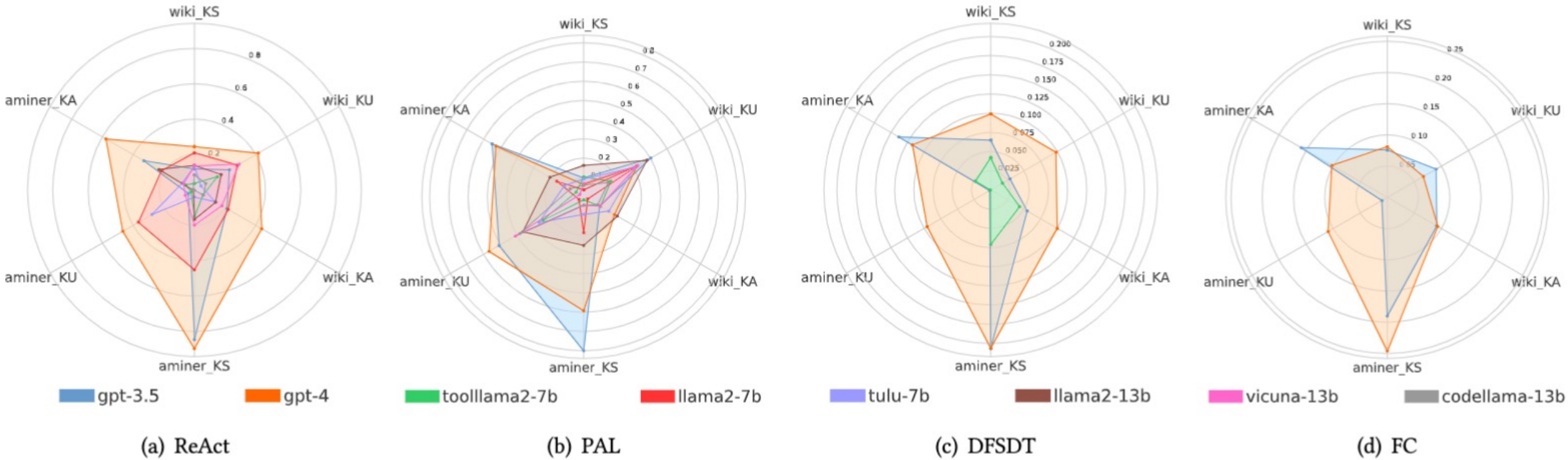


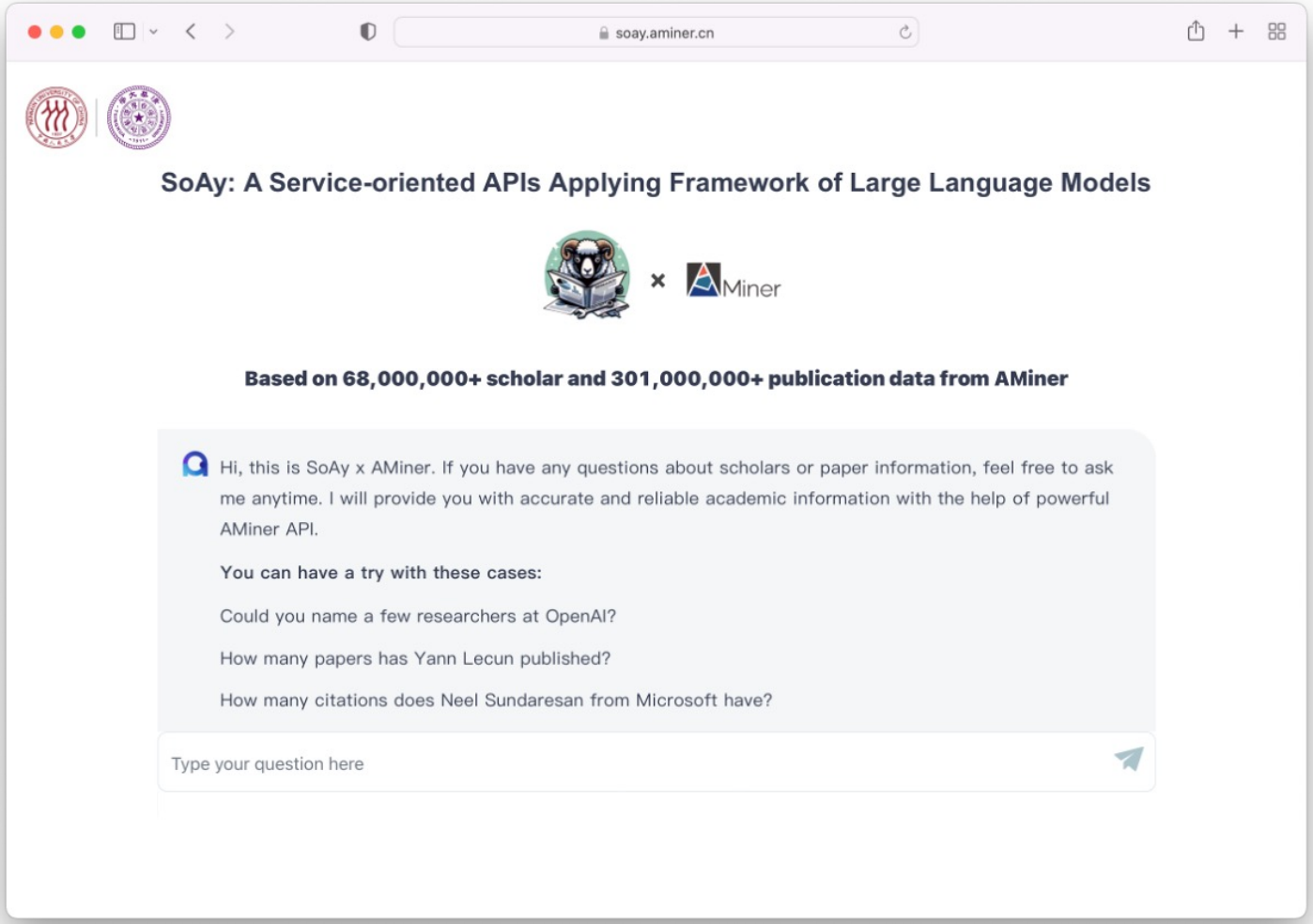
Fig.19 Radar map of single system's performance.

Welcome to Follow our work!

SoAy Applied System Link:
http://soay.aminer.cn

Github Link:
https://github.com/RUCKBReasoning/SoAy

R-Eval Github Link:
https://github.com/THU-KEG/R-Eval



Future Directions

From the viewpoint of tasks

- Structured data (Spreadsheet or word/pdf) processing -> manipulation -> analysis
- Tool creation

From the viewpoint of data

- How to generate instructions consistent with human query distribution?
- How to guarantee the correctness of the answer?

Form the viewpoint of fine-tuning method

- How to obtain the preference data to learn a reward model?
- How to leverage the feedback from model itself, especially on rigorous logical language generation?

Thank you!