Large Language Models (LLMs) on Tabular Data: Prediction, Generation, and Understanding – A Survey

Xi Fang, Weijie Xu, **Fiona Anting Tan**, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, Christos Faloutsos

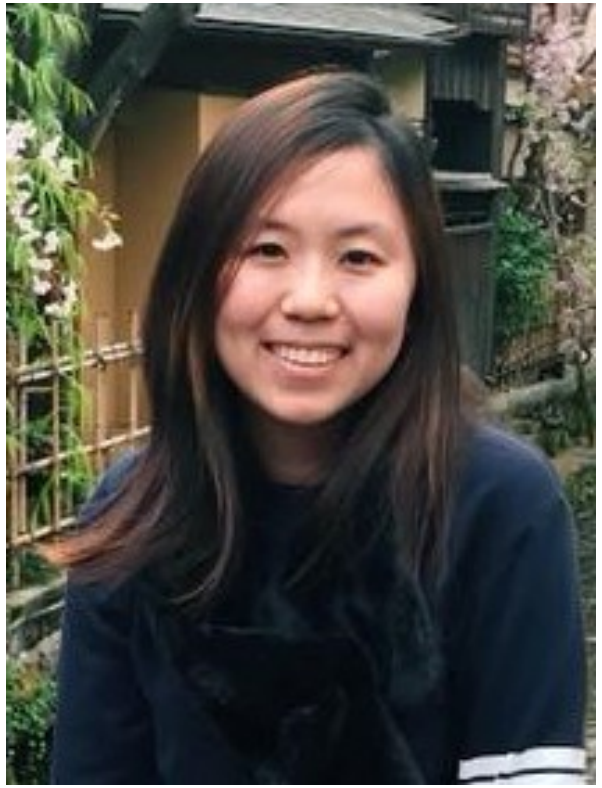
07 May 2024, Sharing with Hyundai Motor Group Innovation Centre Singapore (HMGICS)

About Me

Fiona Anting Tan

- 4th Year PhD at Institute of Data Science, National University of Singapore
- Research interest: Causal Relation Extraction, Text Mining, Natural Language Reasoning
- This paper was done during my internship at Amazon HQ in Seattle ☺

https://tanfiona.github.io/

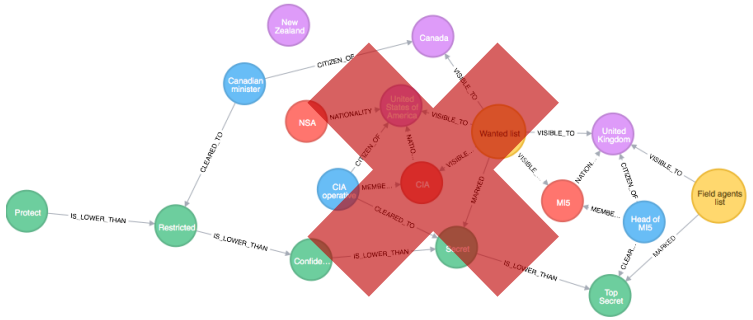


Agenda

- Tabular Data and LLMs
- Using LLMs for Tabular Data
- Tasks: Prediction, Generation and Understanding
- Conclusion & Future Directions

✓ Current Trends
✓ Key Techniques
✓ Possible Tasks

Tabular Data

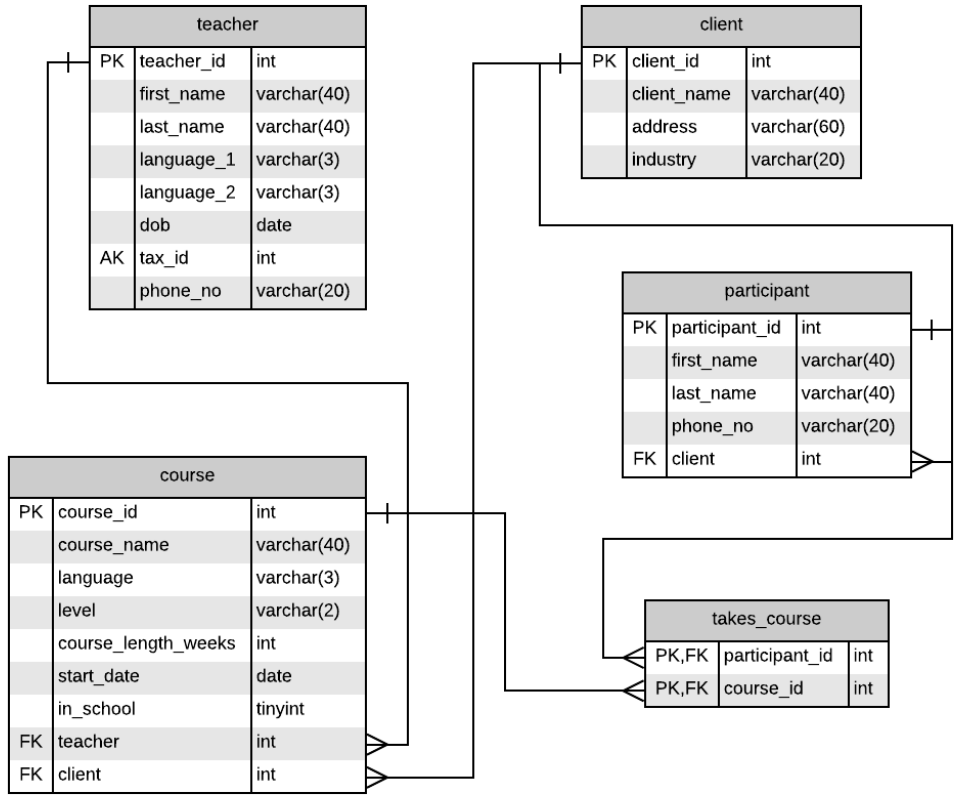


Tabular Data

columns = attributes for those observations

Rows = observations

Player	Minutes	Points	Rebounds	Assists
A	41	20	6	5
B	30	29	7	6
C	22	7	7	2
D	26	3	3	9
E	20	19	8	0
F	9	6	14	14
G	14	22	8	3
I	22	36	0	9
J	34	8	1	3

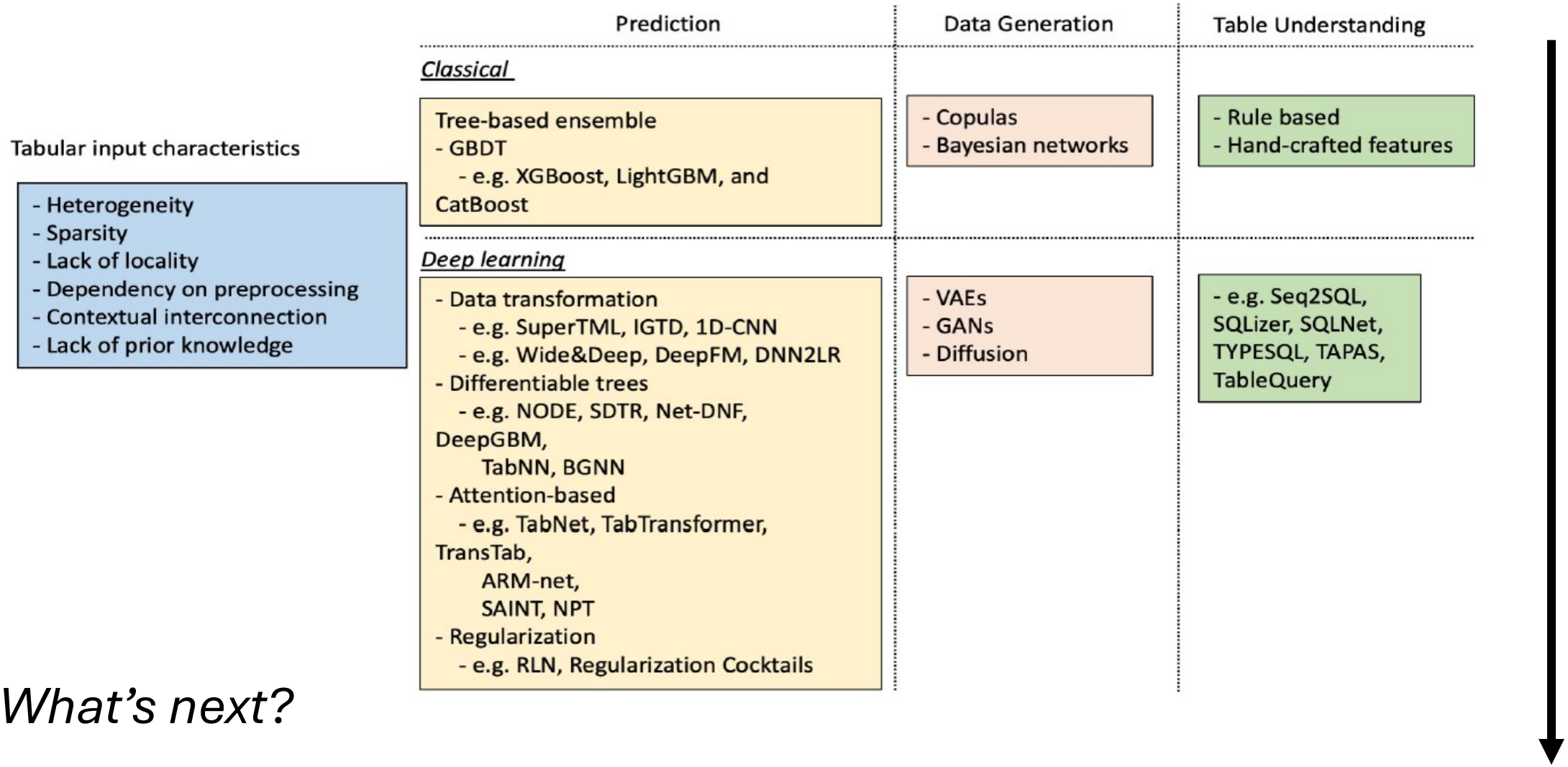


Characteristics of Tabular Data

- Heterogeneity
- Sparsity
- Dependency on pre-processing
- Correlation might matter
- Order invariant
- Lack of prior knowledge
  (of data structure)

Player	Minutes	Points	Rebounds	Assists
A	41	20	6	5
B	30	29		6
C	22			2
D	26	3	3	9
E	20	19	8	0
F		6	14	14
G	14	22	8	3
I	22	36		9
J	34	8	1	3

Evolution of Tabular Data Modelling



What's next?

Evolution of Language Models

Generalizability / Transferability

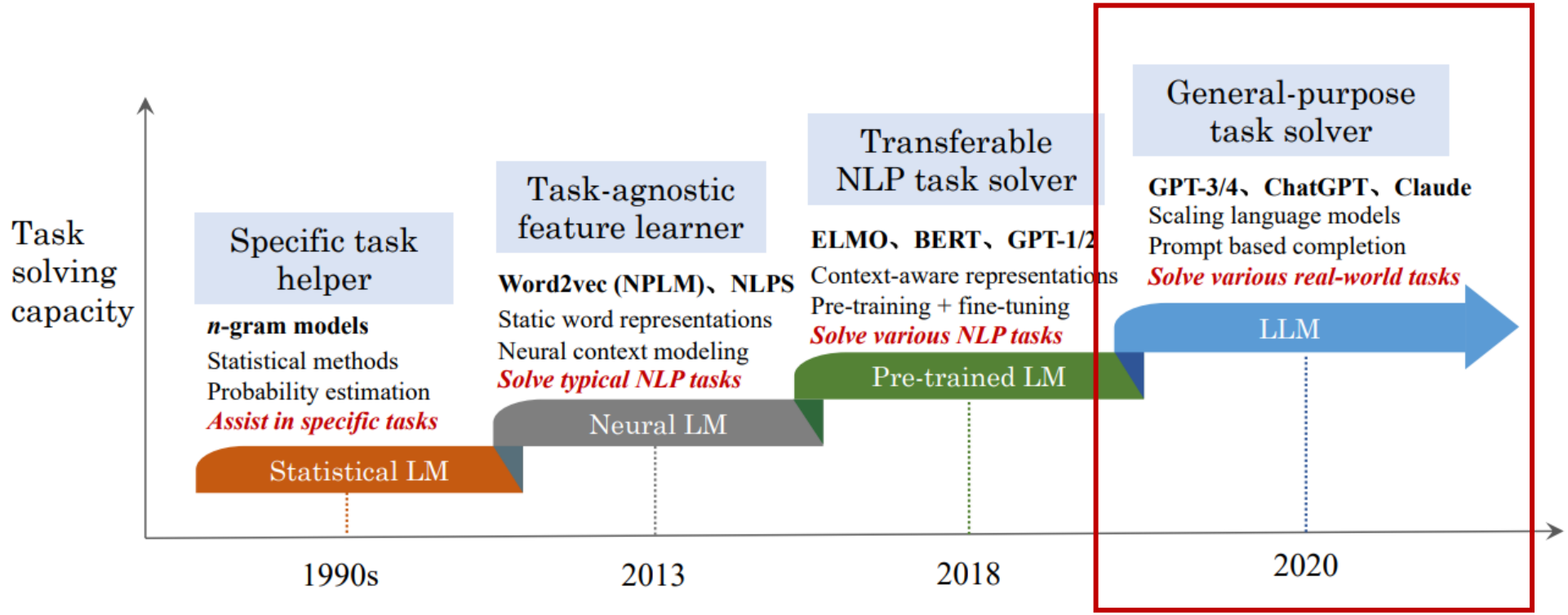


Fig. 2: An evolution process of the four generations of language models (LM) from the perspective of task solving capacity.

For our survey, we focus on models that have ≥ 1 billion parameters.

Zhao et al., A Survey of Large Language Models. CoRR abs/2303.18223 (2023)

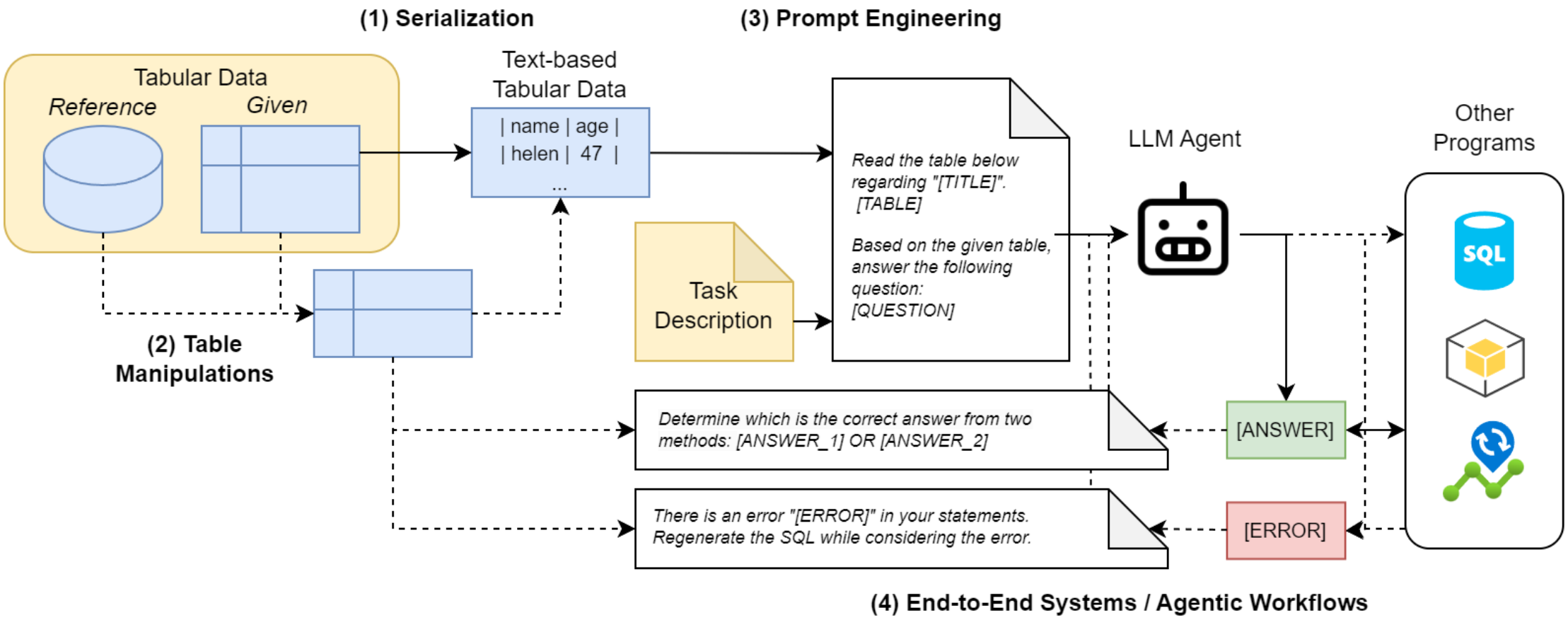
Reasons to use LLMs for Tabular Data

LLMs (compared to ML/ DL methods) :

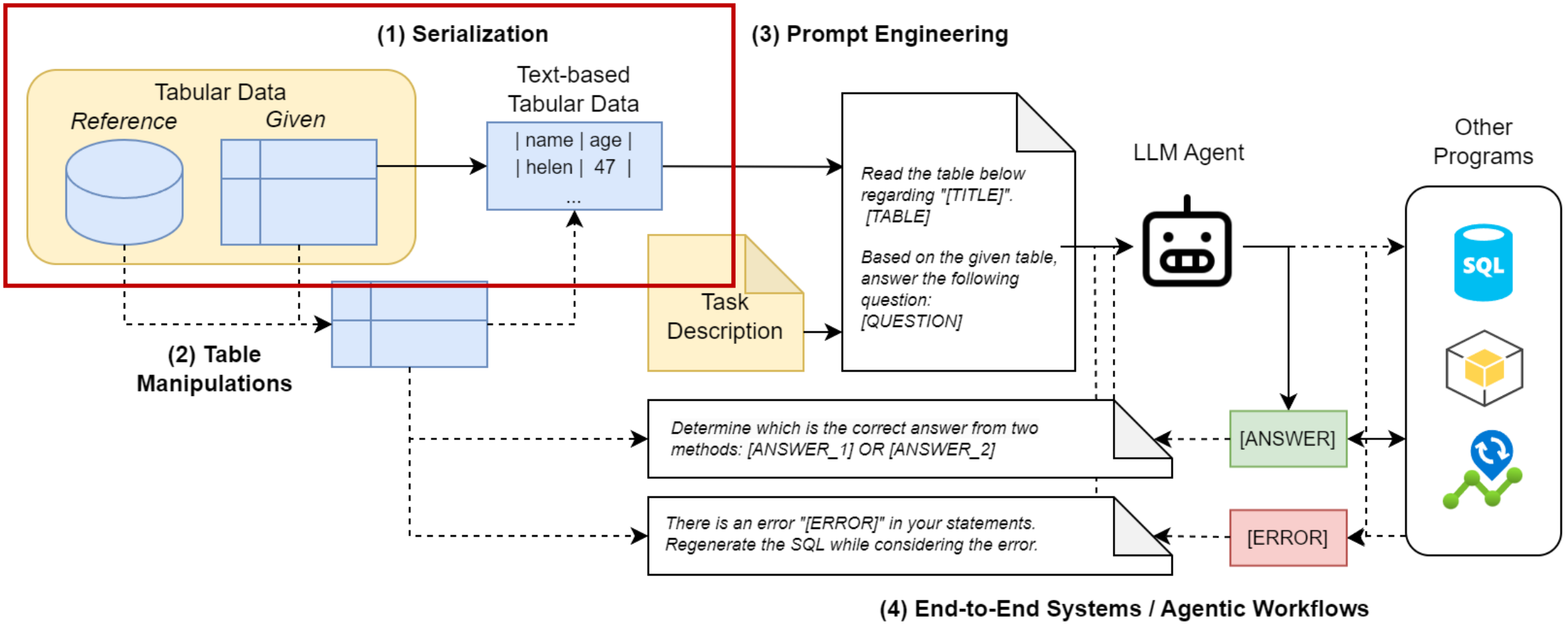
1. **Have superior transfer learning abilities**
   - Useful if you lack data (sample efficient)
2. **Require “less” pre-processing**
   - Handle heterogeneous inputs
   - “No need” to deal with missing values, categorical encoding, etc.
3. **Exhibit general task-solving abilities**
   - Useful for reasoning tasks/ if you need explanations
4. **End-to-end systems/ Agentic workflows**
   - Useful for improved performance using feedback loops
   - Useful for applications that connects to other programs

Using LLMs for Tabular Data

Key Techniques in using LLMs for Tabular Data



Key Techniques in using LLMs for Tabular Data



Serialization

- Text-based

Method	Description	Example	Papers that investigated this
DFLoader	Python code where a dictionary is loaded as a Pandas dataframe	`pd.DataFrame({ name:[‘helen’], age:[47] })`	Singha et al. (2023)
JSON	Row number as indexes, with each row represented as a dictionary of keys (column names) and values	`{"0": {"name": "helen", "age": "47"}}`	Singha et al. (2023); Sui et al. (2023b)
Data Matrix	Dataframe as a list of lists, where the firm item is the column header	`[[‘’,‘name’,‘age’] [0, ‘helen’, 47]]`	Singha et al. (2023)
Markdown	Rows are line-separated, columns are separated by “|” [1]	`| | name | age |` `|:--|:-----|----:|` `|0 |helen | 47|`	Singha et al. (2023); Liu et al. (2023e); Zhang et al. (2023d); Ye et al. (2023b); Zhao et al. (2023d); Sui et al. (2023b)
LaTeX	Rows are separated by “\\\hline”, columns are separated by “&”	\\\hline helen & 47	Jaitly et al. (2023)
X-Separated	Rows are line-separated, columns are separated by “,”, “\t”, “:”, etc.	`, name, age` `0, helen, 47`	Singha et al. (2023); Narayan et al. (2022)
Attribute-Value Pairs	Concatenation of paired columns and cells `{c : v}`	`name:helen ; age:47`	Wang et al. (2023c)
HTML	HTML element for tabular data	`<table><thead><tr><th></th> <th>name</th><th>age</th></tr> </thead><tbody><tr><th>0</th> <td>helen</td><td>47</td></tr> </tbody></table>`	Singha et al. (2023); Sui et al. (2023c;b)
Sentences	Rows are converted into sentences using templates	`name is helen, age is 47`	Yu et al. (2023); Hegselmann et al. (2023); Gong et al. (2020); Dinh et al. (2022); Jaitly et al. (2023)

Table 1: Text-based serialization methods.

Serialization

text-davinci-003
& gpt-4 (subset of data)

- Text-based

Format	Table Partition		Cell Lookup		Reverse Lookup		Column Retrieval		Row Retrieval		Size Detection		Merged Cell Detection	
	Acc	GPT-4	Acc	GPT-4	Acc	GPT-4	Acc	GPT-4	Acc	GPT-4	Acc	GPT-4	Acc	GPT-4
NL + Sep	93.00%	96.78%	39.67%	72.48%	52.00%	59.12%	60.67%	66.32%	31.00%	48.67%	42.00%	73.12%	71.33%	74.98%
Markdown	92.33%	**98.32%**	43.33%	71.93%	51.00%	57.32%	35.33%	60.12%	**42.33%**	49.98%	40.67%	82.12%	**78.00%**	**82.64%**
JSON	94.00%	97.12%	42.67%	68.32%	54.33%	58.12%	54.33%	64.32%	29.00%	48.32%	42.67%	76.43%	73.33%	78.98%
XML	96.00%	97.64%	43.33%	72.28%	**55.00%**	**60.32%**	41.33%	68.28%	41.00%	**50.28%**	43.67%	80.21%	75.00%	80.32%
HTML	**96.67%**	**98.32%**	**44.00%**	**73.34%**	47.33%	59.45%	**63.33%**	**69.32%**	42.00%	50.19%	**67.00%**	**83.43%**	76.67%	81.28%

Table 1: Average Pass@1 for fact-finding tasks. DFLoader provides overall high pass@1 performance.

Table Formats	ColumnLookupTests	DataTypeLookupTests	NavigationTests	RowLookupTests	Overall
CommaSeparated	64.43	95.00	65.57	78.14	75.78
DFLoader	72.71	95.29	68.29	82.86	**79.79**
DataMatrix	62.57	84.00	56.57	**87.43**	72.64
Json	65.00	**96.43**	**71.43**	78.86	77.93
Markdown	61.43	85.86	48.71	73.29	67.32
TabSeparated	67.00	94.00	64.43	78.14	75.8
HTML	**79.83**	94.67	58.83	52.33	71.4
HTMLNoSpace	73.00	93.50	62.00	59.50	72.00

Table 2: F1 scores for transformation tasks. DFLoader and JSON format, with structural element isolation and repetition, enable high performance on average across transformation tasks.

Table	TableColumnReorderTests	TableReconstructionTests	TableTransposeTests	Overall
CommaSeparated	95.33	74.33	99.00	89.55
DFLoader	99.33	**98.00**	98.33	**98.55**
DataMatrix	92.67	90.67	0.00	61.11
Json	**99.67**	85.00	**100.00**	94.89
Markdown	50.00	24.33	34.00	36.11
TabSeparated	93.33	92.33	50.00	78.55
HTML	50.00	86.00	83.33	73.11
HTMLNoSpace	83.33	84.00	83.33	83.55

Format	TabFact	HybridQA	SQA	Feverous	ToTTo
	Acc	Acc	Acc	Acc	BLEU-4
NL + Sep	70.26%	45.02%	70.41%	75.15%	**12.70%**
Markdown	68.40%	45.88%	66.59%	71.88%	8.57%
JSON	68.04%	42.40%	70.39%	73.84%	8.82%
XML	70.00%	47.20%	70.74%	73.14%	8.82%
HTML	**71.33%**	**47.29%**	**71.31%**	**75.20%**	12.30%

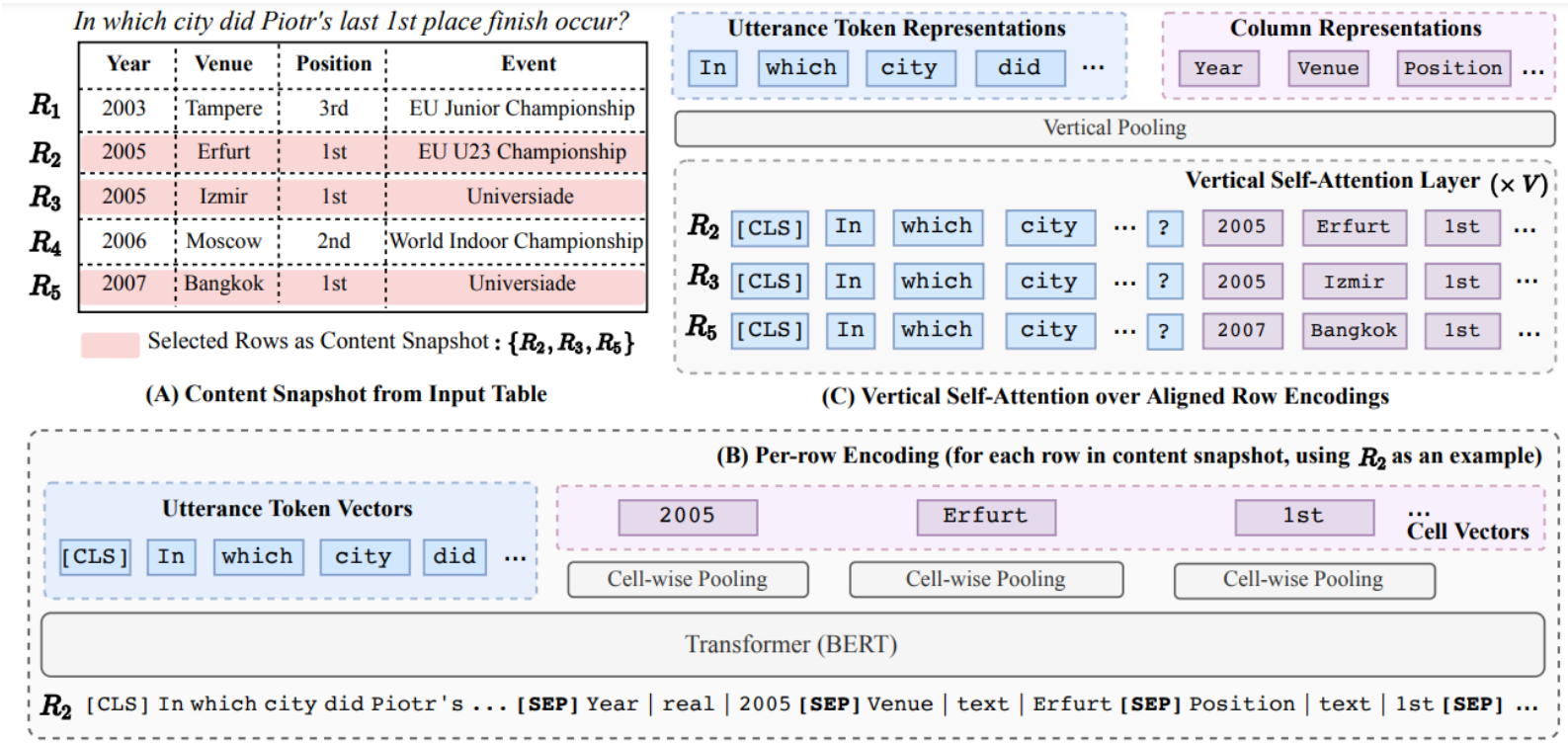
Sui et al., Evaluating and Enhancing Structural Understanding Capabilities of Large Language Models on Tables via Input Designs. CoRR abs/2305.13062 (2023)

Singha et al., Tabular Representation, Noisy Operators, and Impacts on Table Structure Understanding Tasks in LLMs. Table Representation Learning Workshop at NeurIPS 2023

Serialization

- Text-based
- Embedding-based
  - Use **embeddings as inputs** (esp. if fine-tuning downstream model)

TaBERT (not LLM)



Unsupervised learning objectives:

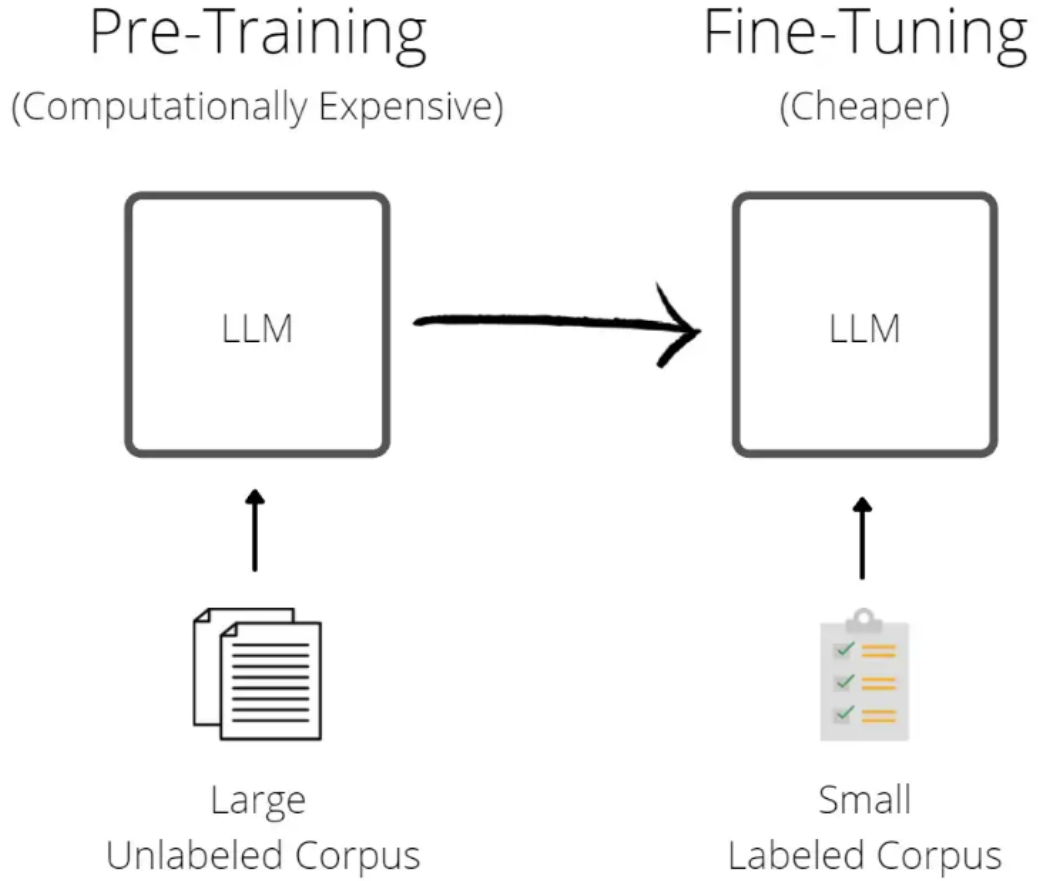
- Masked Column Prediction (MCP)
- Cell Value Recovery (CVR)

Yin et al., TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data. ACL 2020: 8413-8426

Sidetrack… LM Training Methods

- Unsupervised or Self-supervised pre-training (General Task)
- Supervised fine-tuning (Downstream Task)
  - Instruction Tuning (Chatbot-like)
  - Application Specific
- In-context learning (Prompt engineering)

Model weights are changing



https://contenteratechspace.com/a-comprehensive-guide-to-varieties-of-llm-training/

Serialization

- Text-based
- Embedding-based
  - Use **embeddings as inputs** (esp. if fine-tuning downstream model)
  - Use LLM to **generate sentences** (Hegselmann et al., 2023)

Traditional Pre-training

Model	Input	Output Embedding	Downstream Task
TURL	Table + metadata	Entity / Col. / Col. pair	Table interpretation/augmentation
DODUO	Table	Col. / Col. pair	Column type/relation prediction
TAPAS	NL question + table	Question / Table	Semantic parsing
TaBERT	NL question + table	Col. / Table	Semantic parsing
TaPEx	SQL query + table	Row / Table	Table Question Answering
TapTap	Table	Row	Data augmentation/imputation

Cong et al., Observatory: Characterizing Embeddings of Relational Tables. Proc. VLDB Endow. 17(4): 849-862 (2023)

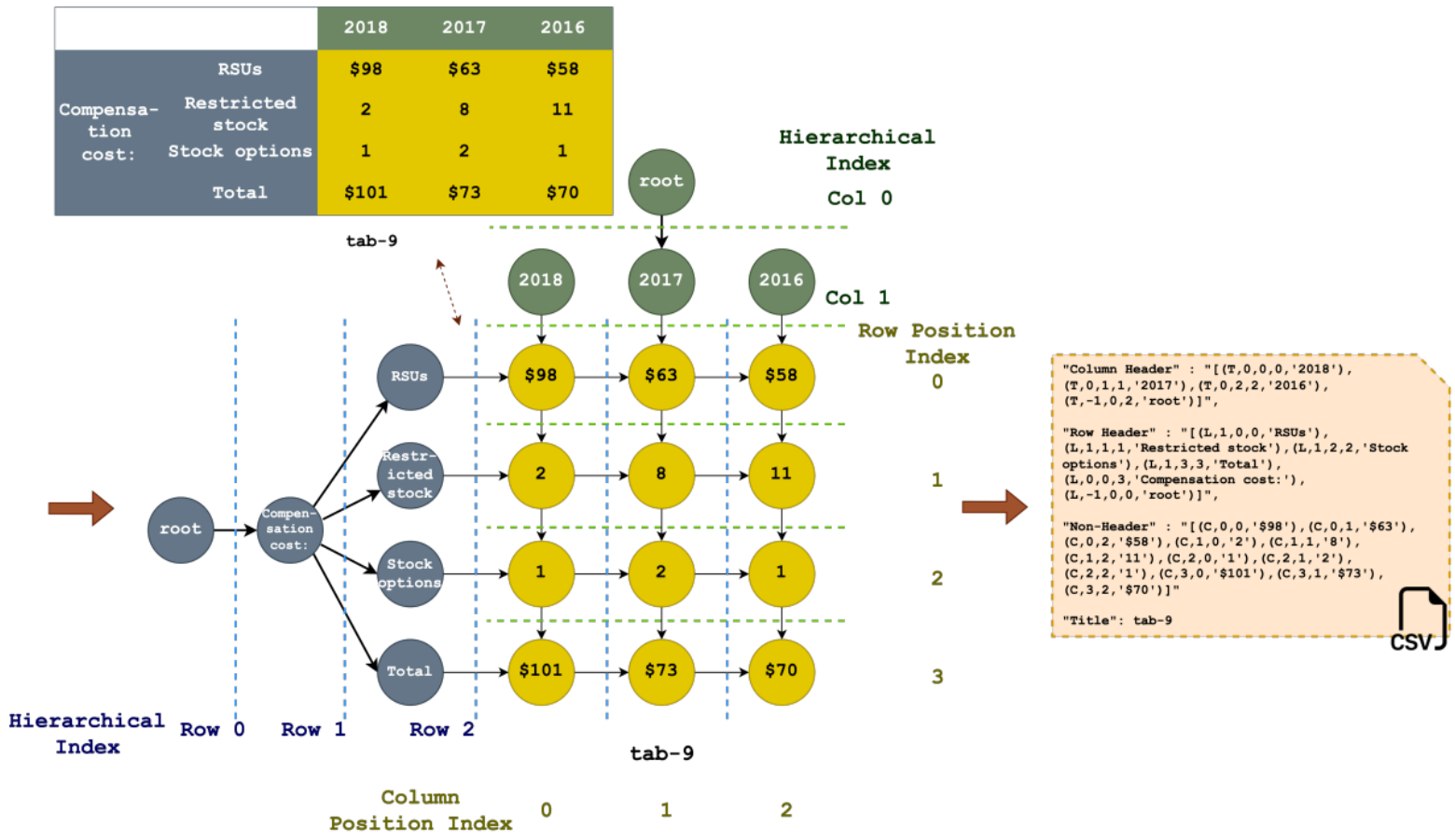
Fine-tuning **on multiple tasks**

LLMs with >1B parameters

- UniTabPT (Sarkar & Lausen, 2023) with 3B parameters (based on T5 and Flan-T5 models))
- TableGPT (Gong et al., 2020) with 1.5B parameters (based on GPT2)
- TableGPT2 (Zha et al., 2023) with 7B parameters (based on Phoenix 7B (Chen et al., 2023b))
- TableLlama (Zhang et al., 2023f) with 7B parameters (based on Llama 2 (Touvron et al., 2023))
- TableGPT with 350M, 3B, 13B or 175B parameters (based on various versions of OpenAI's GPT models) (Li et al., 2023)

Serialization

- Text-based
- Embedding-based
- Graph-based & Tree-based



Zhao et al., Large Language Models are Complex Table Parsers. EMNLP 2023: 14786-14802

Key Techniques in using LLMs for Tabular Data

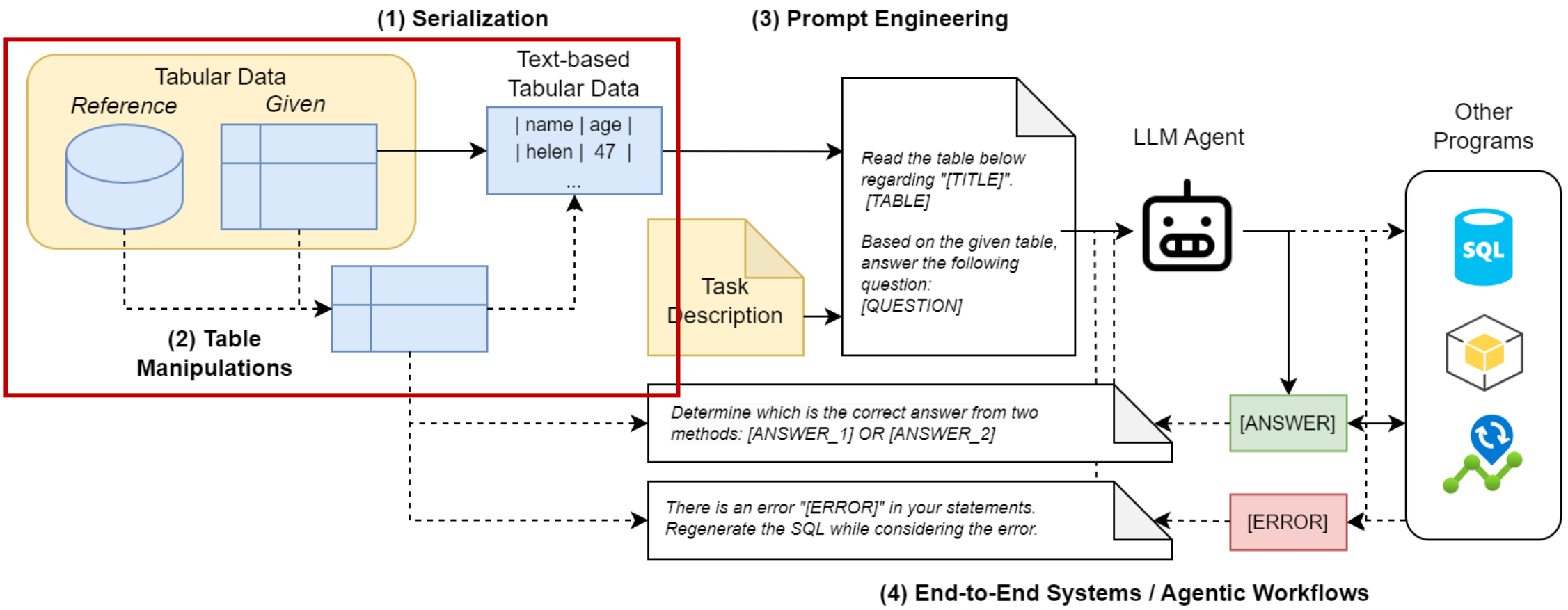


Table Manipulations

Why would we want to manipulate the table?

1. **To fit context lengths**
   - No choice: Restricted context length
   - Have choice: Shorter context length == cheaper & better attention
2. **Main table vs. auxiliary (other) tables**
   - E.g. metadata, table schemas
   - Better performance
3. **Investigate/ ensure robustness of results**
   - E.g. perturb data (transpose and shuffle) shouldn't affect, e.g. QA outcome about sum of a column.

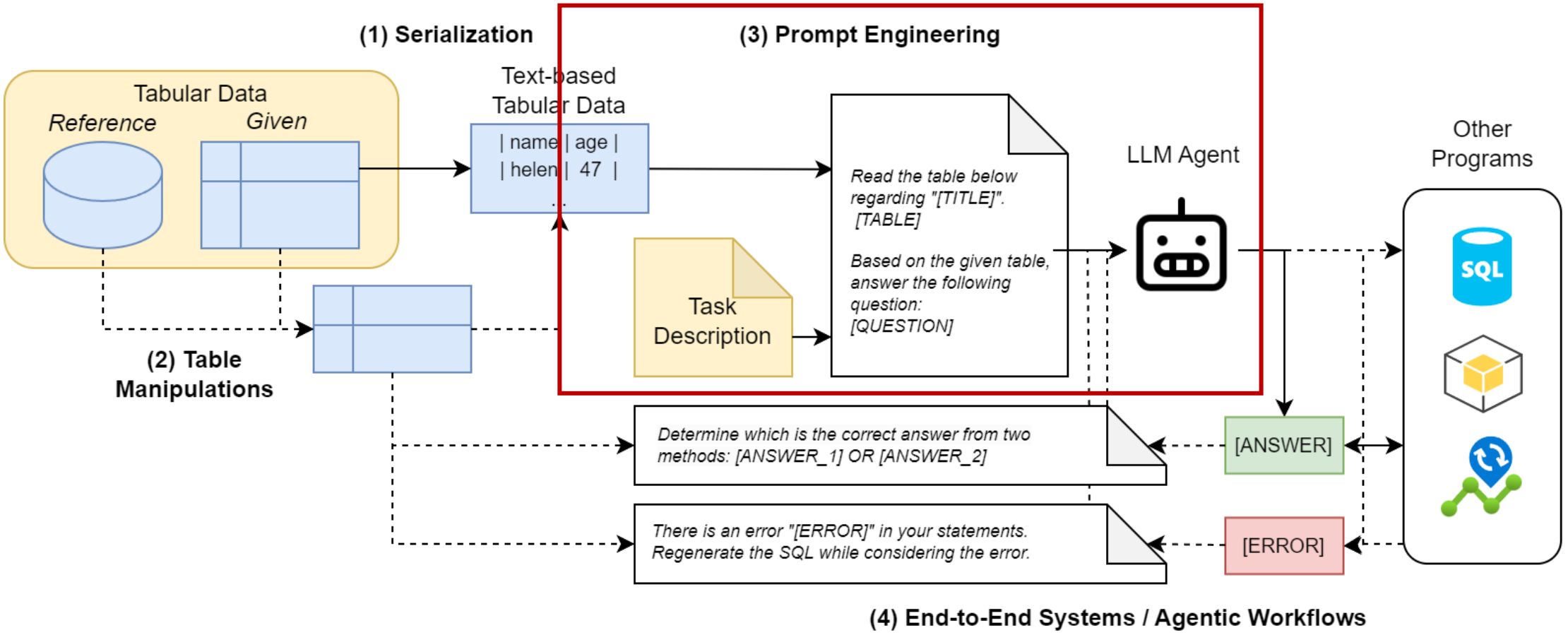
Filtering, sampling, search, retrieval…

Metadata, summary statistics, column descriptions…

Create new features: Bucketing, dummy variables, …

Transpose, shuffling, semantically renaming columns, …

Key Techniques in using LLMs for Tabular Data

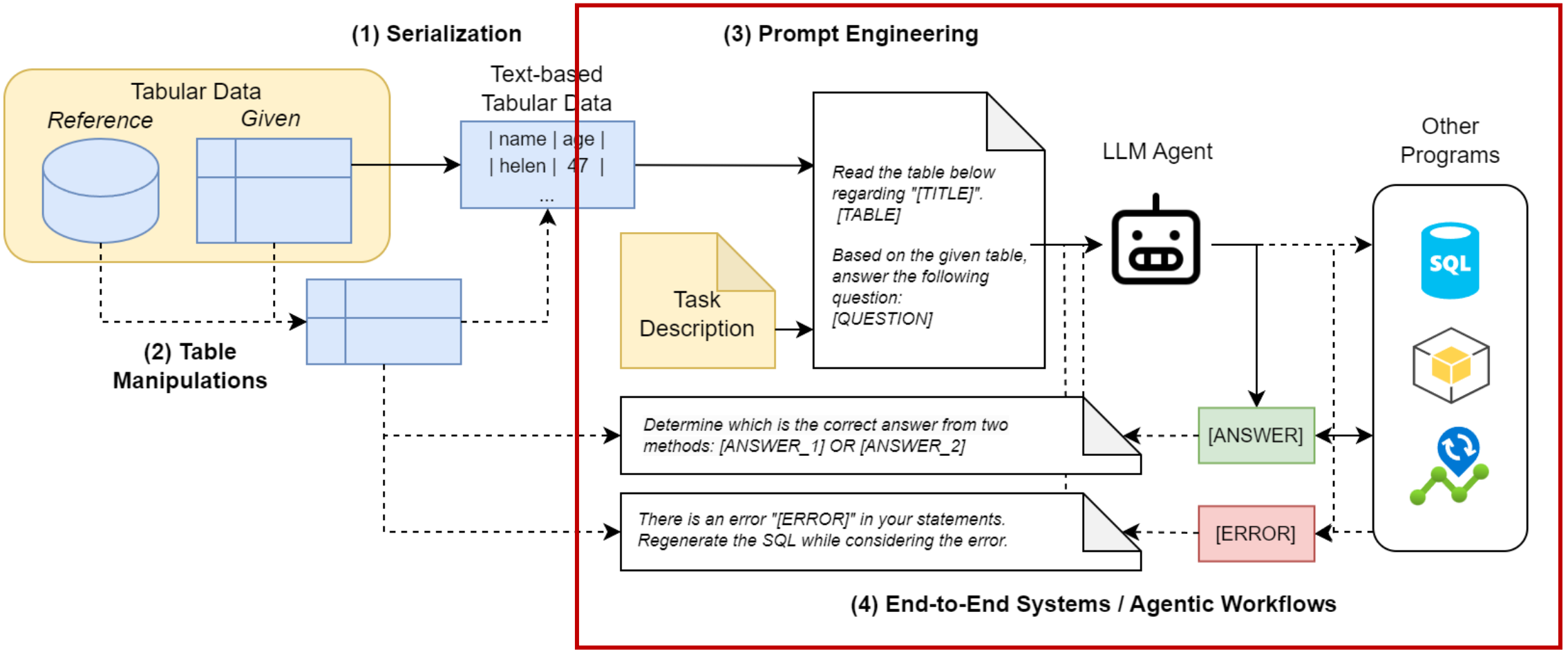


Prompt Engineering

Similar to the rest of the LLM literature:

- **Prompt format**
- **In-context learning**
- **Chain-of-Thought (CoT)** and **Self-Consistency**
- **Retrieval-augmented generation (RAG)**
- **Role-play**

Key Techniques in using LLMs for Tabular Data



End-to-End Systems / Agentic Workflows

- Improve performance
  - Better performance through **self-consistency**
  - **Self-debugging** through **feedback loops**
  - **Decompose task** into subtasks

- Build end-to-end systems, with the ability to
  - Connect to and support **multiple platforms**
  - **Support user-interactions** / Dialogue-based applications
  - **Provide explanations** for outcomes

LLMs for Tabular Data Prediction, Generation and Understanding

Prediction

Let's look at a classification task

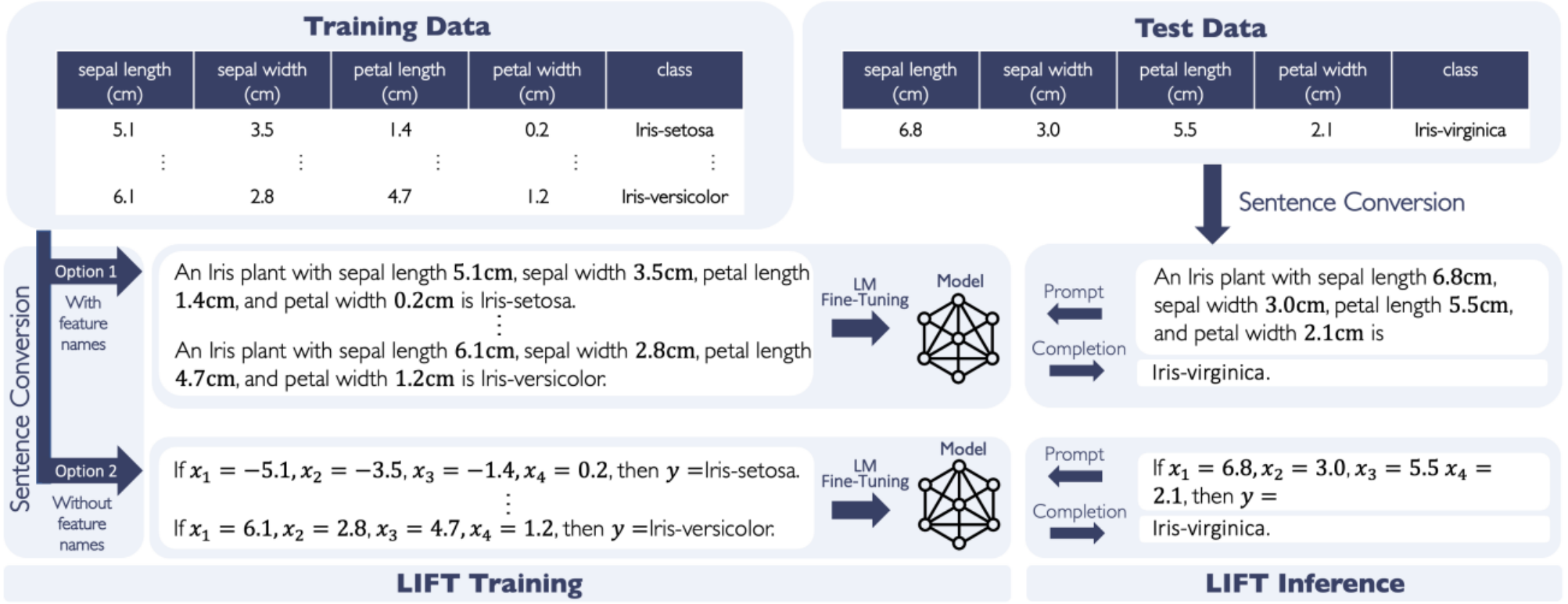


Figure 1: **A high-level illustration of the Language-Interfaced Fine-Tuning (LIFT) framework.**

Dinh et al., LIFT: Language-Interfaced Fine-Tuning for Non-language Machine Learning Tasks. NeurIPS 2022

Prediction

Dataset (ID)	p / c	MCC	LogReg	DT	RBF-SVM	XG	LIFT/GPT-J	LIFT/GPT-3
Tabular Data (OpenML)								
Customers (1511)	8 / 2	68.18	**87.12±0.54**	85.98±0.53	86.36±0.00	85.23±0.00	85.23±1.61	84.85±1.42
Pollution (882)	15 / 2	50.00	58.33±11.79	**77.78±3.93**	58.33±6.81	63.89±7.86	63.89±3.93	63.89±7.86
Spambase (44)	57 / 2	60.59	93.27±0.00	90.7±0.14	93.70±0.00	**95.87±0.00**	94.03±0.54	94.90±0.36
Hill-Valley (1479)	100 / 2	49.79	77.78±0.00	56.38±0.89	68.72±0.00	59.26±0.00	**100.00±0.20**	**99.73±0.19**
IRIS (61)	4 / 3	33.33	96.67±0.00	97.77±3.85	**100.00±0.00**	**100.00±0.00**	96.67±0.00	97.0±0.00
TAE (48)	5 / 3	35.48	45.16±4.56	65.59±5.49	53.76±6.63	**66.67±8.05**	61.29±6.97	65.59±6.63
CMC (23)	9 / 3	42.71	49.49±0.83	56.72±0.32	56.50±0.97	52.43±0.42	49.83±0.28	**57.74±0.89**
Wine (187)	13 / 3	38.89	**100.00±0.00**	93.52±2.62	**100.00±0.00**	97.22±0.00	93.52±1.31	92.59±1.31
Vehicle (54)	18 / 4	25.88	80.39±1.00	63.92±2.37	**81.18±0.48**	73.14±0.28	64.31±2.37	70.20±2.73
LED (40496)	7 / 10	11.00	68.67±0.94	66.33±2.87	68.00±0.82	66.00±0.82	65.33±0.47	**69.33±2.05**
OPT (28)	64 / 10	10.14	96.53±0.22	89.8±1.09	97.95±0.00	97.48±0.17	98.22±0.11	**98.99±0.30**
Mfeat (12)	216 / 10	10.00	97.67±0.12	87.67±1.05	**98.83±0.24**	96.75±0.00	94.17±1.75	93.08±0.24
Margin (1491)	64 / 100	0.94	81.35±0.15	43.86±1.21	**81.98±0.30**	70.21±0.29	50.23±1.33	59.37±0.92
Texture (1493)	64 / 100	0.94	81.67±0.97	46.88±1.93	**83.44±0.89**	70.73±1.41	50.32±2.18	67.50±1.42

- LIFT achieves comparable performance to most baselines.
- When the number of classes is large (~100s), both LIFT/GPT models have lower accuracies than many baselines.

Dinh et al., LIFT: Language-Interfaced Fine-Tuning for Non-language Machine Learning Tasks. NeurIPS 2022

Prediction

- Do we need the pre-trained LLM on natural language?

Table 8: **Accuracies (↑) of LIFT with different LMs.** We compare variants of LIFT with different LMs: LIFT/GPTs using GPTs pretrained on natural language data (our models), LIFT/Rand-GPT-J using a randomly initialized GPT-J, LIFT/CodeGen and LIFT/CodeParrot using LMs pretrained on programming language data, and LIFT/Gibberish using GPT-J fine-tuned on gibberish data.

Dataset (ID)	MCC	LIFT/GPT-3	LIFT/GPT-J	LIFT/Rand-GPT-J	LIFT/Gibberish	LIFT/CodeGen	LIFT/CodeParrot
Blobs (2)	25.00	96.67± 0.24	96.17± 0.59	25.65± 1.58	96.42± 0.24	93.67± 0.72	93.39± 1.82
Two Circles (6)	50.00	81.42± 0.82	75.92± 1.65	49.88± 5.01	68.67± 1.50	53.02± 0.66	50.08± 2.47
Iris (61)	33.33	97.0± 0.00	96.67± 0.00	27.78± 20.79	94.44± 1.57	43.31± 6.67	60.00± 8.82
Customers (1511)	68.18	84.85± 1.42	85.23± 1.61	52.47± 7.15	67.43± 1.42	45.96± 8.96	43.11± 3.34
Wine (187)	38.89	92.59± 1.31	93.52± 1.31	22.22± 15.71	84.26± 3.46	77.78± 0.00	33.88± 3.87
LED (40496)	11.0	69.33± 2.05	65.33± 0.47	11.68± 4.44	72.67± 1.25	11.00± 4.00	23.46± 13.85

Dinh et al., LIFT: Language-Interfaced Fine-Tuning for Non-language Machine Learning Tasks. NeurIPS 2022

Prediction

- Finetuning or not? Does the training data size matter?

Table 5: **Comparison of accuracies (↑) between ICL and fine-tuning with LIFT on OpenML datasets.** "LIFT/Full-Data" and "LIFT/Subset" represent LIFT on the full dataset and and its subset used correspondingly in the ICL setting (number of prompts). Here, the size of subset is chosen to satisfy the LMs' context length. Overall, LIFT/GPTs on full data achieve the best performances. However, when using the same number of samples, LIFT and ICL are more comparable in most cases. Note that both methods may be worse than MCC due to the limited training data in some cases.

Dataset (ID)	#Prompts	MCC	GPT-J			GPT-3		
			In-Context	LIFT/Subset	LIFT/Full-data	In-Context	LIFT/Subset	LIFT/Full-data
Breast (13)	35	70.69	56.90±19.51	**58.62±2.44**	64.94±11.97	62.07±1.41	**70.69±0.00**	71.26±1.62
TAE (48)	50	35.48	**34.33±1.47**	32.26±9.50	61.29±4.56	**37.64±4.02**	33.33±1.52	65.59±6.63
Vehicle (54)	14	25.88	**25.49±0.55**	26.04±1.69	64.31±2.37	**28.82±2.10**	23.73±2.27	70.20±2.73
Hamster (893)	43	53.33	48.89±3.14	**60.00±10.88**	55.55±16.63	**57.78±6.29**	53.33±0.00	53.33±0.00
Customers (1511)	29	68.18	56.06±17.14	**59.85±2.84**	85.23±1.61	60.61±1.42	**63.26±6.96**	84.85±1.42
LED (40496)	33	68.67	10.00±0.82	**13.04±3.27**	65.33±0.47	8.00±1.63	**11.33±2.62**	69.33±2.05

Dinh et al., LIFT: Language-Interfaced Fine-Tuning for Non-language Machine Learning Tasks. NeurIPS 2022

Prediction

- Exploring CoT: Ask GPT-3 to explain its own prediction result

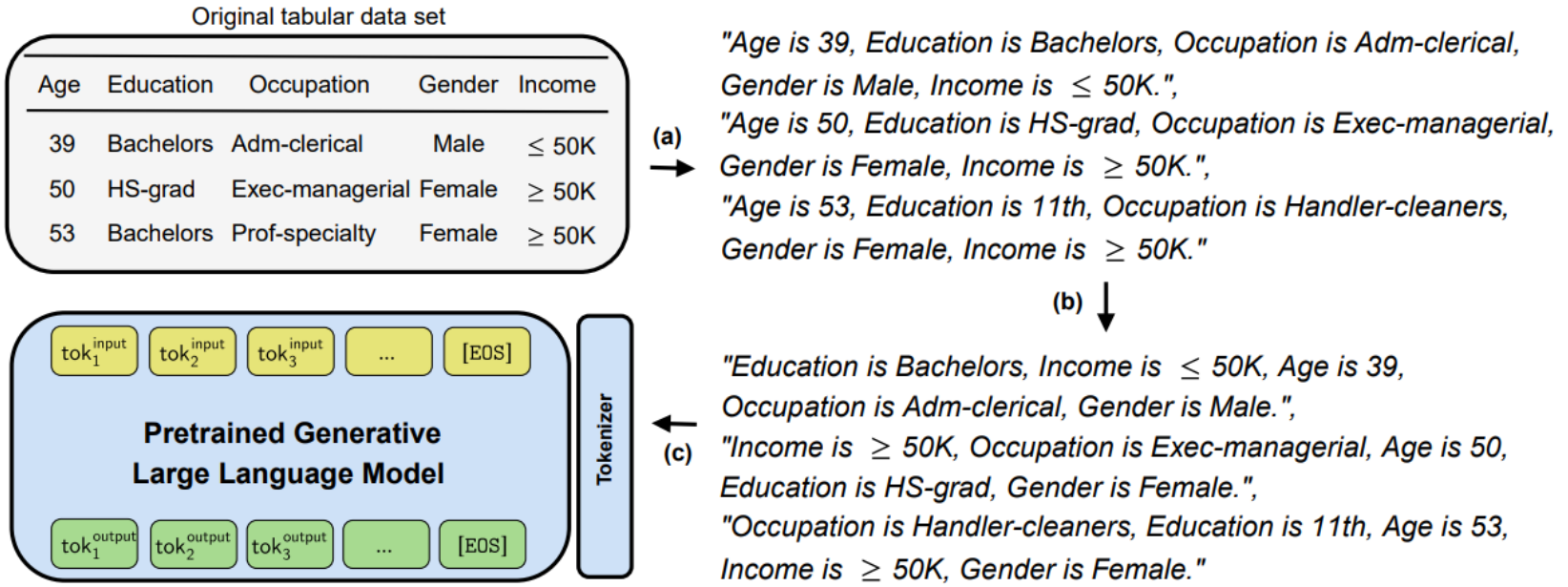
	Prompt or Generated Text
Raw input	Checking Account: less than 200 DM; Loan Duration: 48 months; Credit History: no credits/paid back duly; Loan Purpose: education; Number of applying for credits: 18424; Savings Account: less than 100 DM; Employment Time: 1~4 years; Installment Rate: 1%; Sex: Female; Other Debtors: No; Property: building society savings agreements/life insurance; Age: 32; Housing: self-owned house; Job: highly qualified employee/ self-employed; Foreigner: no
First prompt (soliciting prediction)	A 32-year-old female is applying for a loan of 18424 credits for 48 months for other purposes. She has a checking account with less than 200 Deutsche Mark and a savings account with less than 100 Deutsche Mark. She had no credits or paid back all credits duly. The installment rate of this applicant in percentage is 1%. She is employed for more than one year but less than four years and is recognized as a highly qualified employee or is self-employed. She has lived in a self-owned house. She owns some building society savings agreements or life insurance. Given such information, will the bank accept or decline this loan application?
First response	accept
Second prompt (soliciting explanation)	A 32-year-old female is applying for a loan of 18424 credits for 48 months for other purposes. She has a checking account with less than 200 Deutsche Mark and a savings account with less than 100 Deutsche Mark. She had no credits or paid back all credits duly. The installment rate of this applicant in percentage is 1%. She is employed for more than one year but less than four years and is recognized as a highly qualified employee or is self-employed. She has lived in a self-owned house. She owns some building society savings agreements or life insurance. Given such information, will the bank accept or decline this loan application? **\n The bank will accept this loan application. Let's think logically. This is because**
Second response (random seed 1)	the applicant is employed and is qualified in a high level. Since she is employed, she is able to make all the payments. Furthermore, she has a self-owned house and owns some building society savings agreements or life insurance. She has less than 2000 Deutsche Mark in checking and savings account. She had no credits or paid back all credits duly. She is able to pay the loans. All of these are positive factors to this loan. The bank will accept this loan.
Second response (random seed 2)	the applicant has a high income. She has lived in a self-owned house. She has some savings agreements or life insurance. She is a highly qualified employee. She has no negative credits. All of these facts show that she is not in danger of not being able to pay back the loan. Therefore, the bank will accept this loan application.

https://openreview.net/forum?id=s_PJMEGIUfa

Dinh et al., LIFT: Language-Interfaced Fine-Tuning for Non-language Machine Learning Tasks. NeurIPS 2022

Generation

- Distilled GPT-2 (Sanh et al., 2019) with 82M parameters
- GPT-2 (Radford et al., 2019) with 355M parameters



(a) Serialization

(b) Feature order permutation

(c) Fine-tuning

Figure 2: The GReaT data pipeline for the fine-tuning step. First, a textual encoding step transforms

$$p(\boldsymbol{t}) = p(w_1, \ldots, w_j) = \prod_{k=1}^{j} p(w_k | w_1, ..., w_{k-1}).$$

maximize the probability $\prod_{\boldsymbol{t} \in \mathcal{T}} p(\boldsymbol{t})$

Borisov et al., Language Models are Realistic Tabular Data Generators. ICLR 2023

Generation

$p(V_1, \ldots, V_n)$

$p(V_{\setminus\{i\}} | V_i = v_i)$

$p(V_{\setminus\{i_1,\ldots,i_k\}} | V_{i_1} = v_{i_1}, \ldots, V_{i_k} = v_{i_k})$

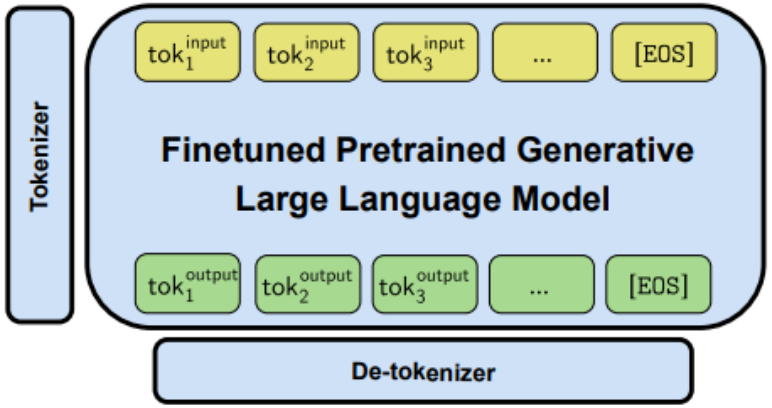
Input text sequences (Arbitrary conditioning)

["***Age***"]

["***Age is 26,***"]

["***Education is Masters, Age is 59,***"]

(a)



(b)

Synthetic tabular data set

Age	Education	Occupation	Gender	Income
23	11th	Adm-clerical	Missing	$\leq$ 50K
26	HS-grad	Sales	Female	$\geq$ 50K
59	Masters	Other-service	Male	$\geq$ 50K

(c)

*"**Age** is 23, Occupation is Adm-clerical, Income is $\leq$ 50K, Gender is Missing, Education is 11th, "*

*"**Age is 26**, Income is $\geq$ 50K, Occupation is Sales, Education is HS-grad, Gender is Female"*

*"**Education is Masters, Age is 59,** Occupation is Other-service, Gender is Male, Income is $\geq$ 50K"*

Figure 3: The sampling procedure of the proposed method for synthetic data generation.

Borisov et al., Language Models are Realistic Tabular Data Generators. ICLR 2023

Generation

Dataset		Original	TVAE	CopulaGAN	CTGAN	**Distill-GReaT**	**GReaT**
Travel (↑)	LR	82.72±0.00	79.58±0.00	73.30±0.00	73.30±0.00	78.53±0.00	**80.10±0.00**
	DT	89.01±0.00	81.68±1.28	73.61±0.26	73.30±0.00	77.38±0.51	**83.56±0.42**
	RF	85.03±0.53	81.68±1.19	73.30±0.00	71.41±0.53	79.90±0.53	**84.30±0.33**
Sick (↑)	LR	96.69±0.00	94.70±0.00	94.57±0.00	94.44±0.00	96.56±0.00	**97.72±0.23**
	DT	98.94±0.29	95.39±0.18	93.77±0.01	92.05±0.41	95.39±0.44	**97.72±0.23**
	RF	99.28±0.06	94.91±0.06	94.57±0.01	94.57±0.00	97.72±0.10	**98.30±0.13**
HELOC (↑)	LR	71.80±0.00	71.04±0.00	42.03±0.00	57.72±0.00	70.58±0.00	**71.90±0.00**
	DT	81.90±0.02	76.39±0.10	42.36±0.10	61.34±0.09	**81.40±0.15**	79.10±0.07
	RF	83.19±0.21	77.24±0.25	42.35±0.34	62.35±0.35	**82.14±0.13**	80.93±0.28
Adult Income (↑)	LR	85.00±0.00	80.53±0.00	80.61±0.00	83.20±0.00	84.65±0.00	**84.77±0.00**
	DT	85.27±0.01	82.80±0.08	76.29±0.06	81.32±0.02	84.49±0.04	**84.81±0.04**
	RF	85.93±0.11	83.48±0.11	80.46±0.21	83.53±0.07	85.25±0.07	**85.42±0.05**
Diabetes (↑)	LR	58.76±0.00	56.34±0.00	40.27±0.00	50.93±0.00	57.33±0.00	**57.34±0.00**
	DT	57.29±0.03	53.30±0.09	38.50±0.02	49.73±0.02	54.10±0.04	**55.23±0.04**
	RF	59.00±0.08	55.17±0.10	37.59±0.31	52.23±0.17	58.03±0.16	**58.34±0.09**
California Housing (↓)	LR	0.40±0.00	0.65±0.00	0.98±0.00	0.61±0.00	0.57±0.00	**0.34±0.00**
	DT	0.32±0.01	0.45±0.01	1.19±0.01	0.82±0.01	0.43±0.01	**0.39±0.01**
	RF	0.21±0.01	0.35±0.01	0.99±0.01	0.62±0.01	0.32±0.01	**0.28±0.01**

Table 1: ML efficiency experiment. The best results are marked in **bold**, the second-best results are underlined.

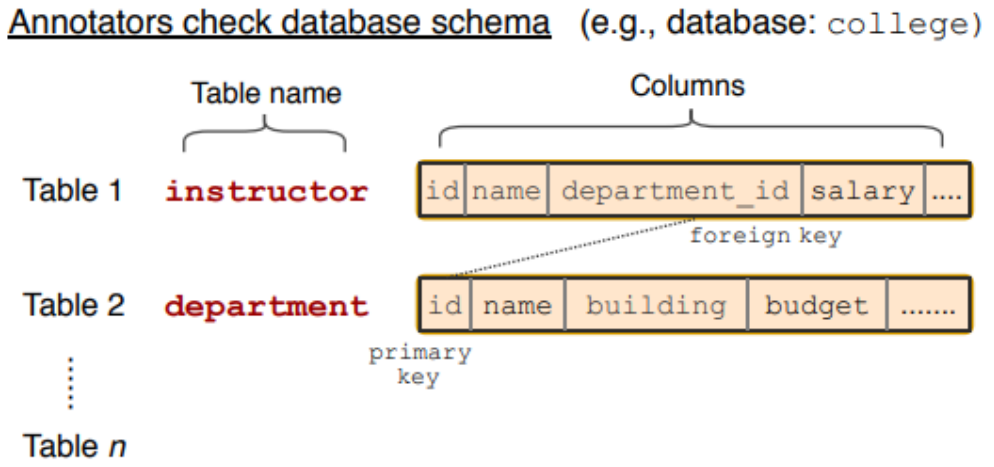
Borisov et al., Language Models are Realistic Tabular Data Generators. ICLR 2023

Understanding

Spider dataset (Wang et al., 2018) and Leaderboard

Rank	Model	Test
1 Nov 2, 2023	MiniSeek *Anonymous* Code and paper coming soon	**91.2**
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

https://yale-lily.github.io/spider



Annotators create:

Complex question What are the name and budget of the departments with average instructor salary greater than the overall average?

Complex SQL

```
SELECT T2.name, T2.budget
FROM instructor as T1 JOIN department as
T2 ON T1.department_id = T2.id
GROUP BY T1.department_id
HAVING avg(T1.salary) >
    (SELECT avg(salary) FROM instructor)
```

Figure 1: Our corpus annotates complex questions and SQLs. The example contains joining of multiple tables, a GROUP BY component, and a nested query.

Understanding

- Text-to-SQL task (Gao et al., 2024) proposes:
  - Question representation

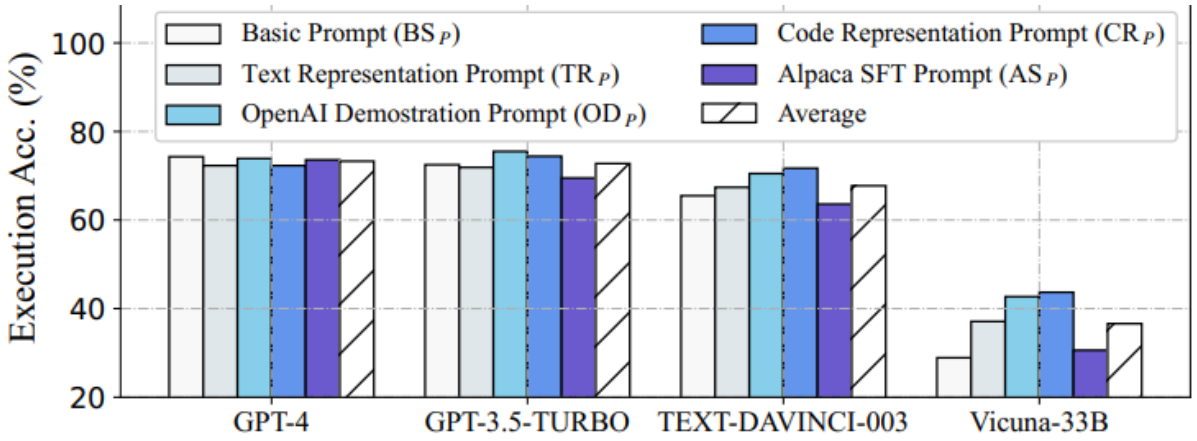


Figure 1: Results of different question representations on Spider-dev under zero-shot scenario.

```
Table continents, columns = [ContId, Continent]
Table countries, columns = [CountryId, CountryName,
↳ Continent]
Q: How many continents are there?
A: SELECT
```

Listing 1: Example of Basic Prompt

```
### Complete sqlite SQL query only and with no
↳ explanation
### SQLite SQL tables, with their properties:
#
# continents(ContId, Continent)
# countries(CountryId, CountryName, Continent)
#
### How many continents are there?
SELECT
```

Listing 3: Example of OpenAI Demostration Prompt

Gao et al., Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. Proc. VLDB Endow. 17(5): 1132-1145 (2024)

Understanding

- **Question Similarity Selection (QTS):** Embeds questions (target and examples), kNN search for top K questions
- **Masked:** Replace table names, column names and values
- **Query:** Preliminary model to generate SQL query using question and database

- Text-to-SQL task (Gao et al., 2024) proposes:
  - Example selection for in-context learning

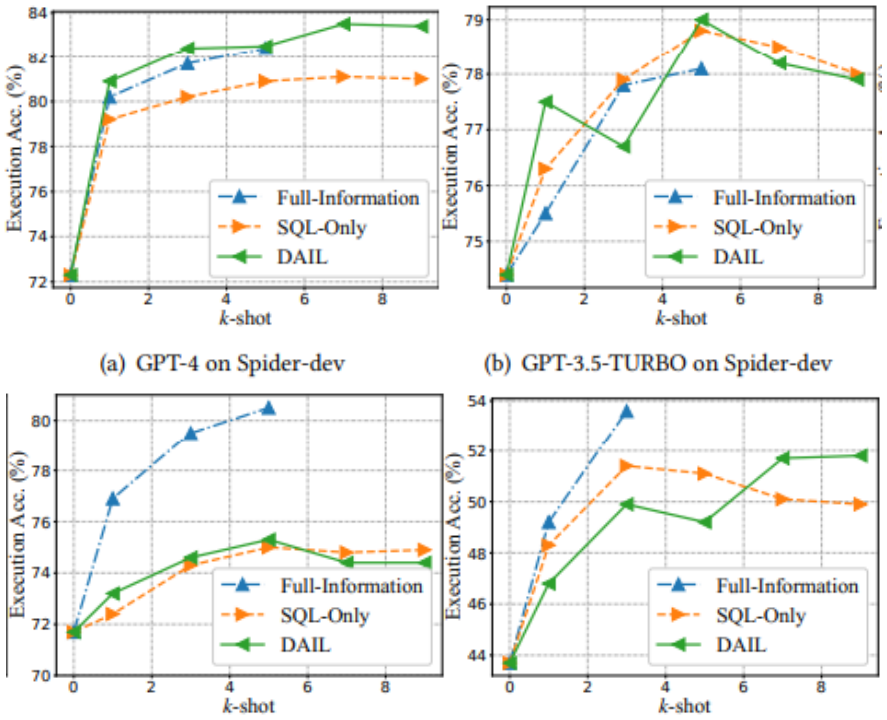
Table 2: Evaluation on Spider-dev with different example selections. The organization is fixed to Full-Information Organization.

Few-shot	Selection	Question Similarity	Query Similarity	GPT-4		GPT-3.5-TURBO		TEXT-DAVINCI-003		Vicuna-33B	
				EM	EX	EM	EX	EM	EX	EM	EX
0-shot	-	-	-	22.1	72.3	34.6	74.4	31.7	71.7	6.9	43.7
1-shot	Random	0.23	0.47	41.7	77.4	45.9	73.9	38.2	70.6	14.4	47.9
	Question Similarity Selection	0.39	0.65	53.3	78.8	51.9	74.3	44.1	72.3	16.5	48.5
	Masked Question Similarity Selection	0.57	0.80	58.2	79.1	57.4	76.0	47.9	75.0	21.4	48.7
	DAIL Selection	0.56	0.95	62.1	80.2	59.5	75.5	51.9	76.9	22.8	49.2
	Upper Limit	0.56	0.98	63.7	81.0	61.4	77.2	53.1	77.5	22.7	49.4
3-shot	Random	0.23	0.48	48.9	79.4	49.0	73.6	41.7	71.6	16.8	46.9
	Question Similarity Selection	0.37	0.63	56.3	79.2	53.8	74.7	52.2	74.1	21.1	47.1
	Masked Question Similarity Selection	0.54	0.78	66.1	81.5	61.1	77.3	59.7	77.0	27.7	52.3
	DAIL Selection	0.53	0.94	69.1	81.7	63.9	77.8	64.4	79.5	30.7	53.6
	Upper Limit	0.53	0.98	71.5	83.4	66.2	79.2	66.7	81.1	31.2	54.4
5-shot	Random	0.23	0.48	51.6	79.5	52.9	75.7	49.0	72.1	-	-
	Question Similarity Selection	0.36	0.61	58.2	79.9	55.9	75.1	54.8	73.2	-	-
	Masked Question Similarity Selection	0.52	0.77	66.8	82.0	62.3	77.9	64.7	78.6	-	-
	DAIL Selection	0.52	0.94	71.9	82.4	66.7	78.1	67.7	80.5	-	-
	Upper Limit	0.51	0.97	74.4	84.4	68.8	79.6	70.7	82.4	-	-

Gao et al., Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. Proc. VLDB Endow. 17(5): 1132-1145 (2024)

Understanding

- Text-to-SQL task (Gao et al., 2024) proposes:
  - Example organization for in-context learning



(a) GPT-4 on Spider-dev (b) GPT-3.5-TURBO on Spider-dev

Figure 4: Evaluation on Spider-dev with different example organizations.

```
/* Given the following database schema: */
${DATABASE_SCHEMA}
/* Answer the following: How many authors are there? */
SELECT count(*) FROM authors

/* Given the following database schema: */
${DATABASE_SCHEMA}
/* Answer the following: How many farms are there? */
SELECT count(*) FROM farm

${TARGET_QUESTION}
```

Listing 6: Example of Full-Information Organization.

```
/* Some SQL examples are provided based on similar
↳ problems: */
SELECT count(*) FROM authors

SELECT count(*) FROM farm

${TARGET_QUESTION}
```

Listing 7: Example of SQL-Only Organization.

```
/* Some example questions and corresponding SQL queries
↳ are provided based on similar problems: */
/* Answer the following: How many authors are there? */
SELECT count(*) FROM authors

/* Answer the following: How many farms are there?. */
SELECT count(*) FROM farm

${TARGET_QUESTION}
```

Listing 8: Example of DAIL Organization.

Gao et al., Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. Proc. VLDB Endow. 17(5): 1132-1145 (2024)

Understanding

- Text-to-SQL task (Gao et al., 2024) proposes:
  - Supervised fine-tuning (on open-sourced models)

LLM	Org.	0-shot		1-shot		3-shot		5-shot	
		EM	EX	EM	EX	EM	EX	EM	EX
LLaMA -7B	FI_O	3.1	13.0	23.4	30.1	23.7	30.3	24.7	30.9
	SO_O	3.1	13.0	13.3	21.4	15.2	24.1	15.3	25.0
	DAIL_O	3.1	13.0	18.5	25.4	22.1	28.1	22.6	29.3
+ SFT	FI_O	63.9	66.7	59.6	61.4	58.7	61.4	59.4	61.5
	SO_O	63.9	66.7	59.8	62.3	58.8	61.1	59.5	62.2
	DAIL_O	63.9	66.7	58.5	61.9	59.8	61.7	58.9	60.9
LLaMA -13B	FI_O	2.4	20.3	21.6	33.8	27.3	38.1	28.5	38.8
	SO_O	2.4	20.3	20.7	33.6	23.2	35.9	27.4	36.9
	DAIL_O	2.4	20.3	13.2	30.0	15.5	32.3	16.2	32.4
+ SFT	FI_O	62.7	67.0	61.9	67.1	60.5	65.0	60.9	65.0
	SO_O	62.7	67.0	61.9	66.2	60.1	64.6	60.2	65.2
	DAIL_O	62.7	67.0	62.5	66.5	60.6	66.0	61.3	66.4

Table 4: Few-shot evaluation results of supervised fine-tuned LLMs on Spider-dev.

Gao et al., Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. Proc. VLDB Endow. 17(5): 1132-1145 (2024)

Understanding

- Text-to-SQL task (Gao et al., 2024) proposes:
  - Question representation
  - Example selection for in-context learning
  - Example organization for in-context learning
  - Supervised fine-tuning*
  - Self-consistency

Rank	Model	Test
1 Nov 2, 2023	MiniSeek *Anonymous* Code and paper coming soon	**91.2**
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

Gao et al., Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. Proc. VLDB Endow. 17(5): 1132-1145 (2024)

Fine-tuning or not?

T-8: Entity Matching

Instruction: Please determine whether the first and second row below may refer to the same entity. Please answer "yes/no" in JSON.

Row-1:
| Brand | Model | Price | Description | ...
| Apple | iPhone-15 | $995 | ... | ...
Row-2:
| Brand | Model | Price | Description | ...
| Apple | iPhone pro | $1500 | ... | ...

Model response:
The first and second product are likely different, because they have very different prices. Therefore, the answer is **{"answer": "No"}.**

T-9: Error Detection (misspellings)

Instruction: Please examine the table below and check which cell may be misspelled. If there is no erroneous cell, return 'None'. Please answer using JSON: {"answer": ""}.

Table:
| State | Capital | Population | ...
| Washington | Olympia | 7.7 million | ...
| Missisipi | Jackson | 2.9 million | ...

Model response:
The cell " Missisipi" is misspelled, and should be corrected as " Mississippi". Therefore, the answer is **{"answer": "Missisipi"}.**

Figure 7: Example table-tasks we generate for (T-8) Entity-matching, and (T-9) Error-detection, using "augmented-completions" that contain reasoning steps similar to chain-of-thought, which when used as training-data in table-tuning, can ground model responses and improve result quality.

Figure 8: Overall quality improvement, between vanilla GPT-3.5 and Table-GPT-3.5.

Figure 9: Overall quality improvement, between vanilla ChatGPT and Table-ChatGPT.

Li et al., Table-GPT: Table-tuned GPT for Diverse Table Tasks. CoRR abs/2310.09263 (2023)

Fine-tuning or not?

Prediction

Algorithm	Type	Method	Resource	Metric	Used Model
TabletSlack & Singh (2023)	Tabular	No Finetune	Low	F1	GPTJ/Tk-Instruct/Flan T5
SummaryBoostManikandan et al. (2023)	Tabular	No Finetune	High	RMSE	GPT3
LIFTDinh et al. (2022)	Tabular	Finetune	High	MAE/RMSE	GPT3/GPTJ
TabLLMHegselmann et al. (2023)	Tabular	Finetune	High	AUC	GPT3/T0
UnipredictWang et al. (2023a)	Tabular	Finetune	Low	ACC	GPT2
GTLZhang et al. (2023a)	Tabular	Finetune	Low	ACC	LLaMA
SerializeLLMJaitly et al. (2023)	Tabular	Finetune	High	AUC	T0
MediTabWang et al. (2023c)	Medical	Finetune	High	PRAUC/AUCROC	BioBert/GPT3.5/UnifiedQA-v2-T5
CTRLLi et al. (2023c)	Finance	Finetune	High	AUC/LogLoss	Roberta/ChatGLM
FinPTYin et al. (2023)	CTR	Finetune	High	F1 Score	FlanT5/ChatGPT/GPT4

Generation

	Used LLM	Fine-tuned or not	Serialization	Metric
GReaT (Borisov et al., 2023b)	GPT2/DistilGPT2	Fine-tuned	Sentences	DCR, MLE
REaLTabFormer (Solatorio & Dupriez, 2023)	GPT2	Fine-tuned		DCR, MLE
TAPTAP (Zhang et al., 2023e)	GPT2/DistilGPT2	Fine-tuned	Sentences	DCR, MLE
TabuLa (Zhao et al., 2023f)	DistilGPT2	Fine-tuned	X-Separated	MLE
CLLM (Seedat et al., 2023)	GPT4	Non Fine-tuned	X-Separated	MLE
TabMT (Gulati & Roysdon, 2023)	Masked Transformers -24layer	Fine-tuned	"[Value]"	MLE

Table 6: Data synthesis methods. "DCR" stands for Distance to the Closest Record and "MLE" stands for Machine Learning Efficiency.

Fine-tuning or not?

Understanding

Paper	Task	Models Explored
DOCMATH-EVAL (Zhao et al., 2023d)	NumQA	GPT4, GPT3.5, WizardLM, Llama-2 7, 13, 70B, CodeLlama 34B, Baichuan, Qwen, WizardMath, Vicuna, Mistral, etc.
Akhtar et al. (2023)	NumQA	TAPAS, DeBERTa, TAPEX, NT5, LUNA, PASTA, ReasTAP, FlanT5, GPT3.5, PaLM
TableGPT (Gong et al., 2020)	NumQA	GPT2
DATER (Ye et al., 2023b)	QA	GPT3 Codex
Chen (2023)	QA	GPT3
cTBLS (Sundar & Heck, 2023)	QA	Custom: Dense Table Retrieval based on RoBERTa + Coarse State Tracking + Response based on GPT3.5
GPT4Table (Sui et al., 2023b)	QA	GPT-3.5, GPT-4
Zhao et al. (2023a)	QA	GPT-3.5
Liu et al. (2023e)	QA	GPT3.5
TableGPT (Zha et al., 2023)	QA	Custom: Phoenix-7B
TAP4LLM (Sui et al., 2023c)	QA	Instruct GPT3.5, GPT4
UniTabPT (Sarkar & Lausen, 2023)	QA	Custom: T5, Flan-T5
Yu et al. (2023)	Multi-modal QA	Custom: Retrieval trained on contrastive loss, Rank by softmax, Generation built on T5
TableLlama (Zhang et al., 2023g)	QA	Custom: TableLlama
DIVKNOWQA Zhao et al. (2023c)	QA	GPT3.5, DSP, ReAct
Jiang et al. (2023)	QA	GPT3.5, ChatGPT3.5
Liu et al. (2023c)	QA & Text2SQL	Vicuna, GPT4
Gao et al. (2024)	Text2SQL	GPT4
Pourreza & Rafiei (2023)	Text2SQL	GPT4
Huang et al. (2023b)	Text2SQL	GPT4
Dong et al. (2023)	Text2SQL	ChatGPT3.5
Chang & Fosler-Lussier (2023)	Text2SQL	GPT3 Codex, ChatGPT3.5
Zhang et al. (2023d)	Text2SQL	LLaMA2 70b
Abraham et al. (2022)	Text2SQL	Custom: Table Selector + Known & Unknown Fields Extractor + AggFn Classifier

Conclusion

- The general-purpose problem-solving capabilities of LLMs inspire researchers to explore using LLMs for tabular data.
- There are four key techniques/ steps: (1) Serialization, (2) Table manipulations, (3) Prompt engineering, and (4) End-to-end systems/ agentic workflows.
- Researchers have explored LLMs for prediction, generation and understanding tasks.
- The need for fine-tuning is task-dependent. It is very important for zero-shot settings.

Future Directions…

- Fairness and Bias
- Hallucination
- More benchmarks
- Model interpretability
- Fine-tuning strategies
- Tabular data challenges: Sparsity, correlation, etc.

Disclaimer: My own opinions…

A modeller's guide

- How should my table be **serialized**?
- Do I need **data manipulation**? Do I need to reduce my data size? If so, how do I select relevant information? Do I have access to additional information? Should I perturb my data to check for robustness?
- Which **pre-trained LLM** should I use? Should I **fine-tune** an (L)LM for my task?
- How should my **prompts be formatted**? Can I employ in-context learning/ CoT/ role-play strategies?
- Does my interaction with the LLM agent need to be multi-turn? In my application, do **end-to-end workflows** make sense?
- Other considerations: Fairness and bias, cost, speed, accessibility, etc.

Disclaimer: My own opinions...

When I would use LLMs for Tabular Data

- Limited data
- Quick prototyping
- Sparsity in meaningful categorical columns
- QA, Text-to-SQL, End-to-end applications

- All numeric data + Column names are not meaningful
- Too many class labels
- No room for error in output
- If traditional methods are good enough

Questions & Feedback?

fiona.anting.tan@gmail.com

Additional Slides

Self-consistency

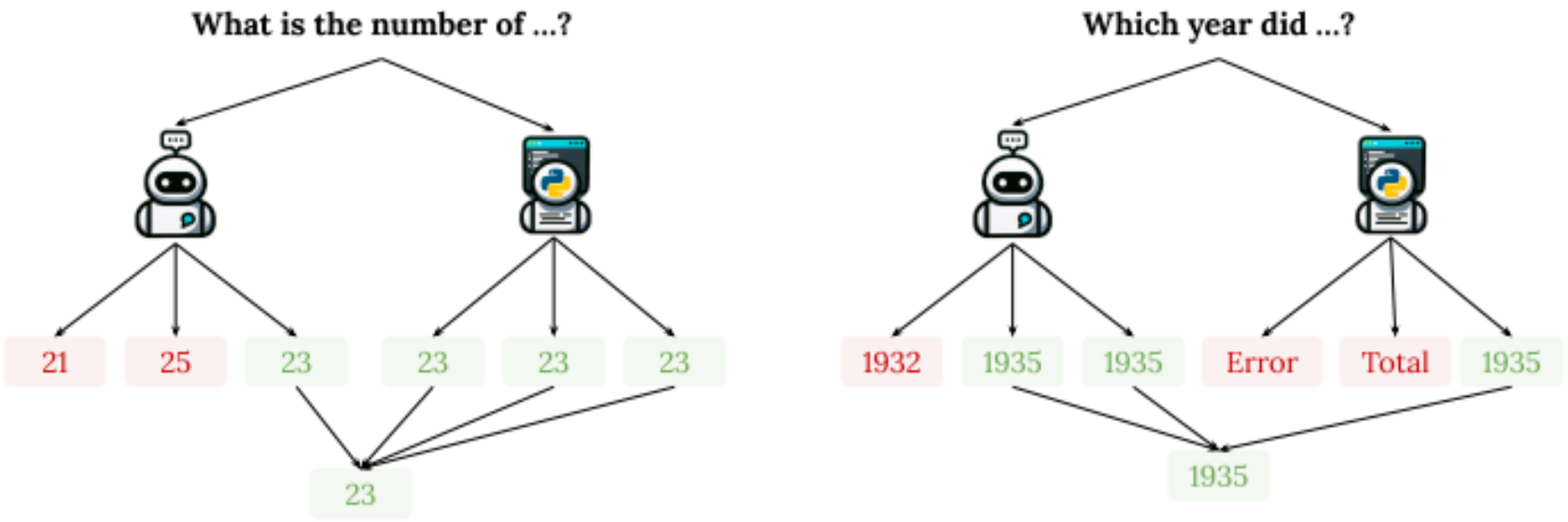


Figure 21: An illustration of *Mix Self-Consistency* by aggreagting outputs from multiple reasoning methods to form a unified, high-confidence prediction..

https://arxiv.org/pdf/2312.16702v1