DataFrame QA: A Universal LLM Framework on DataFrame Question Answering Without Data Exposure

Junyi Ye, Mengnan Du and Guiling Wang

New Jersey Institute of Technology, USA

December 5, 2024

Current LLM-Based Table QA Pipeline

Instruction: Using the table provided, answer the following questions based on the data.

Table:

Product, Price, Quantity Available, Rating
Laptop,$1,000,15,4.5
Smartphone,$800,30,4.7
Tablet,$400,25,4.3
Headphones,$200,50,4.1
Monitor,$300,20,4.2

Question: What is the cheapest product available in the table?

Figure 1: Typical prompt for table QA.

Limitations of current LLM-Based Table QA pipeline:

- LLM's Context Window Limits.
- Inefficient Token Use & High Costs.
- Data Leakage Risk.
- Decreased Performance with Long Prompts.
- Weak Mathematical/Data Analysis Capabilities.
- Processing Unnecessary Data.
- Table miss alignment.

Introduction

- This paper introduces **DataFrame QA**, a novel task that utilizes **LLMs** to generate Pandas queries for information retrieval and data analysis on dataframes, emphasizing safe and non-revealing data handling.
- Emphasizes **data privacy** by relying on dataframe column names and types, not values.
- Addresses limitations of existing LLM-based data analysis methods.

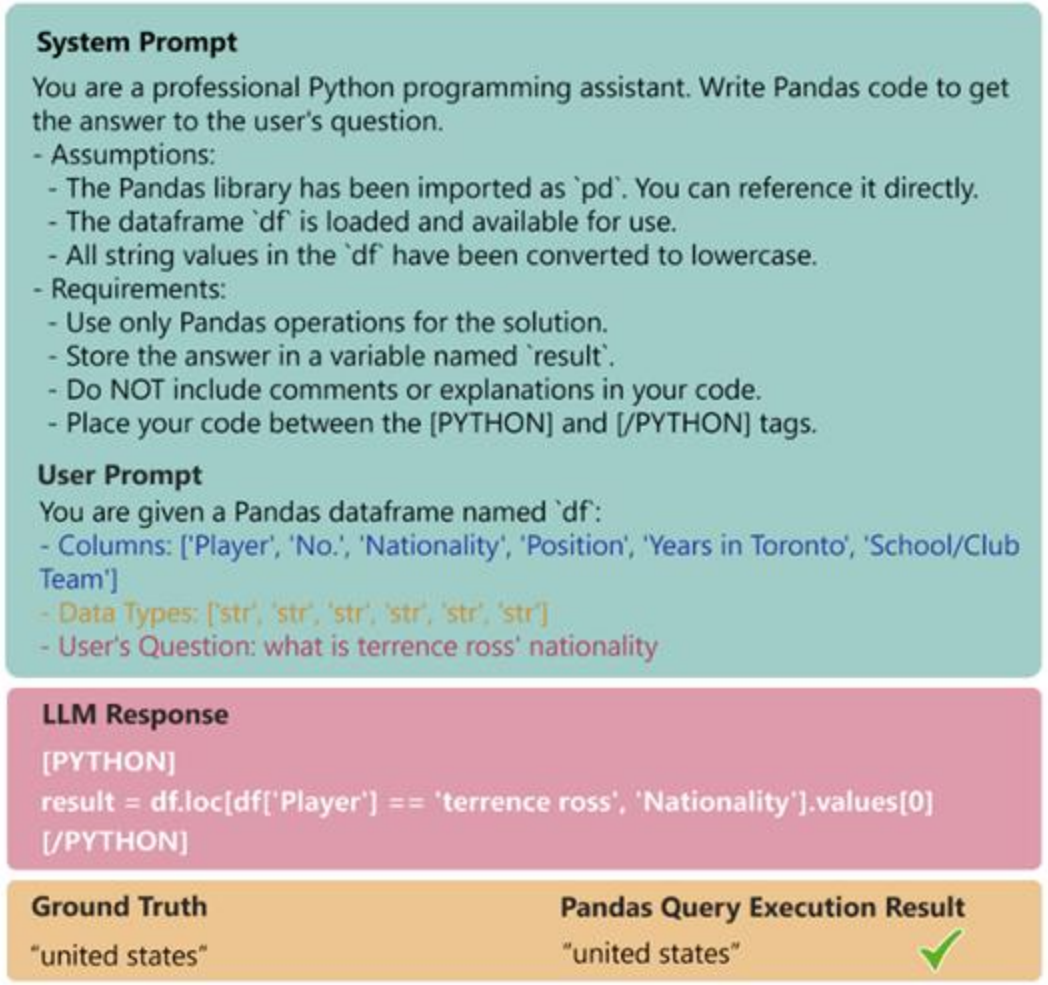

Figure 2: An example where a LLM can generate a correct Pandas query to answer user question.

Methodology

Task: Generate executable Pandas quires for answering user's questions on dataframes.

Focus: Use column headers and metadata, ensuring **data security**.

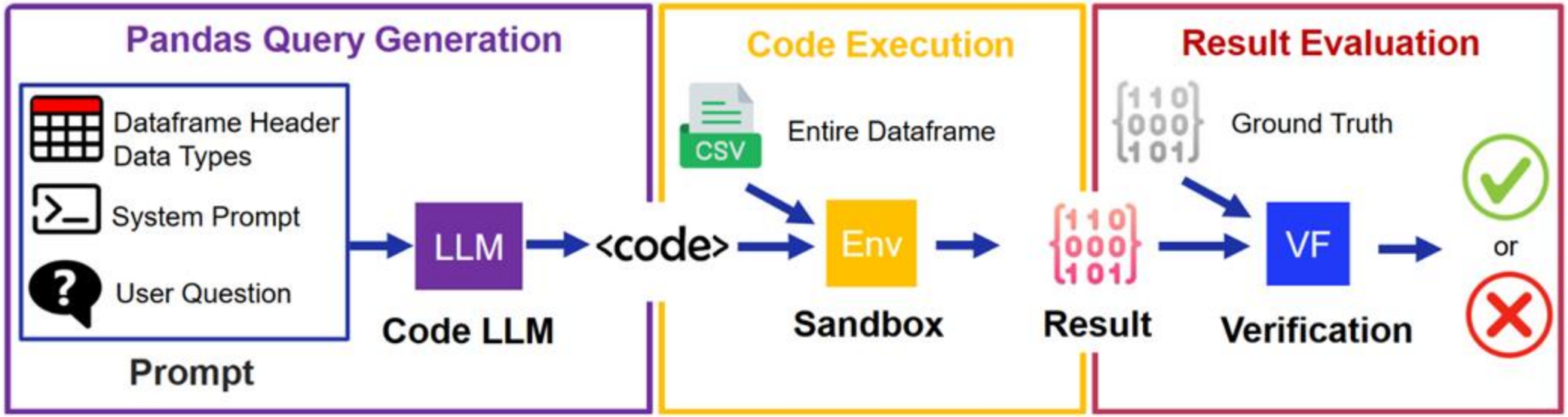

Figure 3: Framework of DataFrame QA. Note that, LLM in the figure can be replaced with any fine-tuned NLP model trained for the DataFrame QA task.

Datasets

1. WikiSQL (Simple Query Dataset) [1]

- ❖ Adapted for DataFrame QA, transforming tables into dataframes.
- ❖ Question types: 71% simple retrieval, 29% aggregation tasks.

2. UCI-DataFrameQA (Complex Dataset)

- ❖ Develop a DataFrame QA dataset reflecting **real-world scenarios** using GPT4.
- ❖ Data sourced from the **UCI dataset** [2], spanning various domains.
- ❖ Represent three real-life data interaction roles: **(1)Data Scientist; (2)General Users; (3)Data Owners**

User Question	Pandas Query	Types
which province is bay of islands in?	`result = df.loc[df['Electorate']=='bay of islands', 'Province'].iloc[0]`	Retrieval
how many combined days did go shiozaki have?	`result = df.loc[df['Wrestler']=='go shiozaki', 'Combined days'].values[0]`	Aggregation
how does the average shell weight vary across different numbers of rings?	`result = df.groupby('Rings')['Shell_weight'].mean()`	Data Analysis
can you create a new column 'volume' as a product of length, diameter, and height, then find the average volume for each sex?	`df['Volume'] = df['Length'] * df['Diameter'] * df['Height']` `result = df.groupby('Sex')['Volume'].mean()`	Data Analysis

Table 1: Examples of Sample Questions and Corresponding Pandas Queries.

Experiment and Results

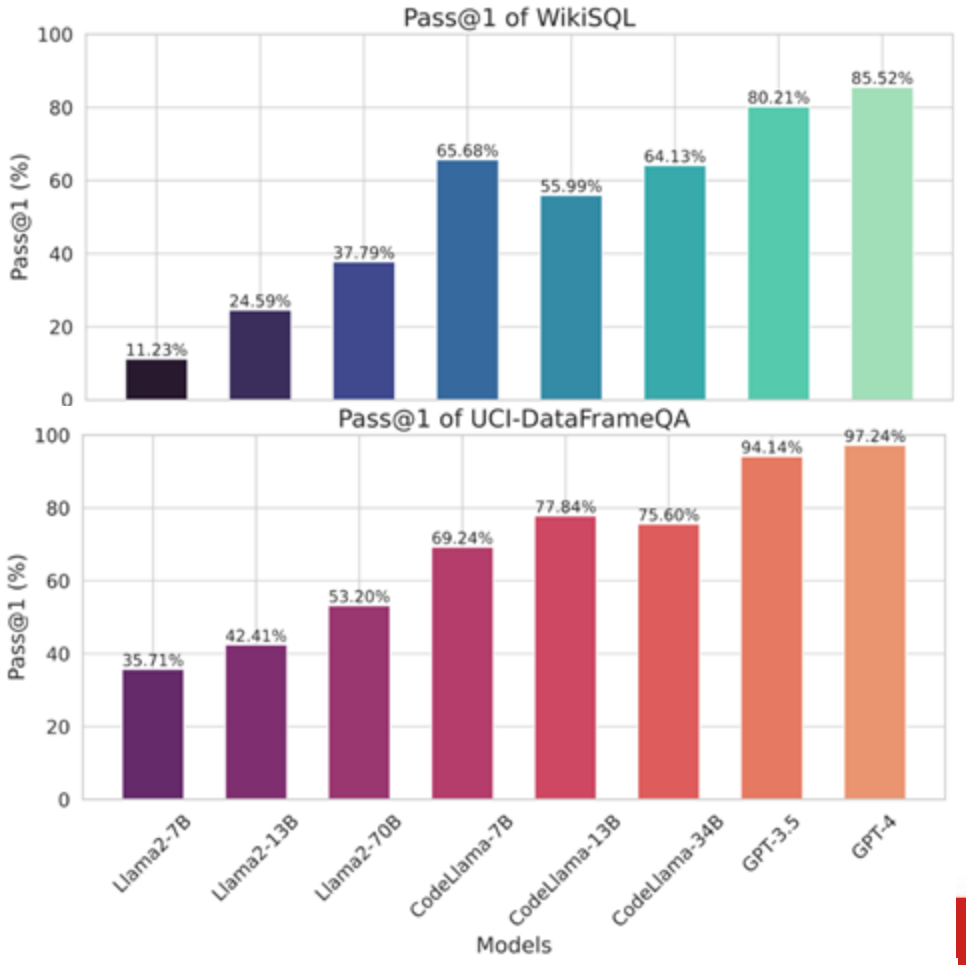

Figure 4: Performance of LLMs on Two Datasets.

Findings:

- **GPT-4 Dominance:** GPT-4 demonstrated exceptional performance on both datasets, achieving high accuracy.
- **Scaling Laws [3]:** Larger models generally perform better, but CodeLlama models are an exception.
- **Comparison to Text-to-SQL:** GPT-4's zero-shot approach and the complexity of Pandas queries slightly hinder its performance compared to specialized Text-to-SQL models.

Error Analysis

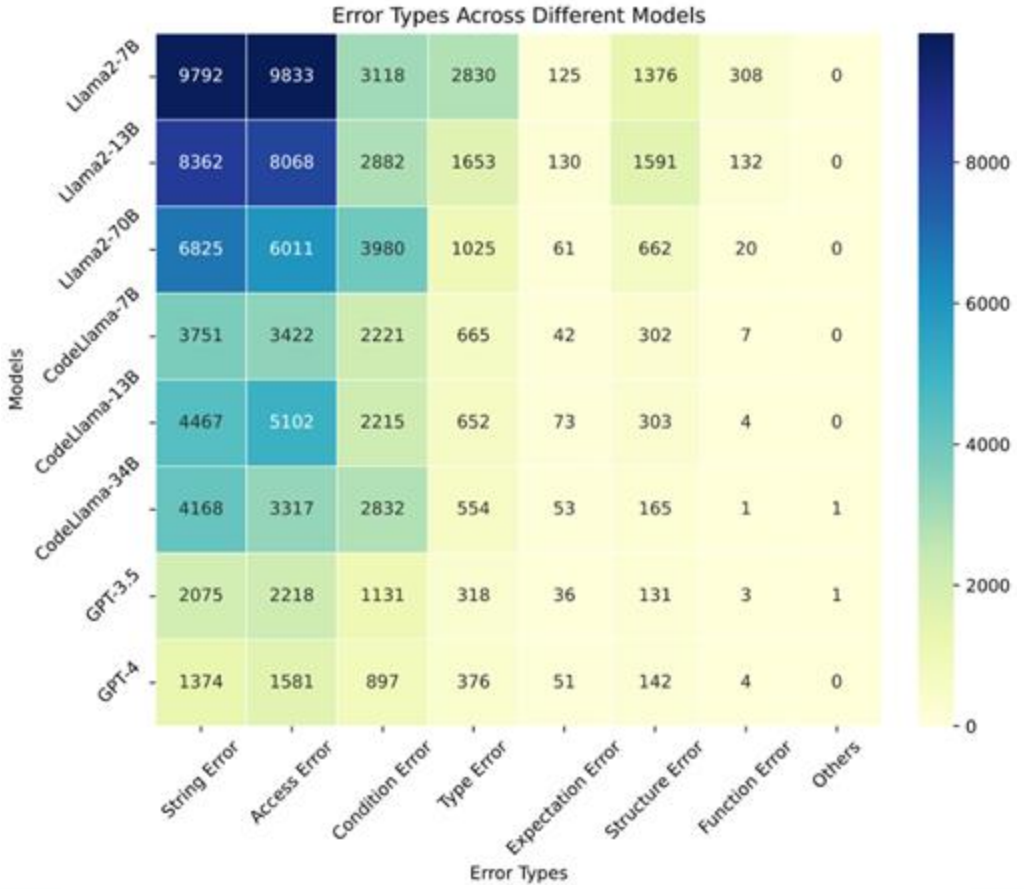

Figure 5: Distribution of Error Types Among Different LLMs on WikiSQL.

Error Type	Eample
Value Retrieval Error	Q : Which province is grey and bell electorate in? P' : `result = df[(df['Electorate']=='grey')|(df['Electorate']=='bell')]['Province'].unique()` P : `result=df[(df['Electorate']=='grey and bell')]['Province'].unique()`
Column Reference Error	Q : What is the variance in resting blood pressure (trestbps) among different heart disease severity levels? P' : `df.groupby('ca')['trestbps'].var()` P : `df.groupby('num')['trestbps'].var()`

Instruction Misalign-ment	Q : What are the mean and median lengths of abalone by each sex category?
	P' : `import pandas as pd`
	`# Group the dataframe by sex`
	`...`
	P : `result = df.groupby('Sex')['Length'].agg(['mean', 'median'])`

Aggregation Error	Q : what is the total amount of allied-unrelated where the component is human capital? P' `result = df[df['Component']=='human capital']['Allied-Unrelated'].sum()` P `result = df[df['Component']=='human capital']['Allied-Unrelated'].count()`
Function-Column Ambiguity Error	Q : what is the average score when the swimsuit score is 8.503? P' : `result = df[df['Swimsuit']=='8.503']['Average'].mean()` P : `result = df[df['Swimsuit']=='8.503']['Average']`
Insufficient Column Data/For-mat Information	Q : which team played on december 5? P' `df['Date'] = pd.to_datetime(df['Date'])` `result = df[df['Date'] == 'december 5']['Team']` P : `result = df[df['Date'] == 'december 5']['Team']`
Coding Syntax Error	Q : how does the average shell weight vary across different numbers of rings? P' : `result = df.groupby('Rings').mean()['Shell_weight']` P : `result = df.groupby('Rings')['Shell_weight'].mean()`
Hallucination Error	Q : i'm interested in knowing the most common age of abalone. can you find that for me? P' : `result = df['Age'].mode()` P : `result = df['Rings'].mode()`

Table 2: Typical Failure Cases in DataFrame QA Task.

Conclusion

- We introduce **DataFrame QA**, **a secure, zero-shot LLM** framework that leverages dataframe headers to address **data privacy** and **minimize extraneous prompts**.
- By enriching prompts with dataset descriptions, the framework improves performance.
- The success of this task relies on both **coding abilities** and **query comprehension**, with GPT-4 achieving high practical accuracy.

Reference

1. Zhong, Victor, Caiming Xiong, and Richard Socher. "Seq2sql: Generating structured queries from natural language using reinforcement learning." *arXiv preprint arXiv:1709.00103* (2017).

1. Blake, Catherine L. "UCI repository of machine learning databases." http://www. ics. uci. edu/~ mlearn/MLRepository. html (1998).

1. Kaplan, Jared, et al. "Scaling laws for neural language models." arXiv preprint arXiv:2001.08361 (2020).

Thank you!
Q&A

GitHub