Goal-Driven Data Story, Narrations and Explanations

Aniya Aggarwal*, **Ankush Gupta***, **Shivangi Bithel***, **Arvind Agarwal***
IBM Research, India
{aniyaagg,ankushgupta,shivangibithel,arvagarw}@in.ibm.com

Abstract

In this paper, we propose a system designed to process and interpret vague, open-ended, and multi-line complex natural language queries, transforming them into coherent, actionable data stories. Our system's modular architecture comprises five components—Question Generation, Answer Generation, NLG/Chart Generation, Chart2Text, and Story Representation—each utilizing LLMs to transform data into human-readable narratives and visualizations. Unlike existing tools, our system uniquely addresses the ambiguity of vague, multi-line queries, setting a new benchmark in data storytelling by tackling complexities no existing system comprehensively handles. Our system is cost-effective, which uses open-source models without extra training and emphasizes transparency by showcasing end-to-end processing and intermediate outputs. This enhances explainability, builds user trust, and clarifies the data story generation process.

1 Introduction

Business intelligence (BI) is critical for enterprise decision-making across functions like sales, HR, and IT. Traditionally, BI relied on static dashboards, manually crafted SQL queries, and complex labor-intensive workflows that were effective but rigid and required technical expertise, limiting in-depth or exploratory analysis. The advent of large language models (LLMs) has transformed BI, raising user expectations for systems that process natural language, handle numerical data, and address complex, multi-faceted queries with intuitive, narrative insights aligned with business goals. While AI and LLMs have been integrated into BI systems, they have primarily handled simpler queries. Modern BI users now demand more sophisticated systems capable of interpreting intricate natural language requirements and providing comprehensive, engaging, and easily understandable answers supported by visual analytics. This growing demand highlights the need for solutions that bridge the gap between complex data analysis and human interpretability, enabling seamless communication of insights without technical expertise (Cxtoday, 2024). We term these insights or narratives *Data Stories*.

Data Storytelling merges data analysis, visualization, and qualitative insights into a unified narrative that highlights the broader significance of data (Knaflic, 2015). Unlike conventional business intelligence, it focuses on aligning insights with business objectives and user context. By leveraging visual aids and addressing various learning styles, this approach enhances comprehension, making complex data more accessible and engaging.

Large Language Models (LLMs) have demonstrated near-human performance in text-based applications, but their capabilities in handling numerical data, complex reasoning, and domain-specific queries remain limited. Existing approaches, such as Text-to-SQL (Yu et al., 2018; Zhong et al., 2017) and Table QA (Chen et al., 2020; Wang et al., 2023; Nan et al., 2022; He et al., 2024) provide only partial solutions to the needs of BI users. These methods often struggle with vague, multi-line queries that require nuanced understanding, advanced numerical reasoning, and the seamless integration of interconnected insights (Figure 1). While Table QA methods can handle more complex queries, they are not scalable to enterprise dataset because of their dependency on input data in prompt. Innovations like the Agentic Framework (Islam et al., 2024) show promise in addressing some of these challenges, but they heavily depend on the design and functionality of underlying tools. As a result, these frameworks often fall short in generating detailed, comprehensive narratives enriched with visual analytics, without requiring significant effort in constructing and optimizing the tool pipeline.

To address these challenges, we propose a goal-driven data story narration system that transforms

*All authors contributed equally to this work.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Industry Track), pages 684–694

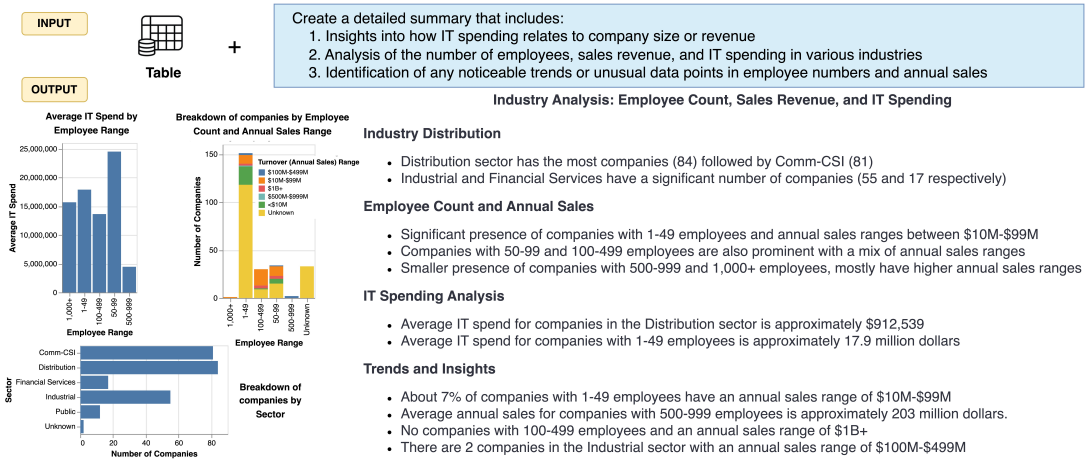

Figure 1: An example of Data Storytelling - a complex BI ask and its associated response generated by our system

vague, open-ended, multi-line queries into structured, coherent, and actionable data stories. It goes beyond existing approaches by offering a holistic framework to resolve ambiguous queries through systematic sub-query generation, extraction of relevant data insights, and seamless narrative presentation tailored to user intent. The system's modular architecture includes: Question Generation, which plans the narrative framework by formulating pivotal questions; Answer Generation, which provides reliable responses; NLG/Chart Generation, which translates insights into text or charts; and Summarization, which compiles the output into a coherent narrative. Each module operates synergistically to create human-readable, verifiable outputs.

Our system is distinct in its ability to handle vague, multi-line queries systematically, ensuring transparent, data-driven results. Through intermediate transparency and evidence-based storytelling, it fosters trust and usability. Its use of open-source models makes it cost-effective, scalable, and accessible to enterprises of all sizes while its modular architecture makes easy to integrate into existing BI solutions. We conduct a human evaluation focusing on relevance, readability and presentability metrics, and our system excels on all these metrics (Table 1). This demonstrates its effectiveness in addressing open-ended queries and meeting business intelligence needs.

2 Data Story Generation

The system is initiated when a user queries tabular data using a natural language utterance. This query is processed through a series of modules, as detailed in Figure 2, culminating in a comprehensive data story presented through text and infographics. These modules leverage LLMs and prompt engineering in a zero-shot setting, ensuring the pipeline's versatility across various domains without requiring fine-tuning. For reproducibility, the prompts used in our pipeline are provided in Appendix A.

2.1 Relevancy Check

The pipeline's initial module ensures query relevance to the provided tabular data, preventing unnecessary processing. For example, a query like *"Which films blend humor with tragedy in a way that changes audience perspectives?"* is irrelevant when querying *customer accounts* and should be flagged. Using an LLM, we check relevance by providing the data schema and user query. The LLM responds with "yes" for relevant queries and "no" for irrelevant ones, prompting users to rephrase if needed. Relevant queries proceed to the Question Generation module.

2.2 Question Generation

This module generates hierarchical questions to guide the data storytelling process, using an LLM based on the user's query and dataset. It operates in two phases. In the first phase, Level 1 Question are generated where the LLM identifies key dimensions from the user query and generates high-level questions related to these dimensions, using the query and dataset metadata. For instance, for the user query in Figure 3, the LLM identifies dimensions such as employee count, sales revenue, IT spending, companies and annual sales trends, and generates questions around those dimensions. This process uses prompt engineering in a zero-shot set-

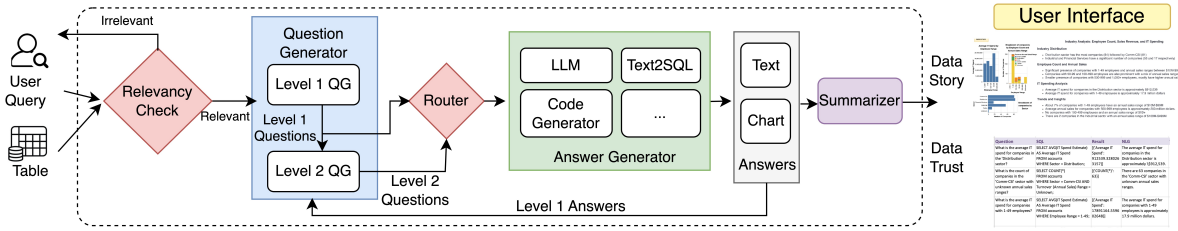

Figure 2: Proposed System Overview

ting. In the second phase, based on answers to Level 1 questions (by executing the pipeline), more detailed sub-questions (Level 2) are created to further explore the data. This drill-down approach enables a more thorough analysis of each key dimension identified in the previous phase, helping to reveal deeper insights and underlying causes. Such a detailed examination is crucial for constructing a comprehensive and meaningful data story (Figure 3). The question generation module ensures relevance, coherence, and engagement, producing questions that are answerable by text-to-SQL systems and contribute to a unified narrative.

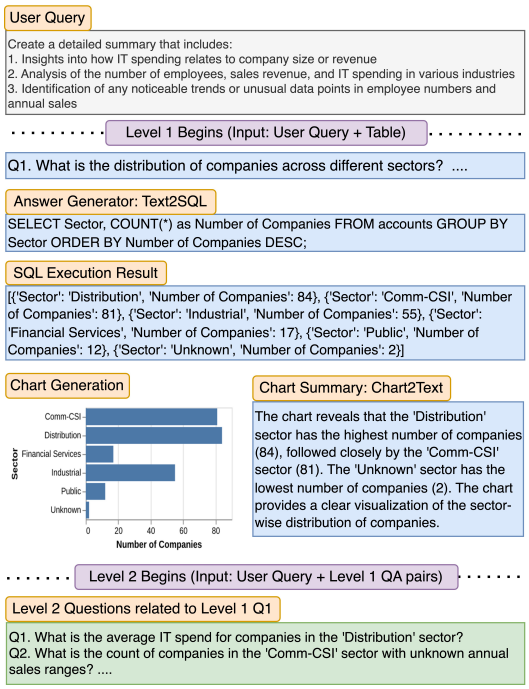

Figure 3: Intermediate Outputs and System Workflow

2.3 Answer Generation

This module handles both Level 1 and Level 2 questions by routing each to the most suitable answering agent, such as LLMs, Text2SQL, Multi-table SQL, or Interactive Python code, based on question type. This multi-agent approach ensures flexibility and future extensibility. Our router uses heuristics to select the appropriate agent, e.g., Text2SQL for analytical questions and LLM for open-ended ones. Our pipeline utilizes a Text2SQL [1] tool to generate SQL queries, retrieving relevant table schema from SQL databases using $SQLDatabase.get\_table\_info()$ method from Langchain's utilities (Utilities, 2024). The generated SQL query is then executed to obtain results, which is fed to the next module of the pipeline.

2.4 NLG/Chart Generation

Our pipeline employs two LLM-based tools for result generation: SQL2NLG and SQL2Chart. SQL2NLG translates SQL execution results into concise, factually accurate natural language summaries, handling smaller result sets. Whereas, SQL2Chart generates a Vega-lite v5 (Satyanarayan et al., 2017) JSON specification for visualizations, later converted into SVG format using vl-convert[2]. The LLM in SQL2Chart generates a visualization plan by identifying the most suitable chart type for the given data context, determining visual encodings (e.g., axes, colors, filters) for the selected chart, and suggesting a clear, descriptive title. In the final post-processing step, the specification is updated with the actual data for rendering.

2.5 Chart2Text

This component leverages the ReAct framework (Yao et al., 2023) and a custom insight generation tool to produce accurate and detailed chart summaries, enhancing the interpretability of data visualizations. While charts highlight trends, textual summaries provide essential context, explain nuances, and emphasize key findings. The tool ensures accuracy by extracting metrics like minimum/maximum values, outliers, and trends, avoiding hallucination - a common issue with LLMs

[1] https://github.com/deepseek-ai/DeepSeek-Coder
[2] https://github.com/vega/vl-convert

when dealing with complex mathematical computations and large datasets. ReAct's step-by-step reasoning enables meaningful and contextually relevant summaries, offering users a thorough understanding of the data.

2.6 Summarization

In the final stage, textual responses and chart interpretations are synthesized into a cohesive data story, combining narrative and visual elements to ensure clarity and coherence. Much like a skilled storyteller weaves narrative and imagery to captivate an audience, our approach combines textual and graphical elements to create a cohesive and engaging data story. An LLM-driven summarizer uses the user query and hierarchical facts from Level 1 and Level 2 questions to produce a structured summary with a title, key facts, and bulleted sub-facts. This approach preserves accuracy while enhancing readability and user comprehension.

2.7 Implementation

We use an open-source LLM named *meta-llama-3-70B-instruct* for Relevancy Check, Question Generation, NLG/Chart Generation, Chart2Text and Summarization modules. While another open-source model named *deepseek-coder-33b-instruct* is used for the Answer Generation (Text2SQL) module.

2.8 Story Representation

The user interface of our proposed system dynamically adjusts the size and positioning of images and textual insights based on the story plot, enhancing storytelling effectiveness. It also displays the end-to-end process and intermediate outputs from each module in a tabular format (Figure 4). Users can view detailed information via tooltips on hovering, fostering trust and improving explainability.

Question	SQL	Result	NLG
What is the average IT spend for companies in the 'Distribution' sector?	SELECT AVG(IT Spend Estimate) AS Average IT Spend FROM accounts WHERE Sector = Distribution;	[{'Average IT Spend': 912539.3280263157}]	The average IT spend for companies in the Distribution sector is approximately $912,539.
What is the count of companies in the 'Comm-CSI' sector with unknown annual sales ranges?	SELECT COUNT(*) FROM accounts WHERE Sector = Comm-CSI AND Turnover (Annual Sales) Range = Unknown;	[{'COUNT(*)': 63}]	There are 63 companies in the 'Comm-CSI' sector with unknown annual sales ranges.

Figure 4: Example results from intermediate modules

3 Evaluation

In order to measure the effectiveness of our approach, we use human evaluation, which is a critical component of our research for several reasons. Firstly, there is a lack of specialized labeled datasets for vague, multi-line queries that can effectively measure the nuanced capabilities of data storytelling systems. Secondly, the subjectivity inherent in evaluating the quality of data stories means that automated metrics alone cannot fully capture the relevance, readability, and presentation of the generated narratives. Traditional evaluation metrics often fail to address the qualitative aspects of human-centric tasks, such as the clarity and engagement of the produced content. Moreover, evaluating complex data storytelling systems requires metrics that go beyond mere technical accuracy, encompassing dimensions like user satisfaction and the practical utility of the generated stories. Existing metrics are frequently insufficient to gauge these subjective criteria effectively.

3.1 Human Evaluation

As the first system explicitly designed to handle vague, open-ended, and multi-line queries in the business intelligence domain, our work addresses challenges that existing solutions have not yet tackled. This novelty precludes the availability of established baselines for direct and comprehensive comparison. To evaluate the system's effectiveness, we employ a human-centered evaluation framework, focusing on metrics critical to data storytelling systems: relevance, readability, and presentability.

Despite the lack of directly comparable systems, we benchmark our approach against state-of-the-art solutions like OpenAI Code Interpreter (OpenAI code interpreter, 2023) and LangChain Pandas Agent (Langchain Pandas Dataframe Agent, 2023). These systems, while powerful within their respective scopes, are not explicitly designed for vague, multi-line queries. For the evaluation, we utilized the latest *gpt-4o-mini* model for both baselines, whereas our system leverages open-source models to ensure cost-effective and scalable deployment.

Four unbiased volunteers, each with over 7 years of industry experience in data science and analytics, have been recruited for this evaluation. Each participant is provided with five datasets and tasked with asking a total of 10 queries each within the application's scope. They evaluate the systems on three criteria: whether the story is (A) Relevant and Grounded, (B) Readable and Interesting, and (C) Presentable, using a 1 [Very Dissatisfied] to 5 [Very Satisfied] scale.

Datasets: We utilize a diverse array of five publicly available datasets to ensure a comprehensive evaluation of our approach. These datasets span

various domains, sizes, and user contexts, allowing us to assess the performance of our methods under different conditions and query types. The datasets include Customer Shopping Trends (3900 rows x 18 columns) (2024), Employee Attrition & Performance (1470 rows x 35 columns) (2024), Netflix Movies (8809 rows x 12 columns) (2024), Vehicle Sales (558837 rows x 16 columns) (2024), and Online Sales Data (240 rows x 9 columns) (2024).

Table 1: Our System vs Baseline Performance

	Our System	Pandas Agent	Code Interpreter
Relevant & Grounded	**4.18**	3.09	2.97
Readable & Interesting	**4.31**	2.81	2.64
Presentable	**4.28**	2.76	2.32

3.2 Analysis of Human Evaluation Results

As shown in Table 1, our system consistently excels in all three metrics, demonstrating its ability to provide responses that are not only relevant, grounded, interesting, and understandable, but also more presentable than both the baselines. Analysis of baseline outputs highlights key issues in current solutions: (A) Output Not Grounded - Not able to utilize the dataset and instead give a generic response to the user query based on just dataset schema (case of hallucination), (B) Giving too technical output making it difficult for the end user to understand, such as Code Interpreter returning “chi-squared test" details like statistics, p-value, degrees of freedom, expected frequencies, etc., (C) Tendency to generate answers focusing only on a specific part of the query.

3.3 Comparison of User Efforts: Traditional BI Tools vs. Proposed System

To highlight the advantages of our proposed system, a BI user was tasked with answering a sample multi-line query using traditional BI tools, which are designed for straightforward queries. As shown in Figure 5, the user had to manually construct each query, making the process time-consuming and inefficient. Many queries returned no meaningful results, while others generated excessive charts, leading to outputs that were not actionable. This required the user to iteratively refine queries, yet numerous key insights remained undiscovered. In contrast, our system seamlessly resolves the same multi-line query in a single step (Figure 1), showcasing its efficiency and ease of use.

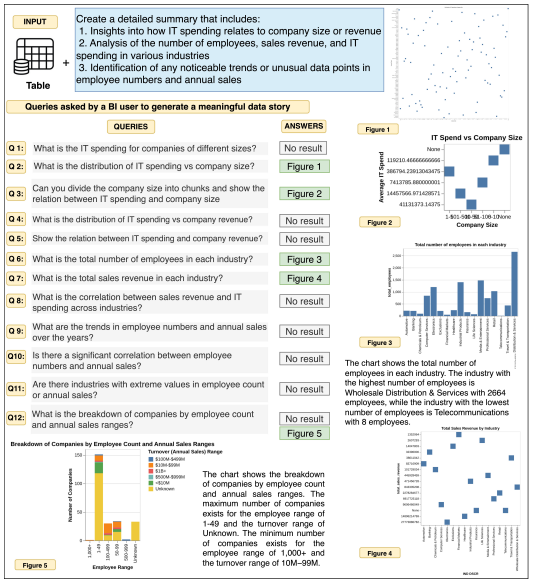

Figure 5: Steps Taken by BI User in Traditional BI tools

Comparison in Table 2 highlights the significant effort required by BI users when using traditional systems versus the seamless, efficient experience offered by our system. By automating the interpretation and analysis of complex queries, our approach bridges the gap between user intent and actionable insights.

4 Path to Deployment

Our system is designed for seamless integration, either within an existing BI system or as a standalone service for BI tasks. Its modular architecture ensures easy deployment and cost-effectiveness, leveraging open-source models. The deployment can be carried out in the following two ways.

Integration of entire pipeline with Existing BI Systems: The data story generation pipeline can be deployed as a streaming API that processes user queries and tabular data, producing narratives in incremental chunks. This approach mitigates long processing times by delivering partial data stories as they are generated, with continuous updates until completion. This strategy has been validated with an internal BI system that features a natural language query interface. In this system, our pipeline is integrated as an additional feature, accessible through an *Insights* tab that triggers the streaming API. The data story, including both text and charts, is displayed in chunks, providing an interactive and dynamic interface to enhance user comprehension.

Module-wise Deployment: Components like the Question Generator, Answer and Chart Genera-

Table 2: Comparison between Current BI Workflow and Our System

Aspect	Current BI Workflow	Our System
Query Breakdown	Requires manual decomposition into 10+ sub-queries.	Automatically interprets the multi-line query.
Analysis	Relies on user expertise to identify relationships and trends from raw data.	Generates relationships, trends, and insights directly.
Effort	High; user needs to frame queries, analyze intermediate results and refine queries iteratively.	Low; single query leads to complete, coherent narrative.
Output Presentation	Separate charts and tables require manual integration.	Unified narrative with integrated visuals and text.

tor, Summarizer, and Chart2Text can be deployed independently as API endpoints. This flexibility enables integration into existing pipelines to address specific sub-problems, with each module functioning as a black box with defined inputs and outputs.

In summary, the use of open-source models and a modular design offers the following advantages:

Cost-Effectiveness and Adaptability: Open-source LLMs in zero-shot settings significantly cut costs compared to proprietary solutions, offering scalability and accessibility. Emphasis on prompt design over fine-tuning enhances adaptability.

Flexibility and Scalability: The modular design allows for independent updates or replacements of components without affecting the entire system, enabling easy future upgrades and adaptations to accommodate evolving requirements.

5 Future Work and Research Challenges

Our system effectively addresses descriptive and, to some extent, diagnostic questions but has scope for growth in predictive and prescriptive analytics. Expanding into these areas will enable forecasting and actionable recommendations, enhancing its utility. Key challenges include integrating advanced forecasting techniques, designing recommendation algorithms, and addressing ethical concerns. Additionally, building a comprehensive benchmark dataset will be crucial for evaluating system performance. Such a dataset would provide a standardized framework for future research, enabling validation of data storytelling approaches and facilitating comparisons with other methods. Furthermore, developing an automatic evaluation system to replace the time-consuming human evaluation process will ensure a more scalable, consistent, and efficient assessment of system performance.

6 Related Work

Addressing complex, open-ended queries over tabular data has spurred research in NLP, database management, and data visualization. This section reviews progress in text-to-SQL, data interpretation, and narrative generation systems.

Text-to-SQL Systems enable non-technical users to query data by translating natural language into SQL. Early systems like Seq2SQL (Zhong et al., 2017) and Spider (Yu et al., 2018) focused on query translation. Recent transformer-based models handle more complex queries but often lack the ability to generate actionable insights, particularly for enterprise-specific open-ended queries.

Tabular Question Answering Systems answer queries directly from tables (Chen et al., 2020; Wang et al., 2023; Nan et al., 2022; He et al., 2024). While these systems perform complex reasoning, they suffer from limited accuracy due to reliance on LLMs and context length constraints, reducing their effectiveness for large datasets.

Insights Extraction Systems, such as InsightPilot (Ma et al., 2023) and JarviX (Liu et al., 2023), focus on extracting insights from data. InsightPilot aligns insights with specific goals, while JarviX combines AutoML tools for summaries and visualizations. Systems like LLM4Vis (Beasley and Abouzied, 2024) and QUIS (Manatkar et al., 2024) create visualizations and exploratory insights. However, these focus on isolated insights rather than cohesive data narratives.

Data Story Systems combine insights with narrative generation but often rely on LLMs, limiting scalability. For instance, DataNarrative (Islam et al., 2024) uses multi-agent systems to generate stories but struggles with large datasets. In contrast, our system employs deterministic SQL execution for precise computations and meaningful narratives. Related works also include data-driven storytelling from notebooks (Zheng et al., 2022), articles (Sultanum and Srinivasan, 2023), and autonomous agents in Data-Copilot (Zhang et al., 2024). Most existing systems, including DataNarrative, are not open-source, hindering direct comparisons. Furthermore, their benchmark datasets, often using small tables, fail to evaluate the scalability of our system effectively.

7 Conclusion

In this paper, we propose a first-of-its-kind system to address vague, multi-line queries by integrating natural language processing with data analysis to generate comprehensive and interpretable data stories. Our system prioritizes adaptability and transparency, offering a dynamic interface that adjusts content presentation and provides insights into the processing pipeline. This design enhances storytelling effectiveness while building user trust through explainability and access to intermediate outputs. By combining state-of-the-art LLMs with practical design considerations, our system marks a significant advancement in data storytelling, delivering a robust tool for generating actionable and understandable insights.

8 Limitations

While our system effectively generates insightful data stories in response to user queries, a few limitations warrant consideration:

Processing Time : Although our system is designed to handle large datasets and broad or open-ended queries, processing times may increase in certain cases. Complex analyses or large datasets can slow down response times, potentially affecting the overall user experience.

Ambiguity in Query Interpretation : Open-ended or vague queries can be interpreted in multiple ways. As a result, our system might not always accurately discern the user's intent, which can lead to less relevant or incomplete answers.

Dependence on Data Quality : Our system's performance is closely tied to the quality, structure, and completeness of the input data. Inconsistent or missing data can result in unreliable insights or errors.

Ethical and Legal Risks : Analyzing open-ended queries on enterprise or sensitive datasets may unintentionally reveal patterns or insights with ethical or legal implications, such as biases or privacy concerns.

Adherence to LLM Token Limits : Our system, which heavily relies on LLMs, must adhere to the strict token limits imposed by the models. As a result, datasets with large schemas may encounter limitations or performance issues.

References

Cole Beasley and Azza Abouzied. 2024. Pipe(line) dreams: Fully automated end-to-end analysis and visualization. In *Proceedings of the 2024 Workshop on Human-In-the-Loop Data Analytics*, HILDA 24, page 1–7, New York, NY, USA. Association for Computing Machinery.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

Cxtoday. 2024. Gartner Magic Quadrant for Analytics and Business Intelligence (ABI) Platforms 2024. Accessed on: Nov 29, 2024.

Xinyi He, Mengyu Zhou, Xinrun Xu, Xiaojun Ma, Rui Ding, Lun Du, Yan Gao, Ran Jia, Xu Chen, Shi Han, Zejian Yuan, and Dongmei Zhang. 2024. Text2analysis: A benchmark of table question answering with advanced data analysis and unclear queries. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):18206–18215.

Mohammed Saidul Islam, Md Tahmid Rahman Laskar, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. 2024. DataNarrative: Automated data-driven storytelling with visualizations and texts. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19253–19286, Miami, Florida, USA. Association for Computational Linguistics.

Cole Nussbaumer Knaflic. 2015. *Storytelling with data: A data visualization guide for business professionals*. John Wiley & Sons.

Langchain Pandas Dataframe Agent. 2023. Accessed on: Nov 20, 2024. [link].

Shang-Ching Liu, ShengKun Wang, Tsungyao Chang, Wenqi Lin, Chung-Wei Hsiung, Yi-Chen Hsieh, Yu-Ping Cheng, Sian-Hong Luo, and Jianwei Zhang. 2023. Jarvix: A llm no code platform for tabular data analysis and optimization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 622–630.

Pingchuan Ma, Rui Ding, Shuai Wang, Shi Han, and Dongmei Zhang. 2023. Insightpilot: An llm-empowered automated data exploration system. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 346–352.

Abhijit Manatkar, Ashlesha Akella, Parthivi Gupta, and Krishnasuri Narayanam. 2024. QUIS: Question-guided insights generation for automated exploratory data analysis. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language*

Processing: Industry Track, pages 1523–1535, Miami, Florida, US. Association for Computational Linguistics.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, et al. 2022. Fetaqa: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

OpenAI code interpreter. 2023. Accessed on: Nov 20, 2024. [link].

Pavan Shubhash. 2024. IBM HR Analytics Employee Attrition & Performance. Kaggle, Accessed on: Nov 20, 2024.

Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2017. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization & Computer Graphics (Proc. InfoVis)*.

Shivam B. 2024. Netflix Movies and TV Shows. Kaggle, Accessed on: Nov 20, 2024.

Shreyansh Verma. 2024. Online sales dataset - popular marketplace data. Kaggle, Accessed on: Nov 20, 2024.

Sourav Banerjee. 2024. Customer Shopping Trends Dataset. Kaggle, Accessed on: Nov 20, 2024.

Nicole Sultanum and Arjun Srinivasan. 2023. Datatales: Investigating the use of large language models for authoring data-driven articles. In *2023 IEEE Visualization and Visual Analytics (VIS)*, pages 231–235. IEEE.

Syed Anwar. 2024. Vehicle Sales Data. Kaggle, Accessed on: Nov 20, 2024.

Langchain Utilities. 2024. Accessed on: Nov 20, 2024. [link].

Dingzirui Wang, Longxu Dou, and Wanxiang Che. 2023. A survey on table-and-text hybridqa: Concepts, methods, challenges and future directions. *Preprint*, arXiv:2212.13465.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Wenqi Zhang, Yongliang Shen, Weiming Lu, and Yueting Zhuang. 2024. Data-copilot: Bridging billions of data and humans with autonomous workflow. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Chengbo Zheng, Dakuo Wang, April Yi Wang, and Xiaojuan Ma. 2022. Telling stories from computational notebooks: Ai-assisted presentation slides creation for presenting data science work. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–20.

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

A Appendix

Relevancy Check Prompt

Check if the given query can be answered using the given data having the following columns. Answerability can be judged by checking if all the
1. columns required to answer the query exist in <Columns></Columns>, or
2. column values required to answer the query exist in <PossibleValues></PossibleValues>
for the given dataset.

Do not assume the domain of the data while answering.

If the query is a generic conversation statement such as a greeting statement which doesn't require the input data to answer the given query, answer "no".

Answer "yes" only if the query can be answered solely based on the given data, otherwise "no".

<Query> $utterance$ </Query>
<Columns> col_1, col_2, ... </Columns>
<PossibleValues>
col_1: $[val_{11}, val_{12}, ...]$
col_2: $[val_{21}, val_{22}, ...]$ ...
</PossibleValues>

Do not assume any other information. Do not generate any extra information.
<answer>

Primary Question Generation Prompt

Given the following user query: $user\ query$ and the dataset schema provided below, generate a set of high-level questions that are broad in scope and provide an overview of the key aspects related to the query. These questions should be answerable using text-to-SQL queries and should focus on the most important and relevant columns in the dataset. The questions should help in understanding the general patterns, trends, and summaries related to the user query. The dataset schema is as follows:

$possible\ values$

* Generate only top 4 questions and no other explanation.
* Make sure the generated questions are not composite and can be answered by text-to-sql.
* Generated questions should provide an overview of the key aspects related to the query.
* Output only the generated questions and enclose them in <question></question> tags.
* Ensure all questions can be formulated into valid SQL queries using the provided dataset schema.

Secondary Question Generation Prompt

Based on the answers to the following questions from Batch 1, generate a second set of questions that delve deeper into the data. These questions should build upon the previous answers and focus on identifying specific patterns, relationships, or anomalies within the data. They should aim to explain why certain trends or patterns were observed in the first set and explore deeper connections between the columns. The dataset schema remains the same:

$possible\ values$

First set of question-answer pairs:

$Question\ and\ answer\ pairs\ from\ Level\ 1$

* Each generated question is enclosed in <question></question> tags.
* Make sure the generated questions are not composite and can be answered by text-to-sql.
* Generate question along with previous batch question number information and no other explanation in the following format -

Q[] (related to Q[]) :`<question>`Generated Question Here`</question>`

Text2SQL Prompt

Task
Generate a SQL query to answer the following question:
`$question$`
Database Schema
This query will run on a database whose schema is represented in this string:
$table\_info$
SQL
Given the database schema, generate a single table based SQL query from the given schema that answers `$question$`:
```sql

**SQL2NLG Prompt**

For the given input context, translate the following data in an appropriate natural language based response. The generated natural language based response should be crisp and short and free of its source information. The generated sentences should be complete with context. Do not explain the data. Ensure that the generated text is supported by the given data.

Context: $question$
Data: $sql\_execution\_results$

Response:
```

SQL2Chart Prompt

< |system| >
You are a helpful assistant highly skilled in recommending and identifying relevant chart type and its associated encodings for visualisations.
< |user| >
Your task is to recommend and generate a visualisation plan based on the given table description and question. Table Description contains a list of n column names with their nature and data type specified alongside. Let's think step by step.
Step 1
Identify the best suited chart type to present the question on the given table description with n columns. Follow the best visualisation practices to suggest an appropriate chart.
Step 2
Identify the required visual encodings related to the chart type identified in previous step to plot the given table description. Use only the given exact column names in table description to specify these encodings.
Step 3
If any information regarding axes variables or color to be used in the chart is available in the input question, use that in the visual encoding. Otherwise, do not specify the unknowns in the specification.
Step 4
Draft a suitable title for the visualisation clearing stating its purpose.
Step 5
Generate the Vega-lite 5 specification in JSON format using the title, encodings, color (if available) found in previous steps.

Do not assume any other information. Generate only the JSON specification. Do not generate any extra or new information. Do not explain the intermediate steps.

Table Description
$table\_context$
Question
$question$
JSON Specification

Chart2Text Prompt

Respond to the human as helpfully and accurately as possible. You have access to the following tools:
{$tools$}
Use a json blob to specify a tool by providing an action key $(tool\ name)$ and an action_input key $(tool\ input)$.
Valid "action" values: "Final Answer" or {tool_names}
Provide only ONE action per $JSON_BLOB, as shown:
```
{{
"action": $TOOL_NAME,
"action_input": $INPUT
}}
```
Follow this format:
Question: input question to answer
Thought: consider previous and subsequent steps
Action:
```
$JSON_BLOB
```
Observation: action result
... $($repeat Thought/Action/Observation maximum 2 times$)$
Thought: I know what to respond
Action:
```
{{
"action": "Final Answer",
"action_input": "Final response to human"
}}
Begin! Reminder to ALWAYS respond with a valid json blob of a single action.
Respond directly if appropriate. Format is Action: ``` $JSON_BLOB ``` then Observation
```

Summarizer Prompt

< |user| >

You are a helpful assistant highly skilled in summarising text in a well-structured format. Your task is to write a concise, fluent, and accurate summary based on the given query and a list of query-relevant facts. The generated summary should contain a list of high level topics, each followed by the related sub-topics. Every topic should have a relevant header with a listed short and concise describing text and sub-topics next to it. The input set of facts contain high level topics in <topic></topic> and the related sub-topical texts in <subtopic></subtopic>. Rearrange and present facts to form a cohesive summary containing a minimum of 5 words but not exceeding 500 words in length. Generate an apt title for the generated summary. Make sure not to miss any important fact from the summary. Do not add any extra facts or information not present in the query-relevant facts. Do not provide any further explanation.

Query: $utterance$

Facts:
<topic>
$primary\_fact_1$
<subtopic>
* $secondary\_fact_{11}$
* $secondary\_fact_{12}$
* ...
</subtopic>
</topic>
<topic>
$primary\_fact_2$
<subtopic>
* $secondary\_fact_{21}$
* $secondary\_fact_{22}$
* ...
</subtopic>
</topic>
...

Summary: