BI-Bench : A Comprehensive Benchmark Dataset and Unsupervised Evaluation for BI Systems

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

Abstract

A comprehensive benchmark is crucial for evaluating automated Business Intelligence (BI) systems and their real-world effectiveness. We propose **BI-Bench**, a holistic, end-to-end benchmarking framework that assesses BI systems based on the quality, relevance, and depth of insights. It categorizes queries into descriptive, diagnostic, predictive, and prescriptive types, aligning with practical BI needs. Our fully automated approach enables custom benchmark generation tailored to specific datasets. Additionally, we introduce an automated evaluation mechanism within BI-Bench that removes reliance on strict ground truth, ensuring scalable and adaptable assessments. By addressing key limitations, it offers a flexible and robust, user-centered methodology for advancing next-generation BI systems.

1 Introduction

AI has significantly advanced the development of automated BI systems, particularly with LLMs and automated code generation. Although there has been significant progress in system development and related areas like Text-to-SQL, a unified benchmark for evaluating overall system performance is still lacking. Existing benchmarks focus on isolated components like Text-to-SQL (Zhong et al., 2017; Yu et al., 2018; Lei et al., 2025), NLG from tabular data (Mahapatra et al., 2016; Chen et al., 2020), or table-based QA (Pasupat and Liang, 2015; He et al., 2023; Ashury-Tahan et al., 2025), providing fragmented benchmarks (Hu et al., 2024) rather than a holistic evaluation.

Although some end-to-end benchmarks exist (Islam et al., 2024; Zhang et al., 2025; Yang et al., 2024), they have a few key limitations. First, they rely on fixed question sets designed to evaluate specific systems and approaches rather than addressing the broader needs of BI users. Second, their rigid evaluation methods depend on strict ground truth, limiting adaptability to new datasets and business contexts. This highlights a critical gap—the absence of a widely accepted benchmark that effectively evaluates BI systems in real-world, user-driven scenarios.

To address this, we propose **BI-Bench**, a *holistic benchmarking framework* that places BI users and their expectations at the core of the evaluation process and is agnostic to the underlying system or approach. It is designed to capture the full spectrum of user needs, from simple to complex queries, across both straightforward and intricate data structures. More specifically, it covers four broad categories of BI queries i.e., descriptive, diagnostic, predictive and prescriptive, aligning with real-world BI requirements.

We release a benchmark dataset and evaluation mechanism, ensuring compatibility with existing BI systems for direct comparison and easy integration with ongoing research. Additionally, BI-Bench features a dynamic benchmark generation pipeline that is generic and automated, allowing users to create custom benchmarks tailored to their datasets. It leverages multiple LLMs to generate questions and metadata, complemented by an efficient automatic verification step that significantly reduces the reliance on human validation, allowing enterprises to tailor datasets to their specific needs.

In addition to data generation, BI-Bench also includes an automated evaluation mechanism that assesses BI system's output across multiple dimensions without relying on predefined ground truth. Our framework evaluates factual correctness, answerability, relevance, and presentation, ensuring a comprehensive performance assessment. At its core is a student-teacher framework, where an expert system (the teacher) evaluates a BI system's output (the student). Since obtaining a perfect expert system is impractical, we introduce the notion of weak experts to assess the student's output in a step-by-step manner, making factual correctness

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 1287–1299
July 28-30, 2025 ©2025 Association for Computational Linguistics

verification more feasible. Other dimensions — answerability, relevance, and presentation — are assessed using an LLM-as-a-judge approach.

A key strength of our framework, both in dataset creation and evaluation, is its full automation and scalability across diverse datasets and BI applications. To ensure reliability, we validated both the benchmark and evaluation methodology with human experts. Dataset validation confirmed its effectiveness with minimal filtering in the final stage, while our evaluation approach demonstrated strong performance. By publicly releasing BI-Bench[1], including both the dataset and evaluation method, we aim to advance research in the BI space and provide a robust foundation for future developments.

2 Related Work

The rapid advancement of BI systems has led to the emergence of several benchmark datasets designed to evaluate different system components. However, existing benchmarks remain fragmented, focusing on isolated tasks rather than addressing the full spectrum of BI user needs.

Text-to-SQL benchmarks (Zhong et al., 2017; Yu et al., 2018; Lei et al., 2025) assess a system's ability to generate SQL queries but do not evaluate whether the final outputs effectively serve BI users.

Code generation benchmarks such as DataSciBench (Zhang et al., 2025), InfiAgent-DABench (Hu et al., 2024), and Text2Analysis (He et al., 2023) focus on generating executable code for tasks like data cleaning and reporting. However, they rely on metrics like executable code ratio and pass rate, which do not directly measure correctness, answerability, or utility of the output.

Data story benchmarks like DataTales (Yang et al., 2024) assess BI narratives based on factuality, insightfulness, and style, utilizing a semi-automated Named Entity Recognition (NER)-based approach for fact-checking. DataNarrative (Islam et al., 2024), on the other hand, extends the evaluation to structured multi-paragraph stories with visualization components, focusing on informativeness, coherence, visualization quality, narrative quality, and factual correctness, all assessed by an LLM-based evaluator agent. However, LLMs' susceptibility to hallucinations makes them unreliable for evaluating factual correctness through direct prompting, and their dependence on a ground truth story poses a significant scalability challenge. Moreover, the informativeness evaluated in these works does not consider factual accuracy.

In contrast, our work addresses these limitations by offering a unified, user-centered benchmark that spans the full BI query spectrum—descriptive, diagnostic, predictive, and prescriptive—and enables both dataset generation and unsupervised evaluation without dependence on strict ground truth.

3 BI Benchmark Dataset Construction

A crucial component of BI-Bench, is a carefully curated and dynamically extensible dataset, designed to facilitate comprehensive and realistic evaluations of BI systems. It captures essential attributes (Table 2) and facilitates detailed analysis of system performance across diverse query types and domains. To complement our dynamic benchmark generation pipeline, BI-Bench includes a ready-to-use benchmark dataset, curated through a multi-stage, semi-automated process that combines the power of large language models (LLMs) with human validation. This ensures that the dataset remains grounded in real-world business scenarios while allowing domain-specific adaptability.

Table 1 presents the structure of our benchmark dataset, capturing essential metadata, such as analytical category, query complexity, and associated tables. This structure allows for a comprehensive evaluation by categorizing queries based on their intent, complexity, and the required data sources.

Our benchmark data builds on 29 diverse datasets from Spider 2.0, spanning domains such as Transportation, Finance, Healthcare, etc. Each database serves as the basis for BI queries, generated through the following five-step construction pipeline.

3.1 NL Question Generation

Natural language queries were generated using Llama-3.3-70B-Instruct model, with tailored prompts for each of the four analytical categories — descriptive, diagnostic, predictive, and prescriptive — across basic, intermediate, and advanced complexity levels. This method ensures generation of diverse and realistic BI queries. A total of 336 NL queries were generated, spanning four analytical categories across 29 datasets. Additionally, 60 descriptive queries from Spider 2.0 were integrated to enhance coverage.

[1] https://github.com/ankush31089/BI-Benchmark

Field	Description
Query ID	Unique identifier for each query.
Natural Language Query	The actual user query expressed in natural language.
Category	Type of analytical query: Descriptive, Diagnostic, Predictive, Prescriptive.
Table(s) Used	The specific tables from the schema that are required to answer the query.
Domain	The business or application domain (e.g., Sales, HR, Healthcare).
Complexity	Difficulty level categorized as Basic, Intermediate, or Advanced, based on the required reasoning and join operations.

Table 1: Structure of the BI Benchmark Dataset

Metric	Value
Total Number of Queries	273
Distinct Domains Covered	10
Queries per Domain	Transportation:20, Healthcare:34, Sports & Entertainment:67, Marketing:19, Finance:15, Software & IT:46, E-commerce:20, Logistics & Retail:31, Legal and Technology:17, Databases:4
Distinct Datasets Used	29
Category Distribution	Descriptive:115, Diagnostic:60, Predictive:59, Prescriptive:39
Complexity Distribution	Basic:93, Intermediate:94, Advanced:86

Table 2: Dataset Statistics

3.2 Answerability Check

To verify the validity of the generated questions, we implemented an automated answerability check using GPT-4o. This intentional shift to a distinct LLM was crucial for mitigating potential biases from relying solely on Llama-3.3-70B-Instruct through the process. GPT-4o assessed each query by providing detailed justifications on whether it could be answered based on the available data schema. Its strong reasoning capabilities enabled an objective feasibility evaluation. As a result, out of 336 queries generated in Step 1, 103 were deemed unanswerable and discarded.

3.3 Table(s) Identification

For each query, we used Llama-3.3-70B-Instruct to identify the relevant tables needed to generate a complete and accurate response. The model analyzed the query's intent and mapped it to the appropriate database tables. For descriptive queries sourced from Spider 2.0, table information was directly extracted from the provided SQL queries.

3.4 Table Completeness Verification

After table identification in Step 3, we used GPT-4o to verify the completeness and accuracy of the identified tables for each query. This step ensured that all necessary tables were included and no relevant data sources were overlooked. GPT-4o provided explanations justifying the inclusion of specific tables, enabling a transparent and auditable verification process. During validation, 79 table assignments were corrected to address incomplete or incorrect selections.

3.5 Human Validation

As a final quality assurance step, a human expert meticulously reviewed each query, its associated tables, and the explanations from Steps 2 and 4. During this process, 20 queries were discarded due to inaccuracies or ambiguities, and 15 table assignments were corrected for accuracy and consistency. This manual validation ensured the reliability of both answerability assessments and table identifications, addressing subtle errors that automated processes might have missed. By combining LLM-based validation with human oversight, our multi-layered approach guaranteed the high quality and robustness of our BI benchmark dataset.

Following this process, BI-Bench delivers a benchmark dataset that comprehensively spans all BI query types, multiple domains, and varying complexity levels. The dataset supports schema linking through a Table(s) Used field, maintains high quality through both automated and expert validation, and enables reproducibility via a publicly available benchmark and methodology.

To illustrate the BI-Bench dataset's structure and diversity, Table 3 presents a selection of sample queries. The complete benchmark, including its schema, prompts used for NL question generation, table identification, answerability check, and table completeness verification and detailed documentation, is publicly available at https://github.com/ankush31089/BI-Benchmark.

We now describe the BI-Bench evaluation pipeline, designed to assess BI system responses in a fully automated, unsupervised manner.

Query ID	Natural Language Query	Category	Tables Used	Domain	Complexity
001	What is the total number of advisories with a CVSS3 score greater than 7?	Descriptive	DEPS_DEV_V1_ADVISORIES	Software & IT	Basic
002	How do different traffic sources impact the conversion rate of users from various age groups?	Diagnostic	THELOOK_ECOMMERCE_EVENTS, THELOOK_ECOMMERCE_USERS, THELOOK_ECOMMERCE_ORDERS	E-commerce	Intermediate
003	What is the relationship between a team\textquotesingle s defensive aggression and their likelihood of conceding goals, and how does this vary across different leagues?	Predictive	EU_SOCCER_MATCH, EU_SOCCER_TEAM_ATTRIBUTES, EU_SOCCER_LEAGUE	Sports & Entertainment	Advanced
004	Which fare conditions should be prioritized to increase revenue on specific routes?	Prescriptive	AIRLINES_TICKET_FLIGHTS, AIRLINES_FLIGHTS, AIRLINES_SEATS	Transportation	Intermediate

Table 3: Sample queries from our BI benchmark dataset.

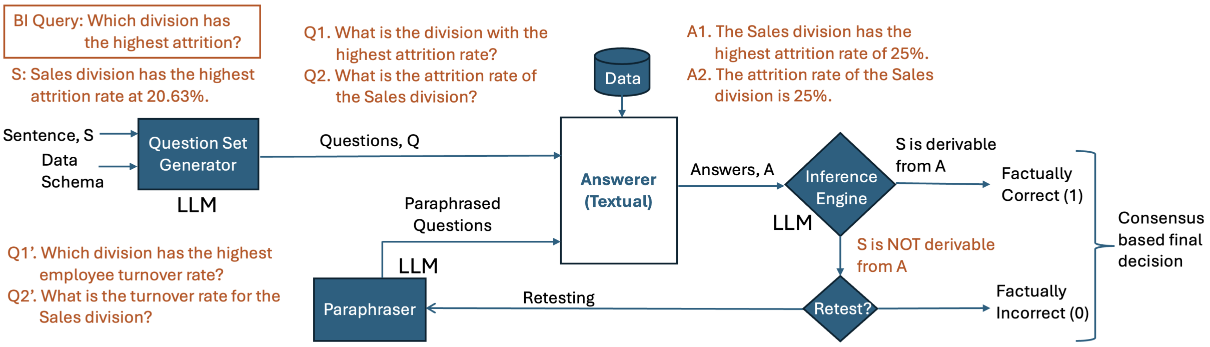
BI Query: Which division has the highest attrition?
S: Sales division has the highest attrition rate at 20.63%.
Q1. What is the division with the highest attrition rate?
Q2. What is the attrition rate of the Sales division?
Data
A1. The Sales division has the highest attrition rate of 25%.
A2. The attrition rate of the Sales division is 25%.
Sentence, S
Data Schema
Question Set Generator
LLM
Questions, Q
Answerer (Textual)
Answers, A
Inference Engine
LLM
S is derivable from A
Factually Correct (1)
Consensus based final decision
Paraphrased Questions
S is NOT derivable from A
Q1'. Which division has the highest employee turnover rate?
Q2'. What is the turnover rate for the Sales division?
LLM
Paraphraser
Retesting
Retest?
Factually Incorrect (0)

Figure 1: Factual Correctness Assessment for a sentence (S) from BI Response on *Employee Attrition* data

4 Automated Evaluation Pipeline

To assess the quality of BI system responses, we introduce an automated, domain-agnostic evaluation pipeline—a core component of BI-Bench. This pipeline provides a detailed and multidimensional evaluation across four key dimensions: Factual Correctness, Answerability, Relevance, and Presentation. The pipeline operates in a zero-shot, unsupervised manner using open-source LLMs and prompt engineering, without requiring any training data or human intervention. For reproducibility and transparency, all prompt templates used in the pipeline are included in the Appendix A.

4.1 Factual Correctness

The evaluation process begins with verifying the factual accuracy of the BI response—a crucial step in ensuring reliable decision-making. We employ a *student-teacher framework*, where the teacher evaluates the student's (BI system's) output. Given the complexity of BI responses, the first step is to break them down into simpler components, allowing a weaker expert to assess each step individually. This structured approach enables accurate evaluation of complex BI queries without requiring highly skilled experts.

To implement this, first, each BI response (Student's output) is decomposed into sentence-level units. Then, an LLM-powered *Question Generator* converts each sentence S into a tailored set of questions, Q grounded in the dataset schema. Each question is answered by a weak expert (*Answerer*), generating an answer set A. We use Text-to-SQL-to-NLG as an answerer for descriptive questions.

Next, an *Inference Engine* determines whether sentence S can be fully or partially inferred from A. If a mismatch is found, S is marked factually incorrect (0), else factually correct (1). To enhance robustness in the cases where a mismatch is found, *paraphrasing and retesting* are employed—using a *Paraphraser* to reframe questions, followed by re-answering and re-inference for two more iterations. The final correctness score is based on consensus across these runs, improving reliability and minimizing LLM-induced variability. The final score is computed as the ratio of number of factually correct sentences to the total number of sentences. This step gives us two sets, factually correct set (FC) and factually incorrect set (FIC).

4.2 Answerability

In BI scenarios where *factual accuracy* is paramount, factually incorrect statements (FIC) not only degrade the utility of a response but may also conflict with factually correct (FC) content, creating confusion. To address this, we penalize such conflicts to ensure that the reliability of the answers in FC content is preserved in the overall *answerability* score. We define *answerability* as the degree to which a user query is *answered correctly*: fully (1), partially (0.5), or not at all (0). An illustrative example of such a conflict is following:

User Query*: Which division has the highest attrition?*

BI Response*: The Sales division has the highest attrition rate at 25% (FC). The top three divisions with high attrition rates are Marketing, Research, and HR, in that order (FIC).*

Here, both FC and FIC parts individually score 1 on answerability, but their conflict reduces trust in the answer. To mitigate this, we introduce a $penalty$ factor (0.1 (low), 0.5 (partial), or 0.9 (high)) to account for the level of the conflict, and apply this to the ans_{FC} score.

$$ans = ans_{FC} \times (1 - penalty \times ans_{FIC})$$

For instance, in this case, a penalty of 0.9 is applied because of total conflict. The final score is rounded to 0, 0.5, or 1. We use LLM-as-a-judge strategy to infer $penalty$, ans_{FC} and ans_{FIC} based on the sets FC and FIC identified from the factual correctness evaluation in previous Section 4.1.

4.3 Relevance

Relevance measures how well a BI response *aligns with the user query, while ensuring factual accuracy*. Similar to Answerability, we apply a penalty-based approach to account for the impact of factually incorrect (FIC) content on the relevance of factually correct (FC) information. We define *relevance* as: Highly relevant (1.0), partially relevant (0.5) and irrelevant i.e. missing key aspects of the query (0). Note that relevance differs from answerability–a response may not directly answer a query but can still be considered relevant and may provide other important insights. The overall relevance score is computed as:

$$rel = rel_{FC} \times (1 - penalty \times rel_{FIC})$$

The score is then rounded to 0, 0.5, or 1. Similar to answerability, we use zero-shot LLM-as-a-judge setup to compute Relevance of FC content (rel_{FC}), Relevance of FIC content (rel_{FIC}), and Degree of conflict between them ($penalty$).

4.4 Presentation Aspects

Presentation is crucial for BI response usability. Following prior work, we evaluate three aspects of presentation in a BI response: Clarity, Coherence, and Narrative Quality.

Clarity measures how easily the response can be understood: (0: difficult to understand, ambiguous, or unclear; 0.5: somewhat clear but could be improved for better comprehension; 1: very clear, unambiguous, and easy to understand).

Coherence evaluates the logical flow and structural organization: (0: disjointed; 0.5: moderately connected; 1: well-structured and logically connected).

Narrative Quality captures the engagement, depth, and insight of the response: (0: flat, lacking depth or engagement; 0.5: somewhat insightful; 1: highly engaging and thought-provoking).

A zero-shot LLM-as-a-judge is tasked with assigning individual scores of 0, 0.5, or 1 for each of these aspects, using the BI response and user query as input. The final presentation score ($pres$) is computed as a weighted average of these three scores and then rounded to the nearest valid score: 0, 0.5, or 1.

$$\begin{aligned} pres = \; & 0.4 \times clarity + 0.3 \times coherence \\ & + 0.3 \times narration \end{aligned}$$

5 Experiments and Results

This section presents the experimental setup and evaluation results for our proposed unsupervised BI evaluation framework. We implemented a baseline BI system to generate responses for a representative subset of our BI-Bench dataset and assessed the quality of these responses using our automated evaluation pipeline. Additionally, we conducted human evaluations to validate the reliability of our automated scores.

5.1 Baseline Implementation

We constructed a baseline BI system that generates natural language responses using a two-stage pipeline: **Text-to-SQL**: translates user queries into executable SQL statements. **SQL-to-NLG**: Transforms SQL query results into natural language explanations. To simulate more context-rich BI answers, we also generated two supplementary questions for each user query. We processed these additional questions through the same pipeline, and all resulting NLG outputs were synthesized into a unified final response.

5.2 Experiment Setup

We conducted experiments on a randomly selected subset of 22 descriptive queries from our BI benchmark dataset. Each query was processed through

Metric	Factual Correctness	Answerability	Relevance	Presentation
Inter-Annotator Agreement (%)	90.9	86.4	86.4	86.4
System Accuracy on Agreed Subset (%)	80.0	89.5	73.7	100.0
Pearson Correlation (r, p)	0.73, $p = 0.0003$	0.94, $p = 0.0000$	0.84, $p = 0.0000$	1.0, $p = 0.0000$

Table 4: Evaluation metrics across four dimensions.

the baseline system to generate a composite response. These responses were then assessed using our automated evaluation pipeline (using Llama-3.1-70B-Instruct model), which scores across four key dimensions: Factual Correctness, Answerability, Relevance, and Presentation.

Each response received dimension-wise scores from the automated system. To validate these automated assessments, we conducted a human evaluation study involving two expert annotators with BI and data analysis backgrounds. These annotators were presented with the selected descriptive queries along with their baseline responses and related data information (target table names and schema). To streamline human annotation and ensure thorough evaluation, we provided annotators with: (1) access to Snowflake UI for direct SQL querying and factual verification; and (2) access to evaluation pipeline internals, such as question set with their corresponding SQLs, SQL execution results and NL-based answers generated during factual correctness checking, for transparency and validation support. With these resources, the annotators were then asked to independently assign human scores for each of the four evaluation dimensions, reflecting their subjective assessments based on their direct interaction with the data and the intermediate outputs of our evaluation pipeline. To minimize subjectivity, annotators were given clear instructions defining each evaluation dimension and were instructed to follow the same scoring criteria and circumstances as those used in our automated pipeline.

5.3 Evaluation Metrics and Analysis

We conducted three analyses to assess the robustness of our automated evaluation framework and its alignment with human judgment:

Inter-Annotator Agreement: To measure consistency between human annotators, we calculate the number of instances where their scores match exactly for each dimension. High agreement rates, as shown in Table 4, indicate strong alignment in the annotators' assessment criteria, thereby reinforcing the validity of the reference scores.

System Accuracy on Agreed Subset: For queries where both annotators provided identical scores (i.e., perfect agreement), we calculated the system's accuracy — defined as the percentage of cases where the system's score matched the average human score. This metric indicates the system's performance relative to high-confidence ground truth data obtained through human annotations. Our system achieves an average accuracy of $\approx 86\%$ across all dimensions, as shown in Table 4.

Correlation with Human Scores: We computed Pearson correlation between system-generated and average human scores (on the agreed subset) for each dimension. Strong, statistically significant correlations (Table 4) indicate high alignment between the automated and human evaluations.

6 Key Lessons Learned and Challenges

Developing this benchmark framework faced several challenges, some of which remain unresolved. A key challenge in constructing the BI benchmark dataset was developing an automated, domain-agnostic pipeline that minimizes human validation while ensuring high-quality question generation and table verification. The goal was to enable scalable, customizable benchmark creation across diverse datasets with minimal manual intervention, balancing automation with generalizability. While BI-Bench enables automated evaluation across different categories, assessing the factual correctness of diagnostic, predictive, and prescriptive questions is still in progress. The main challenge lies in ensuring reliable evaluation, particularly for prescriptive questions, due to dependence on weak expertise and the difficulty of identifying qualified experts. Additionally, our approach, which heavily relies on LLMs, must comply with the strict token limits imposed by the models. As a result, datasets with large schemas or longer BI responses may face limitations or performance issues.

7 Conclusion and Future Work

We presented BI-Bench, a comprehensive, domain-agnostic framework for benchmark creation and unsupervised evaluation of BI systems, eliminating the need for ground-truth annotations. We release a benchmark dataset spanning diverse analytical categories, enriched with metadata for deeper evaluation. Designed for industrial deployment, BI-Bench offers a scalable, domain-agnostic pipeline that automates query generation, verification, and evaluation—minimizing human effort and enabling enterprises to benchmark and improve BI capabilities with minimal overhead. Its modular design supports adaptation to domain-specific datasets, facilitating broader adoption across enterprise use cases. As future work, we aim to refine evaluation for non-descriptive queries—particularly prescriptive analytics—address LLM context limitations, and extend the benchmark to include multi-turn BI dialogues and multimodal insights.

References

Shir Ashury-Tahan, Yifan Mai, Rajmohan C, Ariel Gera, Yotam Perlitz, Asaf Yehudai, Elron Bandel, Leshem Choshen, Eyal Shnarch, Percy Liang, and Michal Shmueli-Scheuer. 2025. The mighty torr: A benchmark for table reasoning and robustness. *Preprint*, arXiv:2502.19412.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. Tabfact: A large-scale dataset for table-based fact verification. *Preprint*, arXiv:1909.02164.

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. 2023. Text2analysis: A benchmark of table question answering with advanced data analysis and unclear queries. *Preprint*, arXiv:2312.13671.

Xueyu Hu, Ziyu Zhao, Shuang Wei, Ziwei Chai, Qianli Ma, Guoyin Wang, Xuwu Wang, Jing Su, Jingjing Xu, Ming Zhu, Yao Cheng, Jianbo Yuan, Jiwei Li, Kun Kuang, Yang Yang, Hongxia Yang, and Fei Wu. 2024. Infiagent-dabench: Evaluating agents on data analysis tasks. *Preprint*, arXiv:2401.05507.

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.

Fangyu Lei, Jixuan Chen, Yuxiao Ye, Ruisheng Cao, Dongchan Shin, Hongjin Su, Zhaoqing Suo, Hongcheng Gao, Wenjing Hu, Pengcheng Yin, Victor Zhong, Caiming Xiong, Ruoxi Sun, Qian Liu, Sida Wang, and Tao Yu. 2025. Spider 2.0: Evaluating language models on real-world enterprise text-to-sql workflows. *Preprint*, arXiv:2411.07763.

Joy Mahapatra, Sudip Kumar Naskar, and Sivaji Bandyopadhyay. 2016. Statistical natural language generation from tabular non-textual data. In *Proceedings of the 9th International Natural Language Generation conference*, pages 143–152, Edinburgh, UK. Association for Computational Linguistics.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. *Preprint*, arXiv:1508.00305.

Yajing Yang, Qian Liu, and Min-Yen Kan. 2024. DataTales: A benchmark for real-world intelligent data narration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10764–10788, Miami, Florida, USA. Association for Computational Linguistics.

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.

Dan Zhang, Sining Zhoubian, Min Cai, Fengzu Li, Lekang Yang, Wei Wang, Tianjiao Dong, Ziniu Hu, Jie Tang, and Yisong Yue. 2025. Datascibench: An llm agent benchmark for data science. *Preprint*, arXiv:2502.13897.

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

A Appendix

Descriptive Question Generation Prompt

Task:
You are an expert in Business Intelligence (BI) and Data Analytics. Your objective is to generate **descriptive analytical questions** that help uncover **patterns, summaries, trends, and key performance indicators** from a business dataset.

What Are Descriptive Analytical Questions?

Descriptive questions focus on understanding **what happened**, providing summaries of historical data such as totals, averages, comparisons, trends over time, and breakdowns by different business dimensions.

Descriptive questions **do NOT explore reasons**, drivers, impacts, correlations, or causality. Avoid "what influences", "why", "what causes", "impact of", etc.

How Should These Questions Be Framed?

1. **Use business-friendly language** — business users do not refer to table or column names in queries.
2. **Avoid explicit mentions of table/column names** in the question text.
3. **Ensure each question reflects natural business thinking**.
4. **Questions should be self-contained and clear**, so they can be answered directly using the dataset.

Complexity Levels

1. **Basic** – Simple aggregations, trends, and summaries *Example: “What was the total revenue last year?”*
2. **Intermediate** – Multi-dimensional summaries or comparisons *Example: “How did sales vary across different regions and product categories?”*

Descriptive Question Generation Prompt (contd.)

3. **Advanced** – More granular breakdowns, trend analysis over time, or customer segmentation *Example: “What are the top customer segments contributing to quarterly growth in product sales?”*

Instructions:

1. Generate exactly **9 descriptive questions** in total : **3 Basic**, **3 Intermediate**, and **3 Advanced**.
2. Each question should be **fully answerable using the given dataset schema**.
3. Avoid questions that require external data or are ambiguous.
4. Use specific **business contexts** like customer behavior, product sales, regional trends, time-based comparisons, etc.
5. **Avoid mentioning actual table or column names in the question text.**
6. **Enclose each question within ‘<question>...</question>‘ tags.**
7. Return a clean, structured output in JSON format with each question and its complexity level.
8. **Again, do not return anything except the raw JSON array. Avoid any headings, notes, or boxed formats.**
9. All 9 questions must be returned in a **single flat JSON array**.
10. Do **not create multiple arrays or group questions by complexity** — just one array with 9 JSON objects.

Ensure:

1. Each question must be fully answerable using only the columns and data types explicitly provided in the schema.
2. Do not invent additional columns or assume missing information.
3. Only use the column names and sample values shown in the schema.
4. If a question depends on unavailable data, skip it.
5. Do not make assumptions about data availability or granularity (e.g., specific time periods, locations, customer types,

Descriptive Question Generation Prompt (contd.)

etc.) unless clearly stated in the schema. - Prefer: “...across time”, “by region”, “per category” - Avoid: “last year”, “premium customers”, “city-wise” if such details are not explicitly part of the dataset.

Dataset Schema:

<schema>

Output Format:

~~~
```json
[
  {
    "question": "<question_text>",
    "complexity": "Basic |
    Intermediate | Advanced"
  }
]
~~~

Return a single JSON array named ‘questions‘, not multiple arrays.

Table Identification (Descriptive) Prompt

Task:

You are given a dataset schema and a natural language question.

Your task is to identify the **minimum set of tables required to answer the question**, based strictly on the columns available in each table.

Instructions:
1. Carefully analyze the question and identify what data elements are required to answer it.
2. Refer to the dataset schema and determine which tables contain those elements.
3. Select **only the relevant tables** — avoid including unnecessary ones.
4. **Do not guess or assume columns/tables that are not explicitly in the schema.**
5. Return the output as a clean **flat JSON array of table names**, without any extra text, formatting, or explanations.
6. Do **not return any explanation or additional formatting**.
7. **Do not print multiple separate JSON arrays — return just one complete JSON array.**

Input:
Schema:
<schema>
Question:
<question>
Output Format:

~~~
```json
["TABLE_NAME_1", "TABLE_NAME_2"]
~~~

Answerability Check Prompt

Task:
You are a **data analyst** responsible for validating whether a given **Business Intelligence (BI) question** can be answered using the provided **dataset schema**. Your objective is to determine:

Answerability Assessment:
Assume the availability of a LLM based agent which can answer **Business Intelligence (BI) question**.
1. Does the dataset contain all necessary tables and columns to answer the question?
2. If any required data is missing, what specifically is absent?

Reasoning & Justification:
1. Provide a clear explanation supporting your assessment.
2. If the dataset is sufficient, justify why.
3. If the dataset is insufficient, identify the missing components.

Missing Data Identification:
1. List missing tables (if any).
2. List missing columns (if any) within existing tables.

Instructions:

Analyze the Schema:
1. Identify the tables and columns that are directly relevant to answering the question.

Determine Answerability:
1. If all necessary data exists, classify as "Answerable".
2. If any critical data is missing, classify as "Not Answerable".

Explain Clearly:
1. Justify why the dataset is sufficient or insufficient.
2. If insufficient, specify what is missing.

Input:

Dataset Schema (TableName_Columns, data types, unique column values):
<schema>

Answerability Check Prompt (contd.)

BI Question:
<query>

Output Format (JSON):

"answerability": "<Answerable / Not Answerable>",
"reasoning": "<Detailed explanation of the assessment>",
"missing_data":
"status": "<Complete / Incomplete>",
"missing_tables": ["<List of missing tables>"],
"missing_columns": "<table_name>": ["<List of missing columns>"]

Result:

Factual Correctness Paraphraser Prompt

You are given:
1. Fact: A statement containing the answer to the questions you need to paraphrase.
2. Set of Questions: A list of questions, each possibly containing comma-separated paraphrases and enclosed within <question></question> tags.

Your Task:
For each question in the input, generate only one new paraphrase that:
1. Retains the same meaning as the original question and its available paraphrases, extracting the same information from the Fact.
2. Ensures that the answers to the new paraphrase, along with the original and existing paraphrases, fully cover the information provided in the Fact.
3. Does not repeat the original question or any of its paraphrases.
4. Uses different wording or sentence structure to create a distinct paraphrase.

Output Format:
Each generated paraphrased question should be placed within <question></question> tags in the output. Ensure that the generated paraphrase is unique and different from the original question and its available paraphrases. Do not explain the output. Do not generate any extra information.

Fact: $fact$
Set of Questions: $questions$
Output:

Factual Correctness Question Gen Prompt

Given a database schema and a specific fact as inputs, your task is to generate a set of questions that can be answered using a text-to-SQL pipeline. These questions should be designed to extract all the information provided in the given fact, with their answers combined to completely overlap with the given input fact. Each question should be carefully crafted based on the attributes, relationships, and constraints defined in the given schema. Ensure the questions are aligned with the database schema, utilizing the correct tables, columns, and relationships. The questions should focus on extracting all the key elements of the given fact, ensuring that the answers to these questions provide a full picture when combined. Each generated question should be placed within <question></question> tags in the output.
Schema: $schema$
Fact: $fact$
Questions:

Factual Correctness Inference Prompt

You are an impartial judge tasked with comparing Fact1 and Fact2. Your goal is to see if all details of Fact1 can be found in Fact2.
Instructions:
1. Check if all details from Fact1, like numbers, names, dates, etc., can be fully inferred from Fact2.
2. Respond with:
"True" if Fact2 fully matches Fact1.
"Partially True" if some details from Fact1 are missing in Fact2.
"False" if Fact2 misses most details or changes any information from Fact1.
3. Do not explain, just state the answer.
4. Always place the response within <result> </result> tags in the output.

Fact1: $fact1$
Fact2: $fact2$
Comparison Response:

Presentation Evaluation Prompt

Task: Evaluate the given Set of Insights related to the given User Query based on the three presentation aspects: Clarity, Coherence, and Narrative Quality. For each aspect, rate the insights on a scale of 1 to 3:

Clarity: How easy is it to understand the insights provided?
1: The insights are difficult to understand, ambiguous and unclear, making the information indigestible for the user.
2: The insights are somewhat clear but could be improved for better understanding.
3: The insights are very clear, unambiguous and easy to understand.
Coherence: How logically organized and connected are the insights? Do the insights flow logically, with each point building on the previous one, making it easy to follow the key trends?
1: The insights are disjointed and lack logical flow.
2: The insights are somewhat connected but could have better transitions and organization.
3: The insights are well-organized with clear connections between them.
Narrative Quality: How engaging, meaningful, and insightful is the narrative? Does it provide deep and thought-provoking insights? Does it add some level of analysis or explanation for the given insights?
1: The insights are dry, with little to no engagement or depth.
2: The insights are somewhat engaging, but could provide more depth or emotional appeal.
3: The insights are highly engaging and provide meaningful and deep analysis.

Provide the rating for each criterion in the following format:
<clarity> 1/2/3 </clarity>
<coherence> 1/2/3 </coherence>
<narrative> 1/2/3 </narrative>

User Query: $query$
Set of Insights: $BI\_response$

Answerability Evaluation Prompt

You are given the following:
1. User Query: A question asked by the BI user in natural language.
2. Factually Correct Insights: A set of accurate facts or statements that may answer the user query.
3. Factually Incorrect Insights: A set of incorrect facts or statements that may also answer the user query.
4. Data Schema: Set of column names with their type and possible values present in the target data.
Your task is to determine whether the User Query is fully, partially, or not answered at all based on both the Factually Correct and Incorrect Insights, following the instructions below.

Scoring Criteria:
"1": The User Query is directly addressed by the insights and provides answers to all the aspects posed in the user query.
"0.5": The User Query is only partially addressed and only some aspects of the query are answered.
"0": The User Query is not addressed at all (i.e., the insights do not provide relevant or sufficient information).

Instructions:
1. Evaluate the User Query based on the Factually Correct Insights first, to determine if it provides a complete answer according to the given schema. If insights are not available or an empty string, assign a score of 0. Record the identified score within $< ans\_fc >$ tags in the output.
2. Check the extent to which Factually Incorrect Insights also answer the user query. If insights are not available or an empty string, assign a score of 0. Record the score within $< ans\_fic >$ tags in the output.
3. If values within $< ans\_fic >$ is more than 0 in step 2, consider whether the answer in Factually Incorrect Insights negatively impact the answerability from Factually Correct Insights by offering incorrect or conflicting information. Assign a penalty score according to the below criteria and record within $< penalty >< /penalty >$ tags in the output.

Answerability Evaluation Prompt (contd.)

Penalty is 1 if Factually Incorrect Insights completely contradicts the correct answer in Factually Correct Insights.
Penalty is 0.5 if Factually Incorrect Insights partially contradicts the correct answer in Factually Correct Insights.
Penalty is 0.1 if Factually Incorrect Insights doesn't contradict the correct answer in Factually Correct Insights at all.
Strictly adhere to the information provided in this request. Always enclose the required scores within the specified tags in the generated output. Do not explain your output.
Schema: *schema*
User Query: *query*
Factually Correct Insights: correct_insights
Factually Incorrect Insights: incorrect_insights
Output:

Relevance Evaluation Prompt (contd.)

Penalty is 1 if majority of the information in Factually Incorrect Insights contradicts the Factually Correct Insights.
Penalty is 0.5 if some information in Factually Incorrect Insights contradicts the Factually Correct Insights.
Penalty is 0.1 if Factually Incorrect Insights don't contradict the Factually Correct Insights at all.
5. Strictly adhere to the information provided in this request. Always enclose the required scores within the specified tags in the generated output. Explain your output briefly.

User Query: *query*
Factually Correct Insights: correct_insights
Factually Incorrect Insights: incorrect_insights
Output:

Relevance Evaluation Prompt

You are given the following:
1. User Query: A question asked by the BI user in natural language.
2. Factually Correct Insights: A set of accurate facts or statements that may answer the user query.
3. Factually Incorrect Insights: A set of incorrect facts or statements that may also answer the user query.
Your task is to determine the Relevancy of the provided set of insights in relation to the given User Query, following the instructions below.

Scoring Criteria:
"0": The insights are not relevant or related with respect to the user query and don't address the user's needs.
"0.5": The insights are somewhat relevant but miss key aspects of the query.
"1": The insights are highly relevant and directly answer the user's query.

Instructions:
1. Relevancy of a set of insights with respect to a user query refers to how closely the insights address the specific information or context requested in the User Query. A set of insights is considered relevant if it directly contributes to answering the user query, aligns with the key aspects of the question, and provides useful, actionable information based on the user's needs.
2. Evaluate the relevancy of Factually Correct Insights with respect to the given User Query, as defined in Step 1. If insights are not available or an empty string, assign a score of 0. Record the identified score within <rel_fc> tags in the output.
3. Evaluate the relevancy of Factually Incorrect Insights with respect to the given User Query, as defined in Step 1. If insights are not available or an empty string, assign a score of 0. Record the score within <rel_fic> tags in the output.
4. If values within <rel_fic> is more than 0 in step 2, consider whether the insights available in Factually Incorrect Insights provide conflicting information as given in Factually Correct Insights. Assign a penalty score according to the below criteria and record within <penalty></penalty> tags in the output.