

2024년 08월 29일

AUTHOR

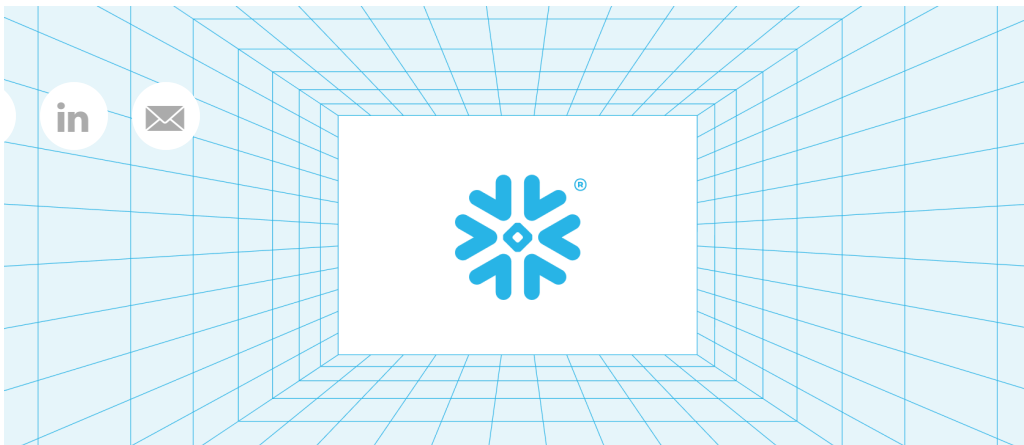


Renee Huang

Snowflake Cortex Analyst: Evaluating Text-to-SQL Accuracy for Real-World Business Intelligence Scenarios

Gen AI

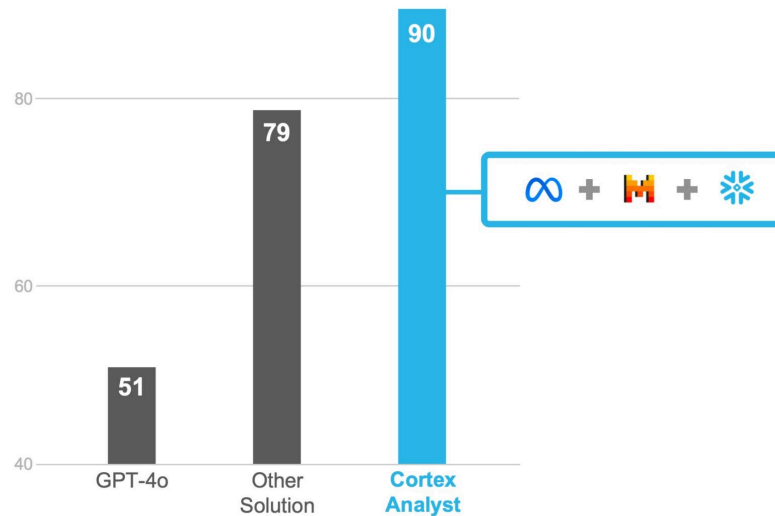
SHARE



Being able to build a system that allows business users to ask intricate, complex and nuanced questions on structured data using natural language has long been a goal for many researchers and data specialists. This task, often referred to as “text-to-SQL,” was historically introduced as a coding challenge. However, it has proven to be incredibly difficult, as it involves complexities not only at the code layer but also on the level of business context. [Snowflake Cortex Analyst](#) addresses this challenge by introducing an agentic AI system, coupled with a semantic model to achieve an impressive 90%+ SQL accuracy on real-world use cases, ensuring business users get accurate and precise answers.

Based on an internal evaluation mirroring real-world use cases, Cortex Analyst outperforms alternatives — nearly 2x as accurate as single-prompt SQL generation from GPT-4o and about 14% more accurate than other solutions on the market. For more details on our system, refer to our [Cortex Analyst: Behind the Scenes](#) blog post.

Text-to-SQL Accuracy (%)



AUTHOR



Renee Huang

Figure 1. SQL generation accuracy for Snowflake Cortex Analyst vs. alternatives

SHARE

Facebook LinkedIn Email In this blog, we delve into the benchmarking and evaluation of Cortex Analyst, highlighting how it outperforms existing solutions in delivering accurate SQL generation for real-world business intelligence (BI) tasks.

Challenges of traditional text-to-SQL benchmarks


What makes a system “accurate,” and how can we build confidence in its accuracy from the perspective of the end business user? Accuracy of text-to-SQL systems has traditionally been evaluated using popular SQL benchmarks like Spider and BIRD. While these benchmarks have offered researchers effective tools for experimentation, the gap between these benchmarks and real-world business use cases is significant.

AUTHOR



Renee Huang

SHARE

   **Figure 2.** Considerations for real-world conversational self-serve analytics evaluation

Looking at Spider [leaderboards](#), where top models achieve 90%+ accuracy and even solutions from a year ago surpass 80%, one might assume that text-to-SQL is a solved problem. So, if that's the case, why isn't everyone already using LLM solutions to interact with data?

The reality is that when applied to real-world business use cases, these solutions often fall short. For instance, when we tested a state-of-the-art language model, GPT-4o, using our internal evaluation set, its accuracy plummeted to 51%. This significant gap between benchmark performance and real-world application reveals several limitations of traditional benchmarks. There are four main pillars that define the gap between nice 90% benchmark numbers vs. real-world BI use cases:

Question complexity: The natural language questions used in existing benchmarks are often not very complex, industry-specific or realistic. They also fail to capture the types of questions frequently asked in BI contexts.

Schema complexity: The databases in these benchmarks don't always represent the complex, messy and often unclear schemas that are common in BI tasks. In addition, time-series tables — which are very common in transactional data, like sales or user web activity — are also significantly underrepresented in existing benchmarks.

SQL complexity: These benchmarks often lack complex queries, such as those containing window functions, CTEs and complex aggregations, which are crucial for real-world BI tasks.

Measure SQL in the context of semantics: Traditional benchmarks fail to account for the importance of aligning SQL queries with specific business definitions and metrics. For instance, how different organizations define key metrics like “Daily Active Users” can vary significantly, impacting the accuracy of SQL queries in real-world applications.

These limitations underscore the need for an evaluation framework that better reflects the complexities of real-world BI tasks.

A more realistic approach to evaluation

To accurately measure the performance of Cortex Analyst, we developed a benchmark suite of 150 questions that mirror the real-world tasks business users encounter, covering topics such as sales, marketing, finance, etc. This approach includes:

Typical BI questions

We categorized questions into three levels — filtering, aggregation and trends — covering a broad spectrum of BI queries. For example:

Level 1 questions (filtering): “Give me the list of the top 10 clients who made the most purchases in the last month.”

Level 2 questions (aggregation): “How much revenue came from book sales in the last two weeks of April 2023?”

Level 3 Questions (trends): “Did any product lines show consecutive revenue increases for five or more days in 2023?”

Schema shapes

Dimension vs. measures column setup

Our initial focus was on ensuring high accuracy across a wide variety of customer use cases on a single view with pre-joined data. For this scope, curated BI tables are typically structured with clear distinctions between dimensions, time dimensions and measures. This often results in a flattened, wide table, which is more suitable for data analytics than a long, narrow table.

AUTHOR



Renee Huang

SHARE

For example, while the following schema shapes are often seen in raw data feeds, they are not ideal for data analytics:

Date	Metric Name	Metric Value
2024/01/01	Sales	1,000,000
2024/01/01	Cost	500,000
2024/01/01	Revenue	500,000

Table 1. An example of table shape not frequently used in BI.

As we expand to support joins and more complex schemas, we are focusing on the most popular data modeling approach: dimensional data modeling. This approach organizes data into facts and dimensions, typically using a star schema or snowflake schema setup, which aligns more closely with best practices in business intelligence.

f in je. data considerations

As previously mentioned, time-series data, like sales or user activity, is often underrepresented in traditional benchmarks. In real-world scenarios, we frequently encounter cases where the time axis is nonconsecutive, which requires a different approach in SQL to handle calculations correctly.

One of the challenges with a nonconsecutive time axis is that it can lead to errors in analysis, particularly in “month-over-month” or “week-over-week” comparisons. Two common SQL approaches to address this — using self-joins or window functions like LEAD and LAG — can produce seemingly correct results, but they might be misleading if the underlying time data isn’t consecutive.

For example:

A self-join explicitly matches the current period with the prior period. This method is generally more robust, especially when handling nonconsecutive time data.

LEAD/LAG functions rely on the ordering of the time axis and may fail to compare the correct periods if the data isn’t consecutive.

Without careful validation and intermediate calculations (like showing both current and previous periods), there’s a risk that end users could make decisions based on inaccurate comparisons.

AUTHOR



Renee Huang

SHARE

In our evaluation data set and within Cortex Analyst, we've accounted for these nuances. We ensure the system favors more reliable methods and provides sufficient intermediate steps for users to validate the results themselves.

See details in the later section, "Evaluation result examples: Business Question 2," on how we handle "day-over-day" type questions differently, considering possible nonconsecutive time axis.

SQL distributions

To flexibly answer a variety of business questions across different schema shapes, it naturally requires our system to generate SQL beyond simple "SELECT... FROM... WHERE" statements. While SQL distribution naturally follows from covering typical BI questions and schema shapes, we also defined key SQL element distribution to ensure our system covers a wide range of SQL capabilities, such as:



- Basic aggregations (SUM, COUNT, MIN, MAX, AVG)

- Advanced aggregations (CORR, PERCENT_RANK, ...)

- CTE, Subqueries

- Window functions

- Join

- Currently, as we focus on single view with pre-joined data, this primarily involves self-joins. As we expand our scope to support cross-table joins, we'll start off my supporting star and snowflake schemas, and eventually expand to joins in arbitrary schema shapes.

AUTHOR



Renee Huang

SHARE

AUTHOR



Renee Huang

SHARE



Figure 3. Distribution of features in our internal evaluation set

Snowflake Cortex Analyst product feature evaluation

To ensure high accuracy and reliability of SQL generation, Cortex Analyst incorporates several key features that enhance its performance in real-world business intelligence tasks. It's critical to ensure the reliability of those features.

Reliable use of defined measures and filters

One of the standout features of Cortex Analyst is its semantic model, which guarantees that user-defined measures and filters are applied highly consistently in the generated SQL queries. This feature is crucial for maintaining accuracy across different use cases.

For example, in the benchmark semantic model, we define named filters to capture enterprise-specific jargon. For example, an enterprise might define a filter "North America" as following:

```
filters:
  - name: North America
    synonyms:
      - NA region
    expr: region in ("United States", "Canada",
"Mexico")
```

Then, for questions like, "What's my total revenue in North America in 2023?", the SQL generated by Cortex Analyst is considered correct only if it applies these definitions precisely.

```
SELECT sum(revenue)
FROM daily_revenue_by_region
WHERE region in ("United States", "Canada",
"Mexico")
and year(dt) = '2023'
```

AUTHOR



Renee Huang

This approach ensures that the SQL output aligns perfectly with the predefined business rules, providing users with trustworthy results.

Literal retrieval

Literal matching is another critical feature in Cortex Analyst that contributes to SQL accuracy. Users can include specific sample values within their queries, and the system ensures that these literals are correctly mapped and retrieved in the SQL output.

SHARE



For columns with high cardinality, it's unrealistic to include all distinct values as sample values. To address this, we introduced integration with Cortex Search, which allows Cortex Analyst to retrieve the correct literals for SQL generation by performing a semantic search over the data stored in those columns. This functionality is currently in private preview.

In our evaluation, we included questions that require exact literal matching, and the generated SQL was only considered accurate when it precisely matched these defined values.

Evaluation metrics and methodology

The Challenge: One question can have multiple correct answers

In text-to-SQL, one might assume that evaluating metrics is straightforward — just execute the gold query and compare it to the generated query for an exact match. This method, known as “execution accuracy,” is widely used in popular benchmarks.¹

However, this approach has significant limitations. It often rejects perfectly valid SQL queries simply because their execution results don't match the gold query exactly. In reality, many business questions can be answered in multiple valid ways.

For instance, consider the question: “Do we sell twice as many toys as books?”

One SQL query might return a simple “YES” or “NO” based on the revenue ratio. Another might output the exact revenues for both toys and books, along with the calculated ratio, allowing the end user to validate the result themselves. Both SQL queries would generate different outputs but answer the question correctly. In fact, many users might prefer the detailed output, as it provides more context and validation.

AUTHOR



Renee Huang

Example 1: Simple YES/NO Output

```
with toys_revenue as (  
  select sum(revenue) as total_toys_revenue  
  from daily_revenue where product = 'Toys'  
) ,  
books_revenue as (  
  select sum(revenue) as total_books_revenue  
  from daily_revenue where product = 'Books'
```

SHARE



```
SELECT  
  CASE  
    WHEN total_toys_revenue >= 2 *  
total_books_revenue THEN 'Yes'  
    ELSE 'No'  
  END AS is_toys_twice_books  
FROM  
  toys_revenue ,  
  books_revenue
```

Example 2: Outputting Revenues and Ratio

```
with toys_revenue as (  
  select sum(revenue) as total_toys_revenue  
  from daily_revenue where product = 'Toys'  
) ,  
books_revenue as (  
  select sum(revenue) as total_books_revenue  
  from daily_revenue where product = 'Books'  
)  
SELECT  
  total_toys_revenue ,  
  total_books_revenue ,  
  div0(total_toys_revenue, total_books_revenue) as  
ratio_toy_to_book
```

```
FROM
    toys_revenue,
    books_revenue
```

Snowflake evaluation approach: Scoring with multiple gold queries

AUTHOR



Renee Huang

To address these challenges, we experimented with several methods for evaluating SQL more flexibly, including using large language models (LLMs) as judges. However, we found that evaluating SQL can be as challenging as generating the correct SQL itself.

Given the limitations of strict execution accuracy and the challenges of using LLMs as judges, our team adopted a more iterative approach:

- 1. Diverse query generation:** We run evaluation questions across [Facebook](#), [LinkedIn](#), [Twitter](#), and [Email](#) language models to generate a broad search space of possible SQL queries.
- 2. Human-informed gold standards:** From this search space, human evaluators manually select all the correct SQL queries to add to our set of acceptable gold queries.
- 3. Flexible scoring:** Instead of requiring an exact match with the gold query, we calculate precision and recall for column matching, setting a more lenient threshold to judge accuracy. This approach focuses on meaningful matches rather than strict exactness.

As Cortex Analyst aims to address more diverse customer use cases, the risks associated with using a single-answer evaluation grow significantly, as it can lead to biased insights during model error analysis. By allowing for multiple correct answers, our evaluation methodology better reflects the nuances of real-world data analysis, providing a more accurate assessment of Cortex Analyst's capabilities.

Evaluation result examples

Here are some examples of evaluation results where Cortex Analyst outperforms the GPT-4o single-shot solution and the Other Solution we benchmarked against.

Business Question	Cortex Analyst	GPT-4o Single shot	Other Solution	Summary
-------------------	----------------	--------------------	----------------	---------

AUTHOR



Renee Huang

SHARE



Q1.Which region sold most on Christmas Day?	Succeeds	Fails	Succeeds	GPT-4o making assumptions of looking into Christmas Day for a random year. Cortex Analyst and Other Solution looks for the most recent year.
Q2.What was the DoD change in COGS during the 2nd week of May 2023, only looking at the North America region?	Succeeds	Fails	Fails	GPT-4o and Other Solution's generated queries are incorrect if the DATE column is nonconsecutive and has missing days.Cortex Analyst-generated query is correct regardless of DATE consecutiveness.
Q3 Show rolling maximum for forecasted revenue from electronics during the 4 weeks leading up to Christmas 2023.	Succeeds	Fails	Fails	GPT-4o and Other Solution's generated queries filtered on time range before calculating rolling window metric, causing incorrect numbers on the first 3 days of the time window.
Q4.Rank the product lines in terms of forecast accuracy.	Succeeds	Fails	Fails	GPT-4o and Other Solution were missing the client's specific metric definition.Cortex Analyst captures the specific metric definition with the semantic model.
Q5. For each week, what was the lowest daily cost of goods sold. show the week start date as well as the date that the low value occurred.	Succeeds	Succeeds	Fails	Other Solution doesn't follow the prompt and shows every day in the week with the min value repeated, making it impossible to know which day the min value occurred.

Business Question 1: Which region sold most on Christmas Day?

Outcome: Cortex Analyst and the Other Solution succeed by correctly identifying the most recent Christmas day, while GPT-4o fails by making an incorrect assumption.

AUTHOR



Renee Huang

The GPT-4o single-shot solution generated a query that picks Christmas Day in 2022, which could mislead users, as they are most likely interested in the most recent Christmas Day when not specified. Cortex Analyst, on the other hand, is designed to identify the most recent Christmas Day — aligning with probable user intent — and notes this assumption in the rewritten question.

Cortex Analyst Answer:

Here's our interpretation of your question: "Which region had the most revenue on Christmas Day in 2023?"



SHARE

```
SELECT
    sales_region,
    SUM(revenue) AS total_sales
FROM daily_revenue_by_region
WHERE
    date = '2023-12-25'
GROUP BY
    sales_region
ORDER BY
    total_sales DESC NULLS LAST
LIMIT 1;
```

Single-prompt GPT-4o Answer:

```
SELECT
    SALES_REGION,
    SUM(REVENUE) AS TOTAL_REVENUE
FROM DAILY_REVENUE_BY_REGION
WHERE DATE = '2022-12-25'
GROUP BY SALES_REGION ORDER BY TOTAL_REVENUE DESC
LIMIT 1;
```

Other Solution Answer:

```
SELECT
    SALES_REGION,
    SUM(REVENUE) AS total_revenue
FROM
    daily_revenue_by_region
WHERE
    DATE = '2023-12-25'
GROUP BY
    SALES_REGION
ORDER BY
    total_revenue DESC
LIMIT 1;
```

AUTHOR



Renee Huang

SHARE

f **LinkedIn** **Question 2: What was the DoD change in COGS during the 2nd week of May 2023, only looking at the North America region?**

Outcome: Cortex Analyst provides a correct solution regardless of nonconsecutive dates, while GPT-4o and Other Solution fail by producing a query that gives incorrect answers on nonconsecutive dates.

At first glance, the SQL generated by Cortex Analyst, GPT-4o and the Other Solution may all appear reasonable. However, if the “Date” column in the underlying data isn’t consecutive and has missing days, GPT-4o and the Other Solution’s use of the LAG function would be incorrect, as it doesn’t ensure comparison with the actual prior date.

This issue becomes even more problematic if the SQL only outputs the current date, without the previous date column. In such cases, the end user cannot verify if the calculation is correctly compared against the previous date, and could easily make decisions based on completely incorrect numbers.

Cortex Analyst ensures accurate calculations even with nonconsecutive dates by validating comparisons against the correct prior date.

Cortex Analyst Answer:

```
WITH daily_cogs AS (
    SELECT
        sales_region,
```

AUTHOR



Renee Huang

SHARE



```
        date,
        cogs
    FROM daily_revenue_by_region
    WHERE
        sales_region = 'North America' AND date >=
        '2023-05-08' AND date < '2023-05-15'
    )
    SELECT
        curr.date AS curr_date,
        prev.date AS prev_date,
        curr.cogs AS curr_cogs,
        prev.cogs AS prev_cogs,
        curr_cogs - prev_cogs AS dod_chg,
        (
            curr_cogs - prev_cogs
        ) / NULLIF(prev_cogs, 0) AS dod_pct_chg
    FROM daily_cogs AS curr
    LEFT JOIN daily_cogs AS prev
        ON (
            curr.date = prev.date + INTERVAL '1 DAY'
        )
    ORDER BY
        curr_date DESC NULLS LAST
```

Single-prompt GPT-4o Answer:

```
WITH cte AS (
    SELECT
        DATE, COGS,
        LAG(COGS) OVER (ORDER BY DATE) AS PREV_COGS
    FROM DAILY_REVENUE_BY_REGION
    WHERE SALES_REGION = 'North America' AND DATE
    BETWEEN '2023-05-08' AND '2023-05-14'
)
SELECT
    DATE, COGS, PREV_COGS,
    (COGS - PREV_COGS) / NULLIF(PREV_COGS, 0) * 100
    AS DOD_CHANGE_PERCENT
FROM cte
    WHERE NOT PREV_COGS IS NULL ORDER BY DATE;
```

Other Solution Answer:

AUTHOR



Renee Huang

SHARE



```
WITH CogsCalc AS (  
    SELECT  
        DATE,  
        COGS,  
        LAG(COGS) OVER (  
            ORDER BY  
                DATE  
        ) AS previous_day_cogs,  
        COGS - LAG(COGS) OVER (  
            ORDER BY  
                DATE  
        ) AS cogs_change  
    FROM  
        daily_revenue_by_region  
        SALES_REGION ILIKE '%North America%'  
        AND DATE >= '2023-05-08'  
        AND DATE <= '2023-05-14'  
)  
SELECT  
    DATE,  
    cogs_change  
FROM  
    CogsCalc  
ORDER BY  
    DATE
```

Business Question 3: Show me the 3 day rolling maximum for forecasted revenue from electronics during the 4 weeks leading up to Christmas 2023.

Outcome: Cortex Analyst succeeds by calculating the rolling maximum before filtering the date range, while GPT-4o and the Other Solution fail by incorrectly applying the date filter first.

This shows another example of very similar-looking calculations giving different results. For *rolling window* questions, the sequence of operations — whether filtering the date range before or after applying the rolling window function — is crucial.

In this example, GPT-4o incorrectly filters the date range first, leading to incorrect values for the first three days of the window. Cortex Analyst, however, performs the rolling calculation first and then applies the date filter, retrieving accurate results.

Cortex Analyst Answer (Calculate rolling max first, then filter to desired range):

AUTHOR



Renee Huang

SHARE



```
WITH rolling_max AS (  
  SELECT  
    date,  
    forecasted_revenue,  
    MAX(forecasted_revenue) OVER (ORDER BY date  
ROWS BETWEEN 2 PRECEDING AND CURRENT ROW) AS  
rolling_max_3d  
  FROM daily_revenue_by_product  
  WHERE  
    product_line = 'Electronics'  
)  
SELECT  
  date,  
  rolling_max_3d  
FROM rolling_max  
WHERE  
  date BETWEEN CAST('2023-12-25' AS DATE) -  
INTERVAL '4 WEEKS' AND CAST('2023-12-25' AS DATE)  
ORDER BY  
  date DESC NULLS LAST;
```

Single-prompt GPT-4o Answer (Filter to date range first, then rolling max):

```
WITH FilteredData AS (  
  SELECT  
    DATE, FORECASTED_REVENUE  
  FROM  
    SNOWPILOT_SYNTHETIC_DATA.REVENUE_TIMESERIES.DAILY_REVENUE_BY_PRODUCT  
  WHERE PRODUCT_LINE = 'Electronics' AND DATE  
    BETWEEN '2023-11-27' AND '2023-12-24'  
)  
, RollingMax AS (  
  SELECT
```



```

DATE, FORECASTED_REVENUE,
MAX(FORECASTED_REVENUE) OVER (ORDER BY DATE ROWS
BETWEEN 2 PRECEDING AND CURRENT ROW) AS
ROLLING_MAX_FORECASTED_REVENUE
    FROM FilteredData
)
SELECT DATE, ROLLING_MAX_FORECASTED_REVENUE FROM
RollingMax ORDER BY DATE;

```

AUTHOR



Renee Huang

SHARE



Other Solution Answer:

```

WITH DateRange AS (
    SELECT
        DATE_SUB('2023-12-25', 27) AS start_date,
        '2023-12-25' AS end_date
)
RollingMax AS (
    SELECT
        DATE,
        MAX(FORECASTED_REVENUE) OVER (
            ORDER BY
                DATE ROWS BETWEEN 2 PRECEDING
                AND CURRENT ROW
        ) AS rolling_max_forecasted_revenue
    FROM
        daily_revenue_by_product,
        DateRange
    WHERE
        DATE BETWEEN start_date
        AND end_date
        AND PRODUCT_LINE ILIKE '%Electronics%'
)
SELECT
    DATE,
    rolling_max_forecasted_revenue
FROM
    RollingMax
ORDER BY
    DATE;

```

Business Question 4: Rank the product lines in terms of forecast accuracy.

Outcome: Cortex Analyst succeeds by aligning with the user's specific metric definition, while GPT-4o and the Other Solution fail by making incorrect assumptions about the metric.

AUTHOR



Renee Huang

When it comes to ranking product lines based on forecast accuracy, the definition of “accuracy” can vary. In this case, Cortex Analyst uses a user-defined semantic model to correctly calculate accuracy based on absolute differences, as specified. GPT-4o and the Other Solution, however, attempt to infer the metric on their own, resulting in a calculation based on percentage differences, which may not align with user expectations.

SHARE



```
measures:
  - name: avg_forecast_error
    formula: AVG(ABS(forecasted_revenue - revenue))
    synonyms: ['forecast accuracy']
    data_type: number
```

Cortex Analyst Answer:

```
WITH accuracy as (
SELECT
    product_line,
    AVG(ABS(forecasted_revenue - revenue)) AS
    avg_forecast_error
FROM daily_revenue_by_product)
GROUP BY
    product_line
)
select product_line,
        avg_forecast_error,
        rank() over (order by avg_forecast_error
asc) as rnk
from accuracy
ORDER BY
    avg_forecast_error ASC;
```

Single-prompt GPT-4o Answer:

AUTHOR



Renee Huang

```
WITH forecast_accuracy AS (  
    SELECT  
    PRODUCT_LINE,  
    AVG(ABS(FORECASTED_REVENUE - REVENUE) /  
    NULLIF(REVENUE, 0)) AS MAPE  
    FROM  
    SNOWPILOT_SYNTHETIC_DATA.REVENUE_TIMESERIES.DAILY_REVENUE_BY_PRODUCT  
    GROUP BY PRODUCT_LINE  
)  
SELECT PRODUCT_LINE, MAPE, RANK() OVER (ORDER BY  
MAPE ASC) AS ACCURACY_RANK  
FROM forecast_accuracy ORDER BY ACCURACY_RANK;
```

SHARE

f n Answer:

```
SELECT  
    PRODUCT_LINE,  
    AVG(ABS(FORECASTED_REVENUE - REVENUE) / REVENUE)  
AS forecast_accuracy  
FROM  
    daily_revenue_by_product  
GROUP BY  
    PRODUCT_LINE  
ORDER BY  
    forecast_accuracy;
```

Business Question 5: For each week, what was the lowest daily cost of goods sold. Show the week start date, as well as the date that the low value occurred.

Outcome: Cortex Analyst and GPT-4o succeed by finding min COGS date for each week and the corresponding date when the min COGS value occurred, whereas the Other Solution fails because it calculates the min COGS value for each week but does not identify which date within the week it occurred.

Cortex Analyst Answer:

AUTHOR



Renee Huang

SHARE



```
WITH weekly_min_cogs AS (  
  SELECT  
    DATE_TRUNC('WEEK', date) AS week_start_date,  
    MIN(cogs) AS min_cogs  
  FROM  
    snowpilot_synthetic_data.revenue_timeseries.daily_revenue  
  GROUP BY  
    DATE_TRUNC('WEEK', date)  
)  
SELECT  
  wmc.week_start_date,  
  wmc.min_cogs,  
  dr.date AS min_cogs_date  
FROM weekly_min_cogs AS wmc  
JOIN snowpilot_synthetic_data.revenue_timeseries.daily_revenue AS dr  
  ON wmc.week_start_date = DATE_TRUNC('WEEK',  
  dr.date)  
  AND wmc.min_cogs = dr.cogs  
ORDER BY  
  wmc.week_start_date DESC NULLS LAST;
```

Single-prompt GPT-4o Answer:

```
WITH WEEKLY_COGS AS (  
  SELECT  
    DATE_TRUNC('WEEK', DATE) AS  
WEEK_START_DATE,  
    DATE,  
    COGS,  
    ROW_NUMBER() OVER (  
      PARTITION BY DATE_TRUNC('WEEK', DATE)  
      ORDER BY  
        COGS ASC  
    ) AS RN  
  FROM  
  
  SNOWPILOT_SYNTHETIC_DATA.REVENUE_TIMESERIES.DAILY_REVENUE
```

```

)
SELECT
    WEEK_START_DATE,
    DATE AS LOW_COGS_DATE,
    COGS AS LOW_COGS
FROM
    WEEKLY_COGS
WHERE
    RN = 1
ORDER BY
    WEEK_START_DATE;

```

AUTHOR



Renee Huang

Other Solution Answer:

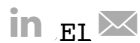
SHARE



```

WITH WeeklyCOGS AS (
    SELECT
        DATE_TRUNC('week', DATE) AS week_start_date,
        DATE,
        MIN(COGS) OVER (PARTITION BY DATE_TRUNC('week',
        DATE)) AS weekly_min_cogs
    FROM
        daily_revenue
)
SELECT
    DISTINCT week_start_date,
    DATE AS low_cogs_date,
    weekly_min_cogs
FROM
    WeeklyCOGS
ORDER BY
    week_start_date,
    low_cogs_date;

```



Conclusion

To summarize, in this blog we discussed our approach to better reflect real-world text-to-SQL challenges, both by constructing an internal evaluation benchmark and incorporating multiple gold queries to enhance the evaluation methodology. Additionally, the Spider team has announced that Spider 2.0-SQL is coming out soon and will be much

more realistic and challenging than Spider 1.0. We'll aim to evaluate our system on the Spider 2.0 data set when it's available.

Interested in trying out Snowflake Cortex Analyst? Build your first Cortex Analyst-powered chat app using this [quickstart guide](#) today!

¹ See [original Spider paper](#) on evaluation metric.

Forward-Looking Statements

This contains express and implied forward-looking statements, including statements regarding (i) Snowflake's business strategy, (ii) Snowflake's products, services, and technology offerings, including those that are under development or not generally available, (iii) market growth, trends, and competitive considerations, and (iv) the integration, interoperability, and availability of Snowflake's products with and on third-party platforms. These forward-looking statements are subject to a number of risks, uncertainties, and assumptions, including those described under the heading "Risk Factors" and elsewhere in the Quarterly Reports on Form 10-Q and Annual Reports of Form 10-K that Snowflake files with the Securities and Exchange Commission. In light of these risks, uncertainties, and assumptions, actual results could differ materially and adversely from those anticipated or implied in the forward-looking statements. As a result, you should not rely on any forward-looking statements as predictions of future events.

SHARE

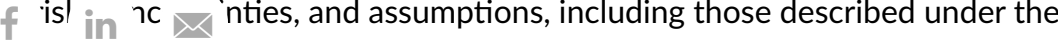


AUTHOR



Renee Huang

SHARE



RELATED CONTENT

2024년 08월 14일	2024년 08월 08일	2024년 08월 20일

Snowflake Cortex Analyst: Behind the Scenes

Building a conversational self-service analytics product for business users is a complex and challenging endeavor. Trust and accuracy have to be at the core of such a product, as business...

[Have a look](#)

Snowflake Cortex Search: High-Quality, Performant Search and Retrieval for Enterprise AI

Search and retrieval systems have always been a critical backbone for knowledge management in enterprises....

[Expand your knowledge](#)

Secure Connections with New Outbound Private Link with Snowflake Support in Preview

For various data engineering, AI and ML workloads, customers need to connect to external systems...

[Find Out How](#)

SHARE



START YOUR 30-DAY FREE TRIAL

[START NOW](#)

 Snowflake Inc.

플랫폼 개요

아키텍처

데이터 애플리케이션

데이터 마켓플레이스

SNOWFLAKE
파트너 네트워크

지원 및 서비스

회사

문의하기

**Sign up for
Snowflake
Communications**

diana.shaw@snow United States

By submitting this form, I understand Snowflake will process my personal information in accordance with their **Privacy Notice**. Additionally, I consent to my information being shared with Event Partners in accordance with Snowflake's **Event Privacy Notice**. I understand I may withdraw my consent or update my preferences **here** at any time.

[Privacy Notice](#) | [Site Terms](#) | [Cookie Settings](#)

© 2024 Snowflake Inc. All Rights Reserved

AUTHOR



Renee Huang



SHARE

