How can we enable business users to get to “actionable insights” fast?

Prof. Aditya Parameswaran

Anshita Arya, Anukriti Goyal, Melissa Licari, Molly Zhang, Richa Verma
Business users have many data questions...

“Is any of my customer having more “Active Users” than usual?”

“What’s the last time this customer paid an invoice?”

“Is my metric really going up?”
But... rather than getting them answered.. They get “triaged”
And, data teams hate answering these one-off questions.

Which means that these queries get unanswered.

Richa, Product Manager  4:22 PM
Hey, can you tell me the number of opportunities broken down by city?

Molly, Data Analyst  4:24 PM
Sorry, unfortunately, we have a backlog for 2-weeks, can you look at this dashboard: bad-dashboard-that-doesnt-work.tableau.com/c/random-product-insights-no-one-want. It might help!
82%

Data Projects Fail

(Gartner, 2019, 2020, 2021)
Hence, we created “DashQueries”

A SlackBot that helps answer business questions in seconds, rather than days.

Here’s the chart I got for you: https://dashqueries.s3.us-west-2.amazonaws.com/chart-1683139745.png

<table>
<thead>
<tr>
<th>City</th>
<th>Avg Sales Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bengaluru</td>
<td>43,890</td>
</tr>
<tr>
<td>Chennai</td>
<td>44.2324</td>
</tr>
<tr>
<td>Delhi</td>
<td>49,9489</td>
</tr>
<tr>
<td>Hyderabad</td>
<td>44,1286</td>
</tr>
<tr>
<td>Kolkata</td>
<td>42,9662</td>
</tr>
<tr>
<td>Mumbai</td>
<td>45,1368</td>
</tr>
<tr>
<td>Pune</td>
<td>39,364</td>
</tr>
</tbody>
</table>

And the SQL I used:

```
SELECT city, AVG(sales.velocity) AS avg_sales_velocity
FROM sales_data
GROUP BY city
ORDER BY avg_sales_velocity DESC;
```

Which basically means this in plain English:

---This query will calculate the average sales velocity by city

Are you satisfied with this answer?

Looks good | Data seems wrong | Query seems wrong
And, we used “Praxis” method to understand...

1. Will business users’ actually trust a natural language query based system to make decisions?
2. What are the querying / exploration /scalability limitations of such a tool?

- 300+ queries ran.
- 15 user tests in two phases.
- 40+ informal demos in conferences, etc.
Live Demo

I am about to show you how it's done.
How DashQueries work

1. Interpret the Query & convert it into a prompt
2. Convert to SQL
3. Run SQL on relevant dataset
4. Send results to Slack

What is the number of opportunities by city?

Here is the number of opportunity by city

Get an answer.
Phase 1

Get the bot out of door
Created a simple LLM pipeline
A working slackbot with basic UX

- Tested with 5 users
- And, internal testing ~100 questions

Questions asked:

- Can you actually tell me the number of opportunities that have sales velocity more than 70 and client revenue more than 1M?
- Tell me how many opportunities were won in Delhi that had an opportunity size of more than 60K
- Why are deals won in Delhi more than Mumbai?
- Where are the largest opportunity sizes
- How many clients have done repeated sales with

---

Richa Verma 1:25 AM
@DashQueries tell me which city has the most opportunities won?

DashQueries 1:25 AM
Runnin' the numbers, gimme a sec.

I am trying my best to answer your questions: tell me which city has the most opportunities won?

How I work?

It's simple, I take your question and interpret (1) table you are curious about, and (2) write an SQL query for it. Once the results are ready, I get them back to you.

Kind of questions I can answer:
1. Tell me whether sales have increased in a <region>?
2. Which opportunities were won/lost?
3. Tell me the average sales for the last quarter/worth?
4. Tell me which product has been selling the most?

Questions I can't answer:
1. Why sales changed?
2. Predict the sales in future...
3. What's the purpose of life?

In the meantime, go back to your work, I'll ping you in this channel once the answer is ready.

Here's the answer I got:

<table>
<thead>
<tr>
<th>city</th>
<th>opportunities won</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mumbai</td>
<td>5333</td>
</tr>
</tbody>
</table>

And the SQL I used:

```sql
SELECT city, COUNT(*) AS OpportunitiesWon
FROM sales_data
WHERE opportunity_status ILIKE 'won'
GROUP BY city
ORDER BY OpportunitiesWon DESC NULLS LAST
LIMIT 1;
```

Which basically means this in plain English:

---

This query returns the city with the most Opportunities Won.

Are you satisfied with this answer? (Yes or No)
What did we learn?

- **Questions answered**: 89% (n = 69)
- **Satisfactory answers**: 72% (n = 62)

**How complex were the queries?**
- Aggregations: 60
- Filter: 50
- Order: 40
- Group by: 30
- Joins: 20

**What kind of errors did the bot produce?**

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wrong Aggregation</td>
<td>5</td>
</tr>
<tr>
<td>Wrong Variable Selection</td>
<td>4</td>
</tr>
<tr>
<td>Categorical Parsing</td>
<td>3</td>
</tr>
<tr>
<td>Group By</td>
<td>2</td>
</tr>
<tr>
<td>No Answer</td>
<td>2</td>
</tr>
<tr>
<td>Wrong Interpretation</td>
<td>2</td>
</tr>
<tr>
<td>Wrong Data Format</td>
<td>1</td>
</tr>
<tr>
<td>Wrong join</td>
<td>1</td>
</tr>
<tr>
<td>Deduplication</td>
<td>1</td>
</tr>
<tr>
<td>System Design Error</td>
<td>1</td>
</tr>
<tr>
<td>Wrong Filter</td>
<td>1</td>
</tr>
<tr>
<td>Result can be better</td>
<td></td>
</tr>
</tbody>
</table>

**Making assumptions**: 70% (n = 10)

**Will the users trust the bot?**

- Depends on how accurately it answers the questions and produces results.
  - n = 12

- n = 29/69
This was us...

60% of the time it works every time
What did we learn?

Trust Increased With Usage

- First impressions were key to building trust
- Data users were more skeptical than non-technical users
- “I like that it shows me the SQL code so that I can run it myself to check it”

Users Want a Feedback Loop

- Some responses were not optimal/what the user was looking for and the user would have to rephrase their question for a better response.
- Users wanted a way to tell us whether the response was satisfactory or not.
Phase 2

Get the bot out of door
Made our bot more robust
Improvements

1. UX Improvements
   a. Number and frequency of slack messages bot sent
   b. Feedback loop (Looks Good, Wrong Data, Wrong Query)
   c. Chart response
   d. Length of the results table

2. Result Quality improvements
   a. Categorical parsing
   b. Recognition of previous vs. current query
   c. Error fixes
A responsive bot that handles data

- Tested with 10 users
- And, internal testing ~100 questions

Questions asked:

- Please give me a line graph of the client revenue sizing compared to the client employee sizing
- What are my biggest revenue streams?
- Tell me about year to year growth of opportunities?
- How much business was done last year in all cities?
- In which technology do we have the highest potential for revenue increase
What did we learn?

Query Analysis

- Specific queries yielded better results
- Most errors were wrong variable selection
- Performed well for complex queries
- Slow response time, but faster than asking a person
Learning & Conclusion

Diverse User Personas
Learning & Conclusion

Building Trust

C'mon son
Learning & Conclusion

Capabilities of LLM
Future Work

Interactive Feedback
Future Work

Semantic Context

CONTEXTTTT!! SET THE CONTEXT FIRST!
Future Work

Extension to BI, Productivity Tools (Zoom, Meet, Email)
Questions?