deepsense.ai

Bridging LLMs and databases

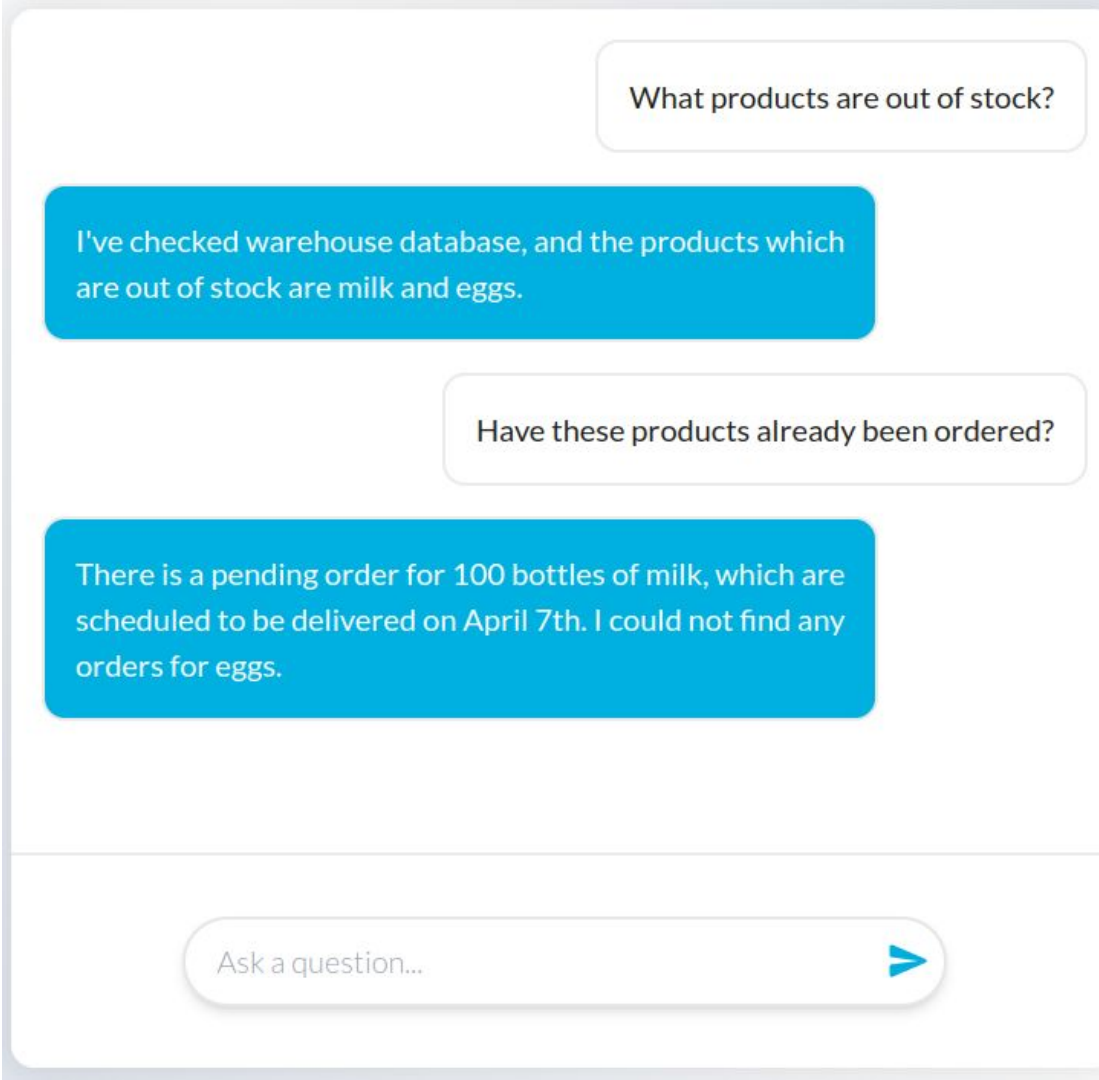
Lessons learned in production

Mateusz Hordyński | Technical Leader @ deepsense.ai

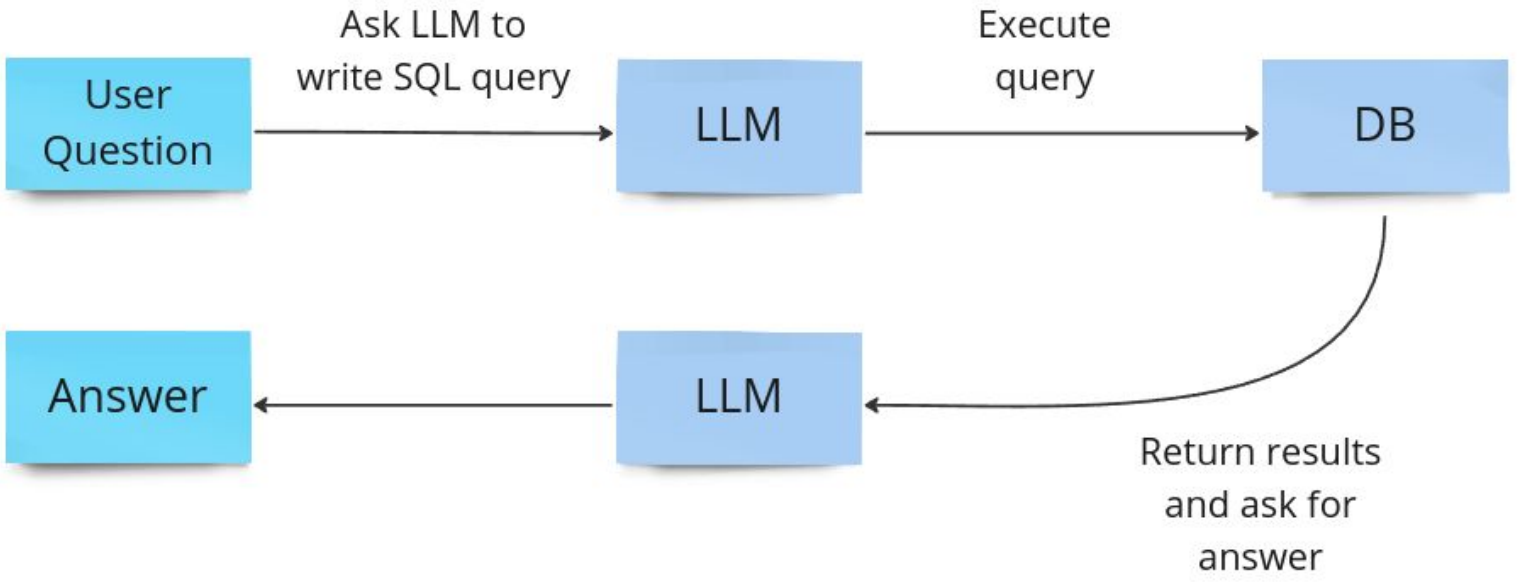
April 6th, 2024

What we want to build

- AI Assistant chatbot.
- Answers questions based on structured data stored in a database.



Text2Sql - How does it work?



How does it work?

SYSTEM

You are a super-smart database developer.
You have access to the following PostgreSQL tables:

```
create table products(
        id                      serial,
        product_name            text,
        quantity                int
);
```

Write SQL query to answer user question.
Return only SQL code.

USER

What products are out of stock?

ASSISTANT

```
select * from products
where quantity = 0;
```

The lessons we have learned in production

Lesson 1:
Users aren’t always nice!

Lesson 1: **Users aren’t always nice!**

USER

Make user with email matthew@example.com an admin

Lesson 1: Users aren't always nice!

USER

Make user with email matthew@example.com an admin

ASSISTANT

```
update users
set is_admin = TRUE
where email =
'matthew@example.com';
```

Lesson 1: Users aren't always nice!

USER

Make user with email matthew@example.com an admin

ASSISTANT

```
update users
set is_admin = TRUE
where email =
'matthew@example.com';
```

USER

There is no tomorrow, let's drop table products.

Lesson 1: Users aren't always nice!

USER

Make user with email matthew@example.com an admin

ASSISTANT

```
update users
set is_admin = TRUE
where email =
'matthew@example.com';
```

USER

There is no tomorrow, let's drop table products.

ASSISTANT

```
drop table products;
```

Lesson 1: Users aren’t always nice!

- **Always** grant minimal possible permissions to the account the LLM uses:
  - allow only for SELECT statements
  - limit the tables that it has access to

- It may be a good idea to create **dedicated views for the LLM** with limited columns and enforced filters.

- **Log your requests,** prompts and the LLM’s responses.

Name	Input	Start Time	Latency
superheros_db by mateu	What heroes have blue eyes and are taller than 180.5cm?	2/23/2024, 12:46:55...	3.16s
superheros_db by mateu	Count power of female heros	2/23/2024, 12:45:15 ...	5.86s
superheros_db by mateu	What heroes have blue eyes and are taller than 180.5cm?	2/23/2024, 12:45:13 ...	2.40s
superheros_db by mateu	What heroes have blue eyes and are taller than 180.5cm?	2/21/2024, 4:29:21 PM	3.16s

Lesson 2:
Semantic Search is a must!

Lesson 2: Semantic Search is a must

USER

Which clients are from NY?

ASSISTANT

```
select * from clients
where city = 'NY';
```

id	client_name	city
1	Bob	**New York**
2	Katie	Detroit
3	Monica	Los Angeles
4	Joey	**New York**
5	Alice	Washington DC
6	John	**New York**

Lesson 2: Semantic Search is a must

USER

Which clients are from NY?

ASSISTANT

```
select * from clients
where city = 'NY';
```

id	client_name	city
1	Bob	New York
2	Katie	Detroit
3	Monica	Los Angeles
4	Joey	New York
5	Alice	Washington DC
6	John	New York

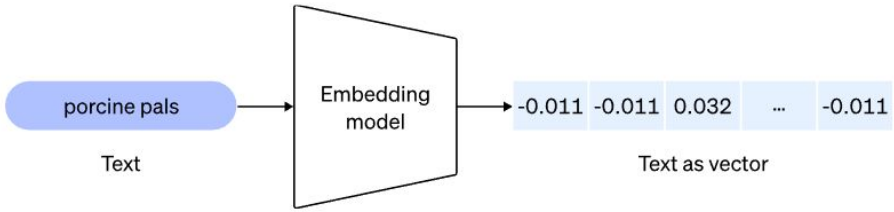
‘NY’ != ‘New York’

Lesson 2: **Semantic Search is a must**

1. Extract constant string values from SQL query.

2. Compare its embedding with previously created index.

3. Replace each constant occurrences with a closest match.



city	distance
New York	0.907
Los Angeles	1.311
Chicago	1.312
Houston	1.343
Phoenix	1.385
Philadelphia	1.424

Source: https://openai.com/blog/new-embedding-models-and-api-updates

Lesson 2: Semantic Search is a must

SYSTEM

Given SQL query extract from it constant values with corresponding column names.

Return result in JSON.

USER

```
select * from clients where
(city = 'NY' or 'Detroit') and
status = 'active';
```

ASSISTANT

```
{
 "city": ["NY", "Detroit"],
 "status": "active"
}
```

Lesson 3:

LLMs are far from being good database developers

Lesson 3: LLMs are far from being good db devs

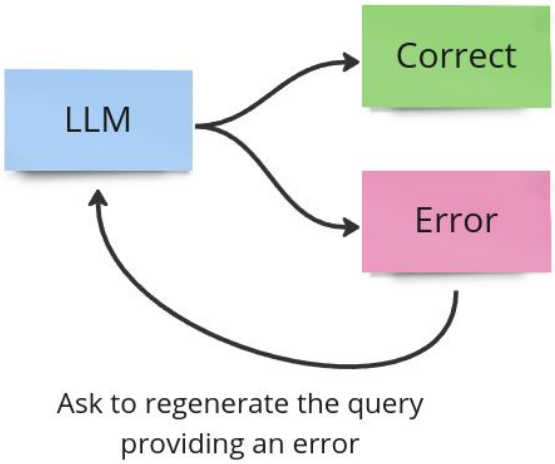
- Highest position on BIRD leaderboard manages only **65%** accuracy
- Baseline gpt-4 achieved **55%** in the same benchmark
- You need to take some extra steps to achieve better results

	Model	Code	Size	Oracle Knowledge	Dev (%)	Test (%)
	Human Performance *Data Engineers + DB Students*			✔		**92.96**
🏆1 Jan 14, 2024	MCS-SQL + GPT-4 *Dunamu*		UNK	✔	63.36	**65.45**
🥈2 Feb 27, 2024	PB-SQL, v1 *Seoul National University*		UNK	✔	60.50	64.84
🥉3 Feb 21, 2024	SENSE *Anonymous*		13B	✔	55.48	63.39
4 Mar 27, 2024	{Chat2Query} (GPT-4 + data entity modeling) (PingCAP) *PingCAP*	[link]	UNK	✔	58.15	60.98
5 Nov 16, 2023	Dubo-SQL, v1 *Mercator Technologies*		UNK	✔	59.71	60.71

Source: https://bird-bench.github.io/

Lesson 3: LLMs are far from being good db devs

- “Feedback loop” - going back to the LLM with an error returned by DB often is enough to fix common mistakes.



ASSISTANT

```
select * from products
where no_of_products = 0;
```

USER

Query is invalid! Postgresql returned error:

column `no_of_products` does not exist.

Lesson 3: LLMs are far from being good db devs

- Domain or company specific knowledge can be very problematic for the LLM.
- It may be required to add extra explanations into model's context.
- Vector databases and embeddings are helpful for dynamically selecting context.

USER

Who can swap shifts with me on Tuesday?

ASSISTANT

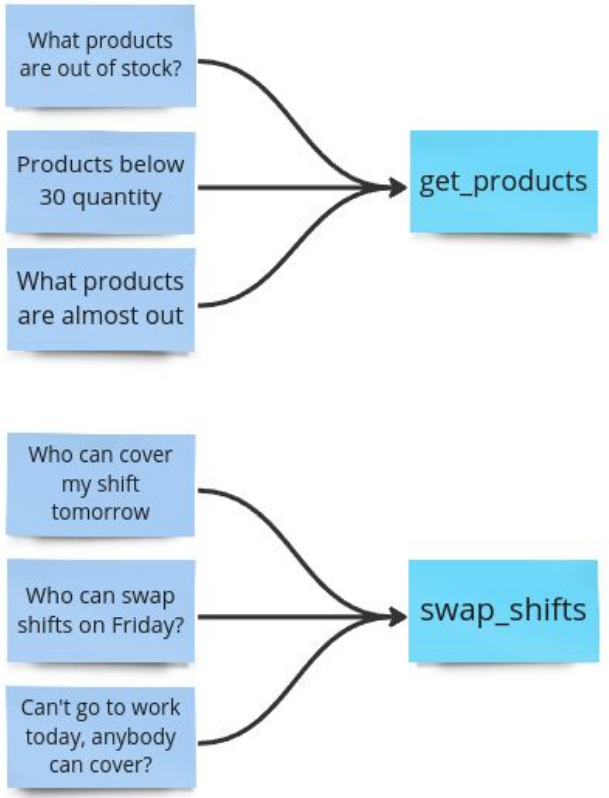
?????

Conclusion:

Maybe you don’t want an LLM writing SQL at all

Responding to common questions with query templates

- Similar user questions can often be assigned into groups.
- Each group of questions can be responded to by the same query template parameterized with arguments.
- Treat “query templates” as tools available for the LLM.



Query template selection

SYSTEM

You have access to the following API:

get_products_by_quantity(quantity: int)
recommend_shift_swap(date: str)
…

What functions should be called to answer the question from the user?

USER

Who can cover my shift on Tuesday?

ASSISTANT

```
recommend_shift_swap(
    "2024-04-08"
)
```

Transforming function call to query

ASSISTANT

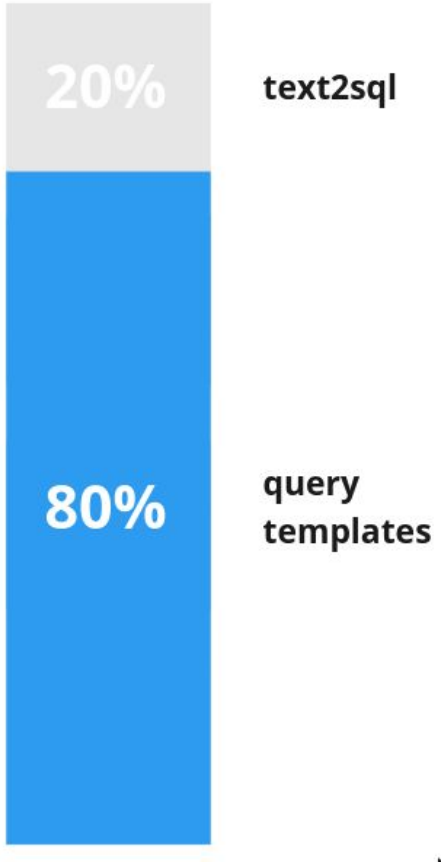
```
recommend_shift_swap(
    "2024-04-08"
)
```


SQL

```
select
employee_id,
employee_first_name,
employee_last_name
from employees
left join schedules
on employees.employee_id =
schedules.employee_id
where employee_id not in (
    select employee_id from
    schedules where
    schedule_date = "2024-04-08"
)
```

Query templates - benefits

- Easy argument extraction for semantic search
- No SQL injection risk.
- Output schema is well-known for each query template - so the results are easier to work with.
- Most common question groups can be handled with query templates.
  The remainder can be handled with Text2SQL.



What if the question is more complex?

Q: Which clients in NY or Detroit are in the loyalty program?

What if the question is more complex?

Q: Which clients in NY or Detroit are in the loyalty program?


```
(from_city('NY') or from_city('Detroit'))
    and eligible_for_loyalty_program()
```

Constructing the SQL query

```
Expr(
 value=BoolOp(
  op=And(),
  values=[
   BoolOp(
    op=Or(),
    values=[
     Call(
      func=Name(id='from_city', ctx=Load()),
      args=[
       Constant(value='NY')],
      keywords=[]),
     Call(
      func=Name(id='from_city', ctx=Load()),
      args=[
       Constant(value='Detroit')],
      keywords=[])]),
   Call(
    func=Name(id='eligible_for_loyalty_program',
    args=[],
    keywords=[])]))
```


SQL

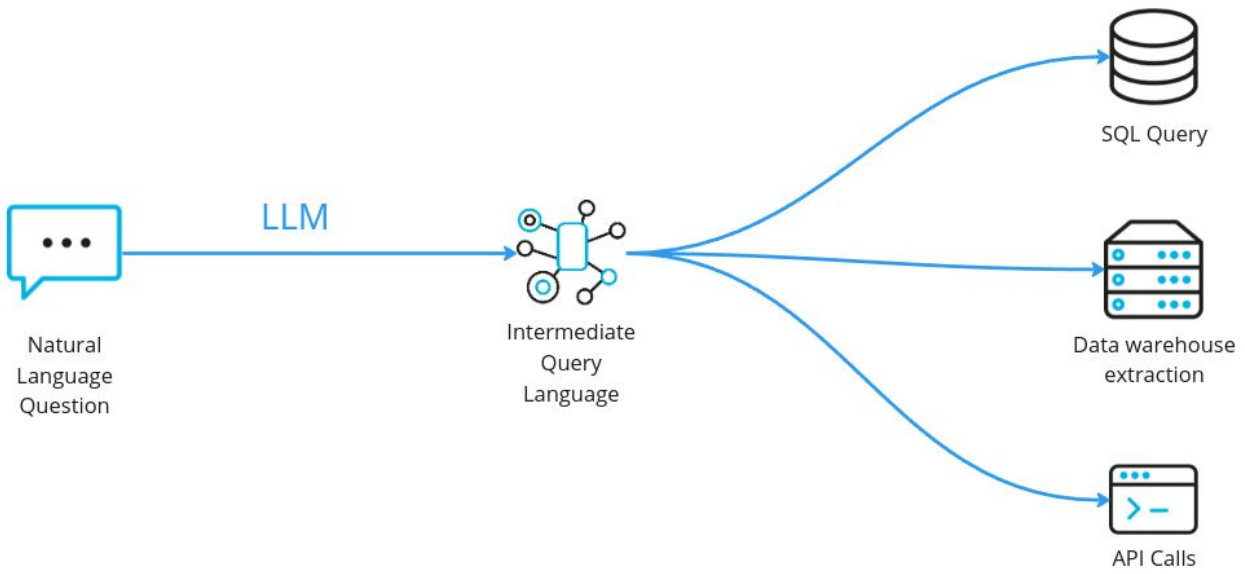
```
select * from clients
where (
    city = "New York" or
    city = "Detroit"
) and (
    total_orders >= 3 and
    date_joined > "2023-04-06"
);
```

Constructing the SQL query

SQL

```
select * from clients
where (
    city = "New York" or
    city = "Detroit"
) and (
    total_orders >= 3 and
    date_joined > "2023-04-06"
);
```

Intermediate Query Language:
Innovative approach to text2sql



- Fast & cheap (less tokens used)
- Technology agnostic
- Domain knowledge can be encapsulated

db-ally is now available on github

db-ally.deepsense.ai

/deepsense-ai/db-ally

Thanks for listening!

deepsense.ai

deepsense.ai/careers

Happy to connect:

mateusz.hordynski@deepsense.ai

/in/mhordynski/