SPECIAL ANNIVERSARY EDITION
LET'S MEET FOR THE 10TH TIME!

bigdata TECHNOLOGY WARSAW SUMMIT

EVENTION CZAS ZAANGAŻOWANY | getindata Part of Xebia

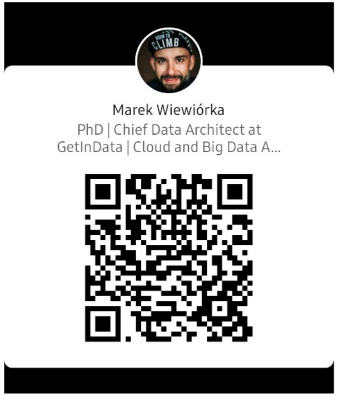
APRIL 10TH-11TH 2024

Your personal LLM and RAG-backed Data Copilot – lessons learned

Marek Wiewiórka, Phd

GetInData | Part of Xebia

About me



- Chief Data Architect @**GetInData | Part of Xebia**
- PhD(2023), Research Assistant at the Warsaw University of Technology
- Personally, a keen long-distance runner and gravel bike enthusiast

...how to turn best practices into AI coding assistant

1. Why do we need yet another (open-source) Copilot?
2. How can we build one?
3. Architecture and evaluation
4. DEMO

(Data) Context is king!

getindata
Part of Xebia

- *Explicit* and *precise* data context of your whole data platform
- Data transformation codebase
- Data models with comments and table relationships
- Other user queries
- Lineage and human curated dataset descriptions from data catalogs

Customized and specialized models are the future.

getindata
Part of Xebia

OpenAI

We believe that in the future, the vast majority of organizations will develop customized models that are personalized to their industry, business, or use case. With a variety of techniques available to build a custom model, organizations of all sizes can develop personalized models to realize more meaningful, specific impact from their AI implementations. The key is to clearly scope the use case, design and implement evaluation systems, choose the right techniques, and be prepared to iterate over time for the model to reach optimal performance.

With OpenAI, most organizations can see meaningful results quickly with the self-serve fine-tuning API. For any organizations that need to more deeply fine-tune their models or imbue new, domain-specific knowledge into the model, our Custom Model programs can help.

databricks   Why Databricks   Product   Solutions   Resources   About   DATA + AI SUMMIT

Build high-quality generative AI applications with DBRX customized for your unique data

by Jonathan Frankle, Ali Ghodsi, Naveen Rao, Hanlin Tang, Abhinav Venigalla and Matei Zaharia

March 27, 2024 in Company Blog

Share this post

Databricks' mission is to deliver data intelligence to every enterprise by allowing organizations to understand and use their unique data to build their own AI systems. Today, we are excited to advance our mission by open sourcing DBRX, a general purpose large language model (LLM) built by our Mosaic Research team that outperforms all established open source models on standard benchmarks. We believe that pushing the boundary of open source models enables generative AI for all enterprises that is customizable and transparent.

sqlcoder-7b and others

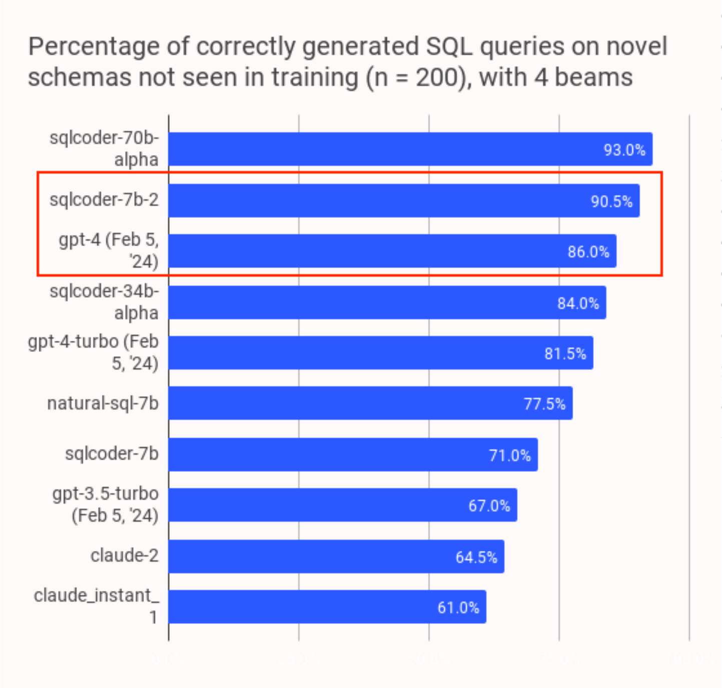
- Many other small (7-34b) models licensed for commercial use, e.g. :

✓ starcoder2

✓ dolphincoder

✓ deepseeek-coder

✓ opencodeinterpreter



When quantized can be even run locally!

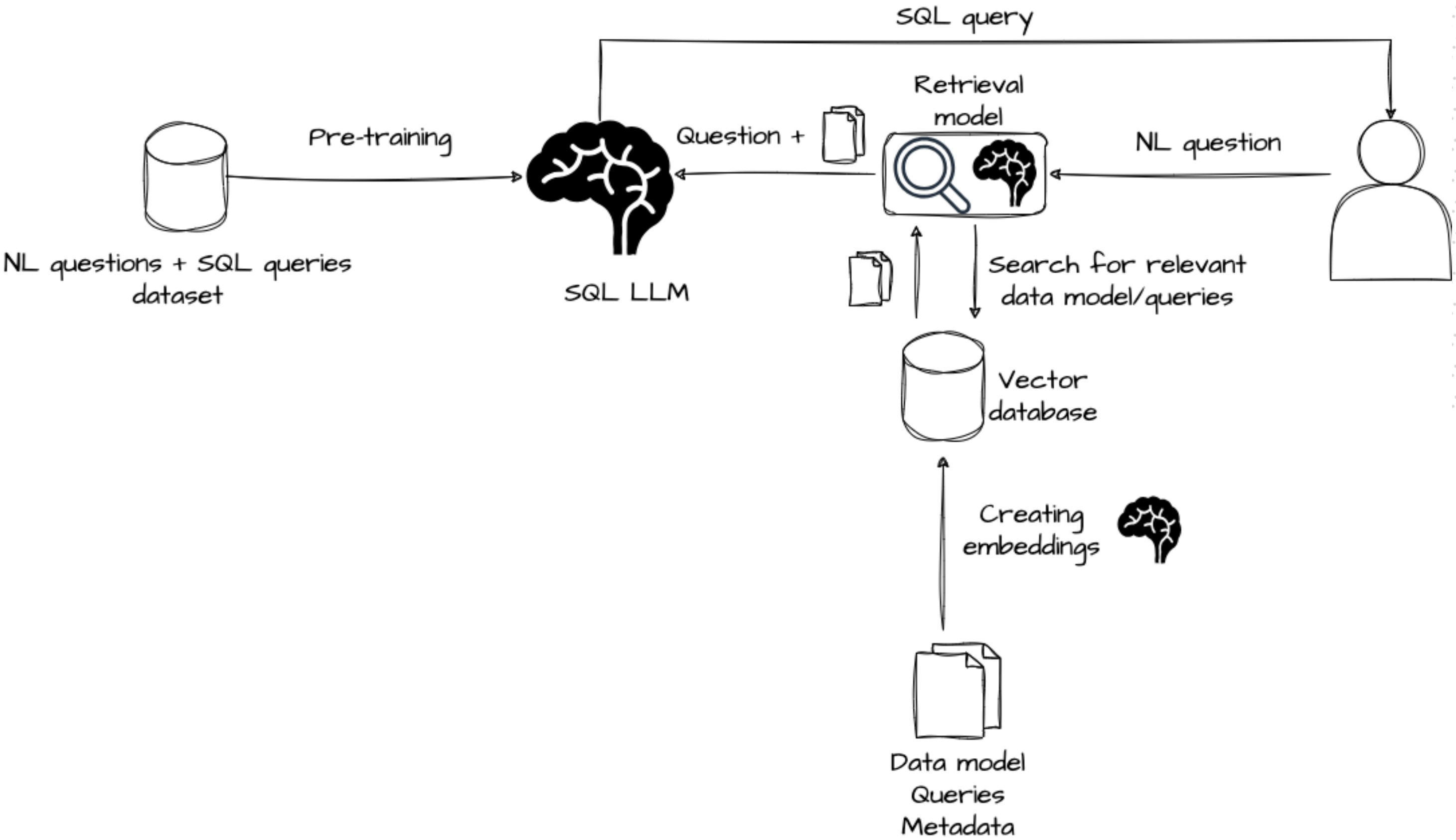
How turn your best practices into Copilots ?

- Vector database as a *knowledge base* - what ?
- Prompts as instructions following *best practices* - how ?
- LLM to *combine* both...

Retrieval-Augmented Generation(RAG)

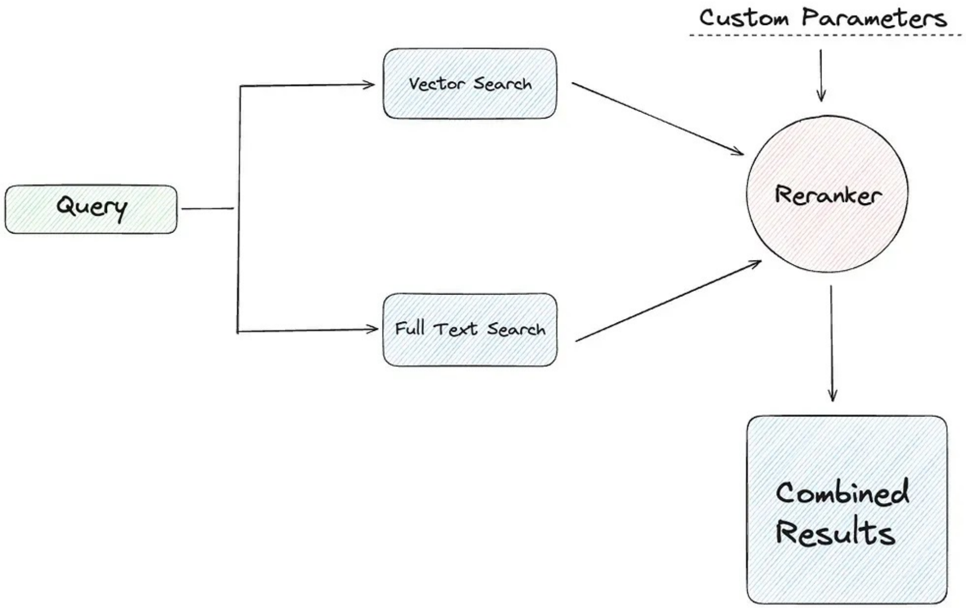
RAG for Text-to-SQL

getindata
Part of Xebia



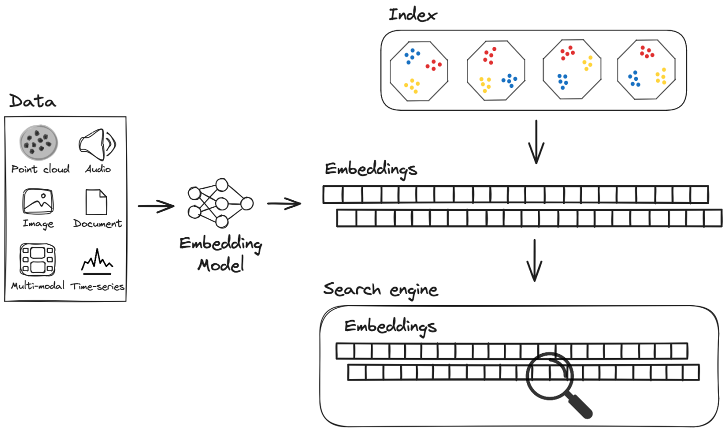
Hybrid search

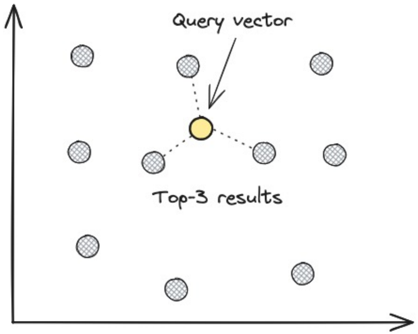
- combination of keyword and vector search

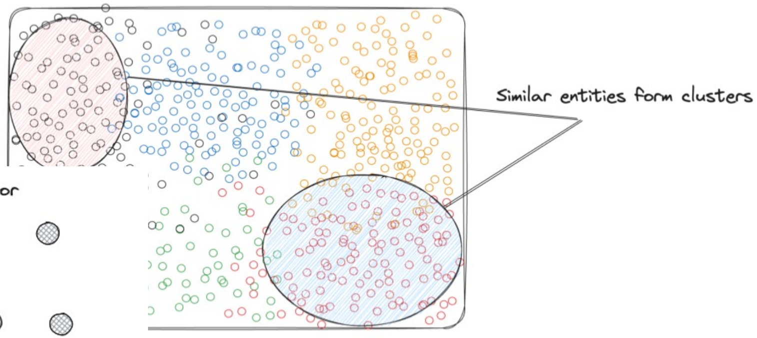


Vector search

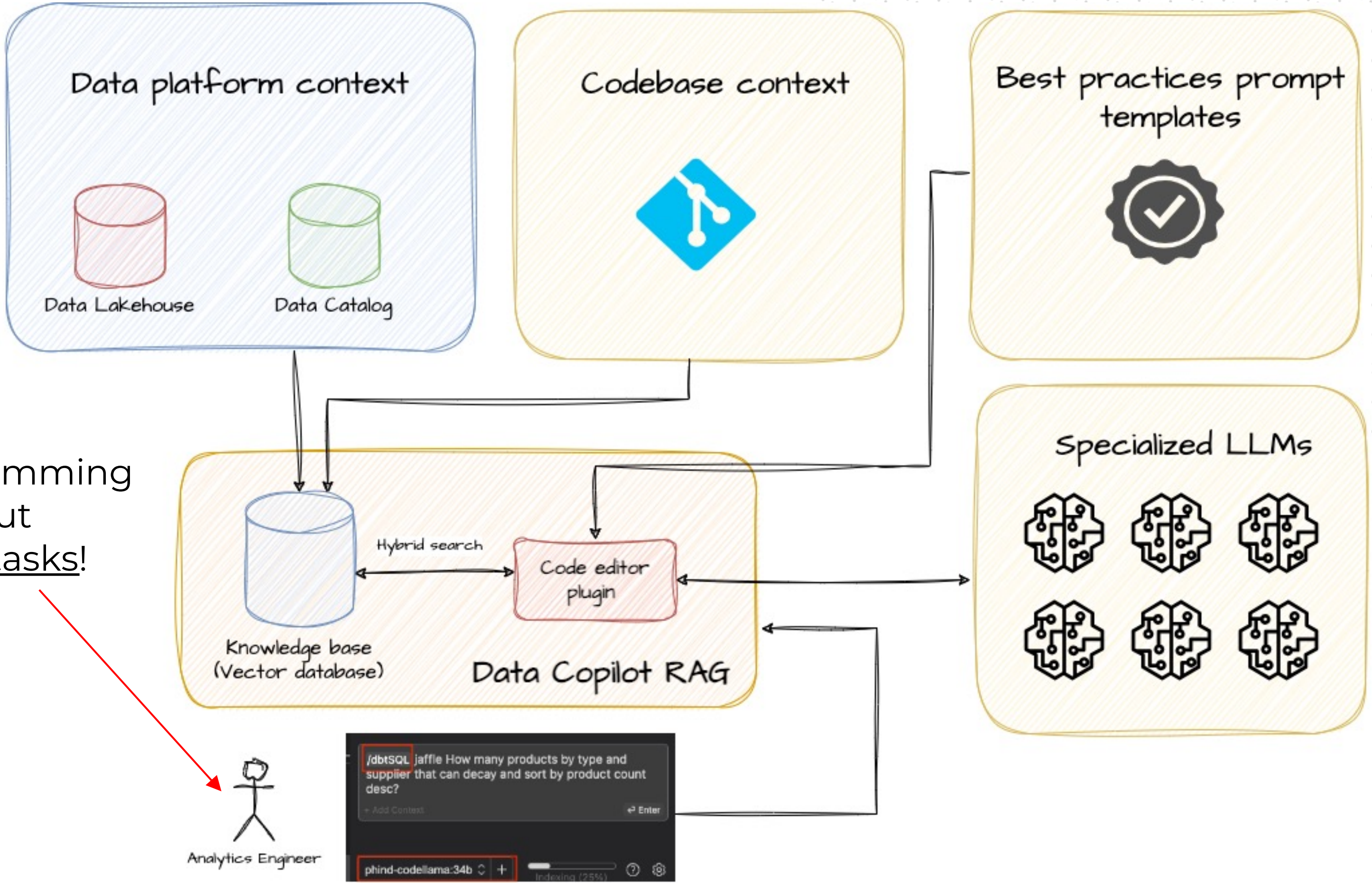
- a technique used to search for similar items based on their vector representations, called embeddings
- Approximate Nearest Neighbours algorithms







Data Copilot RAG architecture

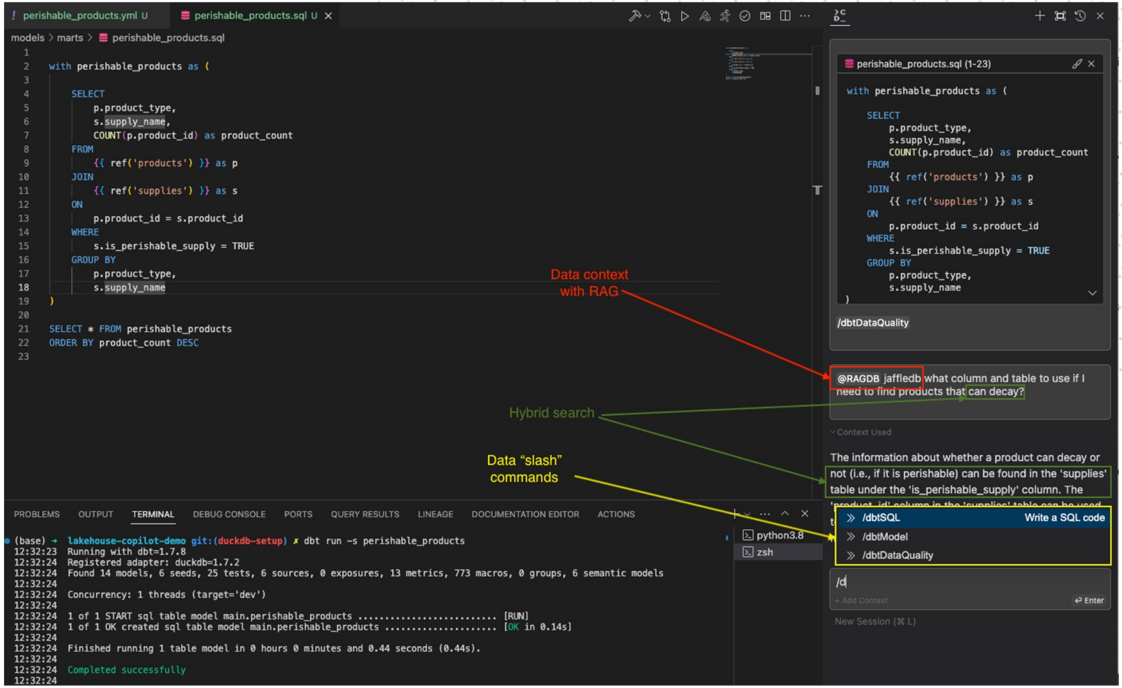


Data programming is more about <u>repeatable tasks</u>!

GID Data Copilot (GDC)

getindata
Part of Xebia

- An extensible AI **programming assistant** for **SQL** and **dbt** code
- Powered by:
  - **Large Language Models** (SOTA LLMs)
  - Robust **Retrieval Augmented Generation** (RAG) architecture
  - **Hybrid search** techniques
  - Fast **Vector Database**
  - Curated **Prompts**
  - Builtin **Data commands**



Continue - an open-source autopilot

getindata
Part of Xebia

- support for both tasks and tab autocompletion
- highly extensible
  - use any LLM model you wish - also ***multiple, specialized models*** for different languages or tasks
  - support for many ***model providers***, such as Ollama, vLLM, LM Studio
  - custom ***context providers*** for more control over LLMs augmentation
  - custom ***slash commands*** that can combine own prompts, contexts and models for specialized, reusable tasks
- support for VSCode and Jetbrains
- secure (i.e. can be run locally, on-premise or cloud VPC)
- translate your best practices into ”slash data commands”

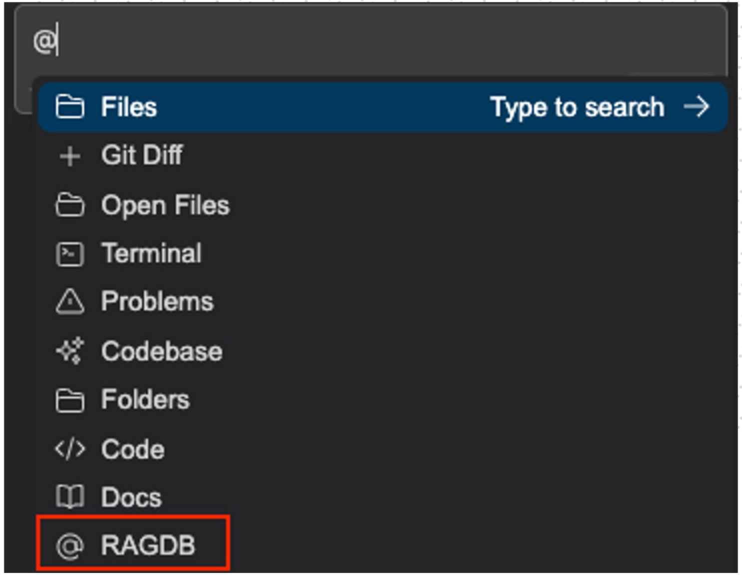
Continue - a custom context provider

```typescript
const RagContextProvider: CustomContextProvider = {
  title: "ragdb",
  displayTitle: "RAGDB",
  description:
    "Retrieve DB schema from our vector database of internal documents",
  type: "normal",

  getContextItems: async (
    query: string,
    extras: ContextProviderExtras
  ): Promise<ContextItem[]> => {
    console.info(extras.fullInput)
    const inputArray = extras.fullInput.split(' ');
    const db = inputArray[0];
    const userQuestion = inputArray.slice(2).join(' ');
    const response = await fetch("http://localhost:8000/retrieve", {
      method: "POST",
      headers: {
        'content-type': 'application/json;charset=UTF-8',
      },
      body: JSON.stringify({ query: userQuestion }),
    });

    const results = await response.json();

    return results.map((result: { title: any; contents: any; }) => ({
      name: result.title,
      description: result.title,
      content: result.contents,
    }));
  },
};
```

generate context

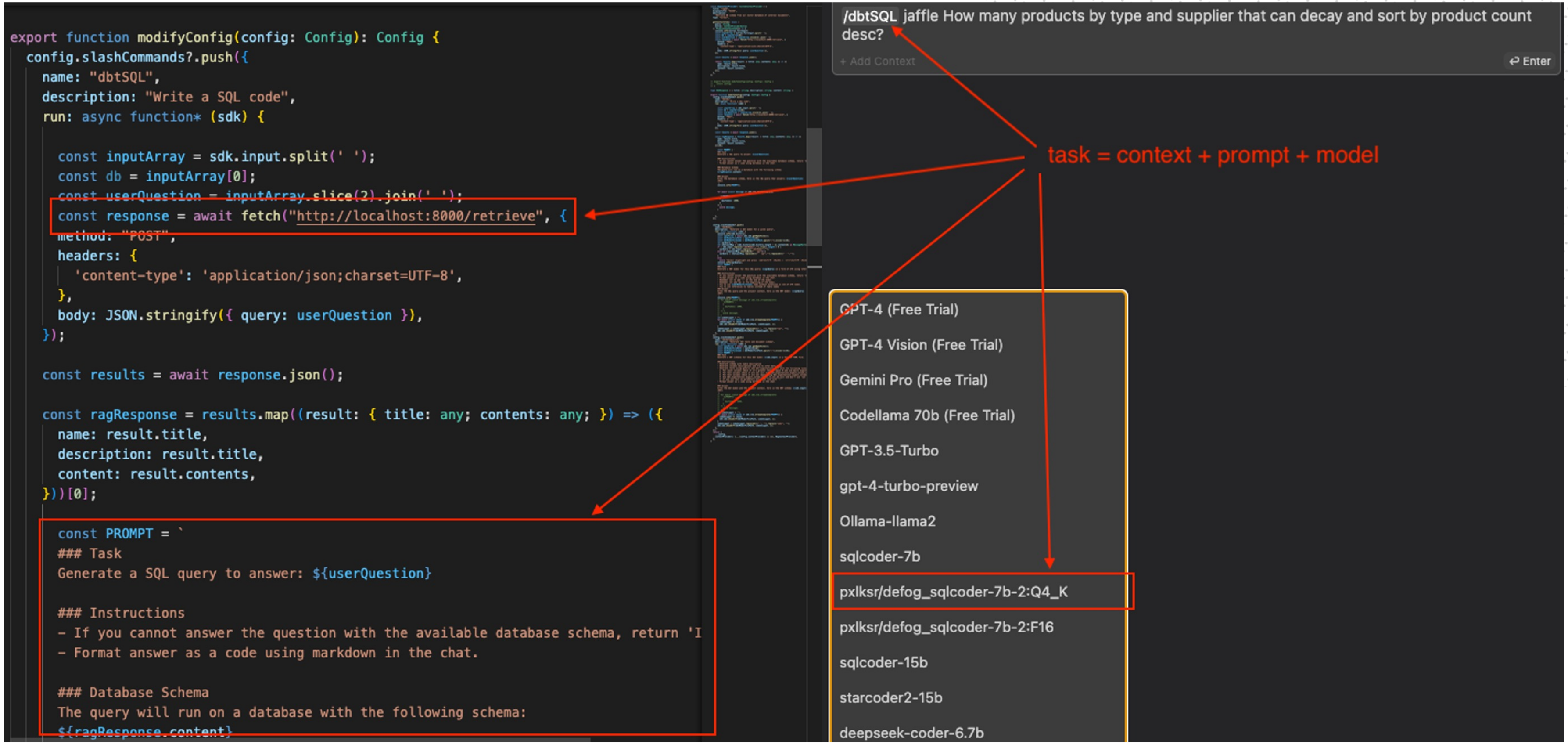
call retriever

user question



dbtSQL task = custom(context + prompt + model)

getindata
Part of Xebia



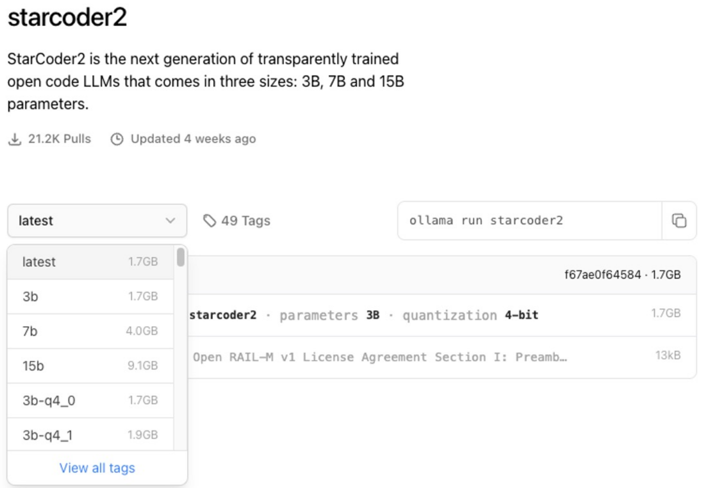
Ollama

- fast and easy self-hosting of LLMs almost *everywhere*
- hybrid CPU+GPU inference
- powered by llama.cpp
- rich library of existing LLMs in different flavours
- GGUF - fast and memory efficient quantization for GPU
- Serve model with one-liner:

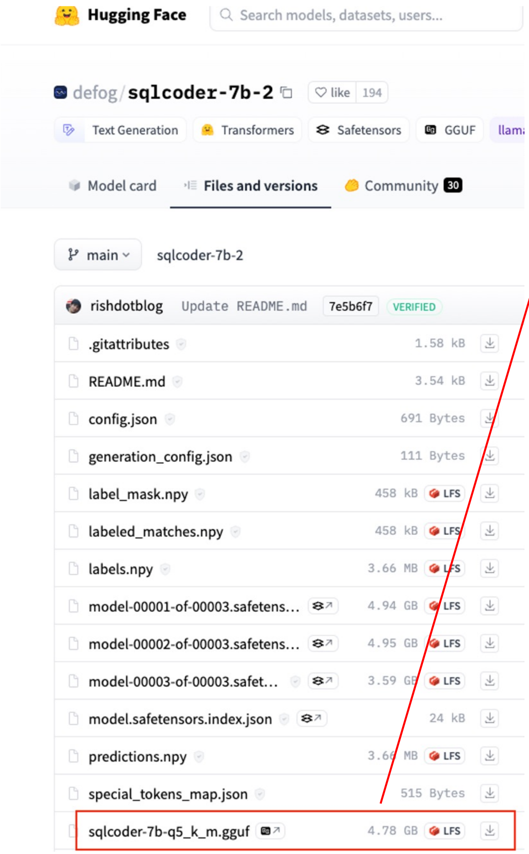
```
ollama run starcoder2:7b
```

- vLLM for production deployments

(Our video tutorial)



Ollama - custom model in 4 steps



1. Download a model in the GGUF format
2. Create a Modelfile, e.g.:

```
FROM ./sqlcoder-7b-q5_k_m.gguf
TEMPLATE """{{ .Prompt }}"""
PARAMETER stop "<|endoftext|>"
```

3. Create a model with Ollama

```
ollama create sqlcoder-7b-2 -f Modefile
```

4. Serve it

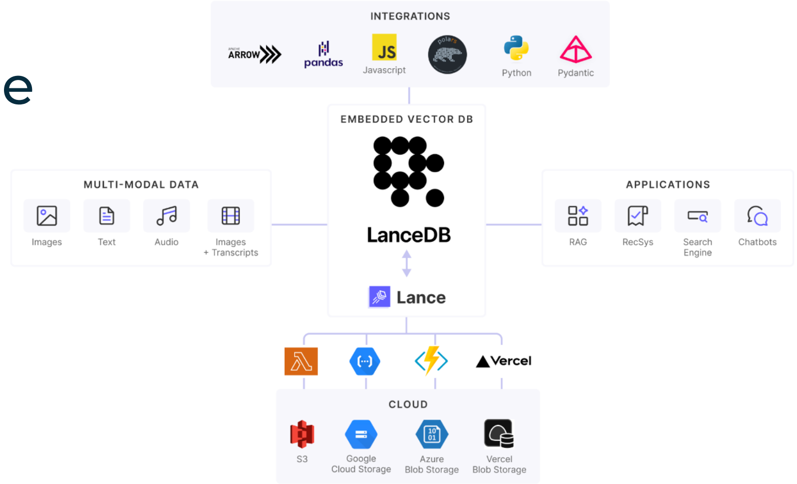
```
ollama run sqlcoder-7b-2
```

LanceDB

getindata
Part of Xebia

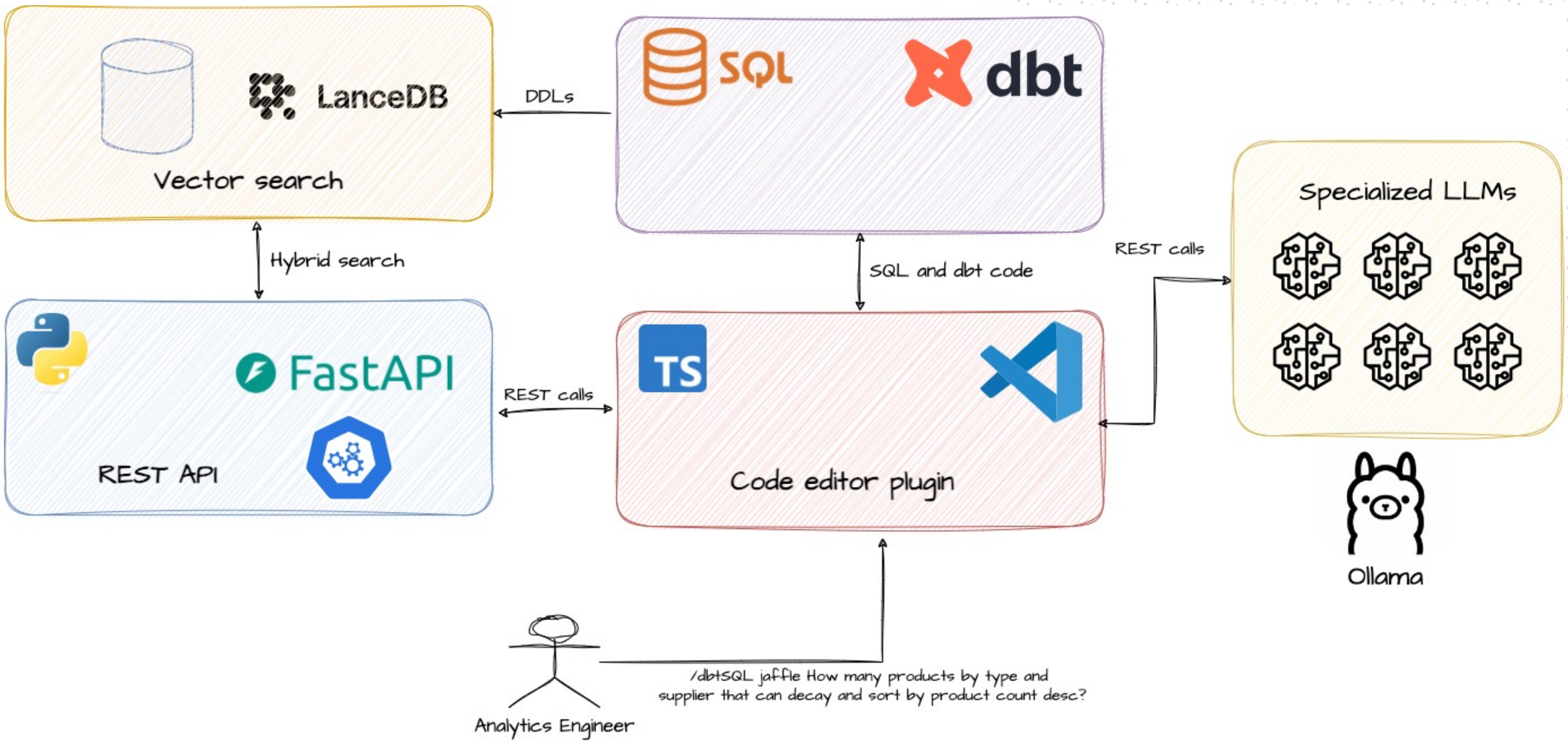
- fast (Rust❤️), serverless and embeddable - DuckDB for ML
- powered by Lance file format for ML (versioning, zero-copy)
- multi-modal
- support for hybrid (semantic + keyword) search
- Llamaindex integration
- Python API and fast data exchange

  with polars and Arrow



Technical architecture

getindata
Part of Xebia



Question representation[1]

```
Table continents, columns = [ContId, Continent]
Table countries, columns = [CountryId, CountryName,
↳ Continent]
Q: How many continents are there?
A: SELECT
```

Listing 1: Example of Basic Prompt

```
Given the following database schema:
continents: ContId, Continent
countries: CountryId, CountryName, Continent

Answer the following: How many continents are there?
SELECT
```

Listing 2: Example of Text Representation Prompt

```
### Complete sqlite SQL query only and with no
↳ explanation
### SQLite SQL tables, with their properties:
#
# continents(ContId, Continent)
# countries(CountryId, CountryName, Continent)
#
### How many continents are there?
SELECT
```

Listing 3: Example of OpenAI Demostration Prompt

```
/* Given the following database schema: */
CREATE TABLE continents(
    ContId int primary key,
    Continent text,
    foreign key(ContId) references countries(Continent)
);

CREATE TABLE countries(
    CountryId int primary key,
    CountryName text,
    Continent int,
    foreign key(Continent) references continents(ContId)
);

/* Answer the following: How many continents are there?
↳ */
SELECT
```

Listing 4: Example of Code Representation Prompt

[1]Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation

LLMs evaluation 1/2

- Not meant to be yet another benchmark, such as: Spider or Bird-SQL
- Jaffle Shop example - simple but not trivial
- 4 typical data tasks
  - Data model exploration/discovery
  - SQL: an easy one (single table) and more complex (joins with sorting and aggregations)
  - dbt model generation
  - dbt tests generation based on rules

LLMs evaluation 2/2

Model	Licence	size [b]	Data discovery	SQL - simple	SQL - complex	dbt - model	dbt - tests
deepseek-coder	deepseek	33	+	+	+/-	+/-	-
deepseek-coder	deepseek	6.7	+/-	+/-	+/-	-/+	-
codellama	Llama2	70	+	+	-/+	-	-
starcoder2	bigcode-openrail-m	15	-	-	-	-	-
sqlcoder	CC BY-SA 4.0	7	-	+/-	+/-	N/A	N/A
phind-codellama	Llama2	34	+	+	+	+	+/-
wizardcoder	Llama2	33	+	+	+/-	+/-	-/+
gpt-3.5-turbo	Commercial	N/A	+	+	+	+	+/-
gpt-4	Commercial	N/A	+	+	+	+	+
gpt-4-turbo-preview	Commercial	N/A	+	+	+	+	+/-
Gemini Pro	Commercial	N/A	+	+/-	+	+/-	-
OpenCodeInterpreter	Apache-2.0	33	+	+	-/+	-	-

+- perfect or almost perfect

+/- - not optimal or some minor tweaks needed

-/+ - not very helpful, serious hallucinations

- - totally useless

A handful of conclusions...with a grain of salt

getindata
Part of Xebia

- NL-to-SQL and dbt code generation are ***challenging***
- ***commercial*** models are in most cases still better... but
- there are very promising ***open-source ~30b alternatives***
- ***SQL-dedicated*** and fine-tuned models can turn out a bit a disappointing (beam search?), e.g. :
    - unnecessary joins elimination
    - wrong data types inference
    - occasional hallucinations

Future directions

- Implementation of in-context learning such as Query Similarity Selection (few-shot strategy)
- Model(s) fine-tuning using using dbt examples, especially for data quality aspects
- Fine-tuning focused on platform-to-platform migrations

Welcome to the GID data copilot

DEMO

AI & LLMOps free consultation

getindata
Part of Xebia

Thank you !