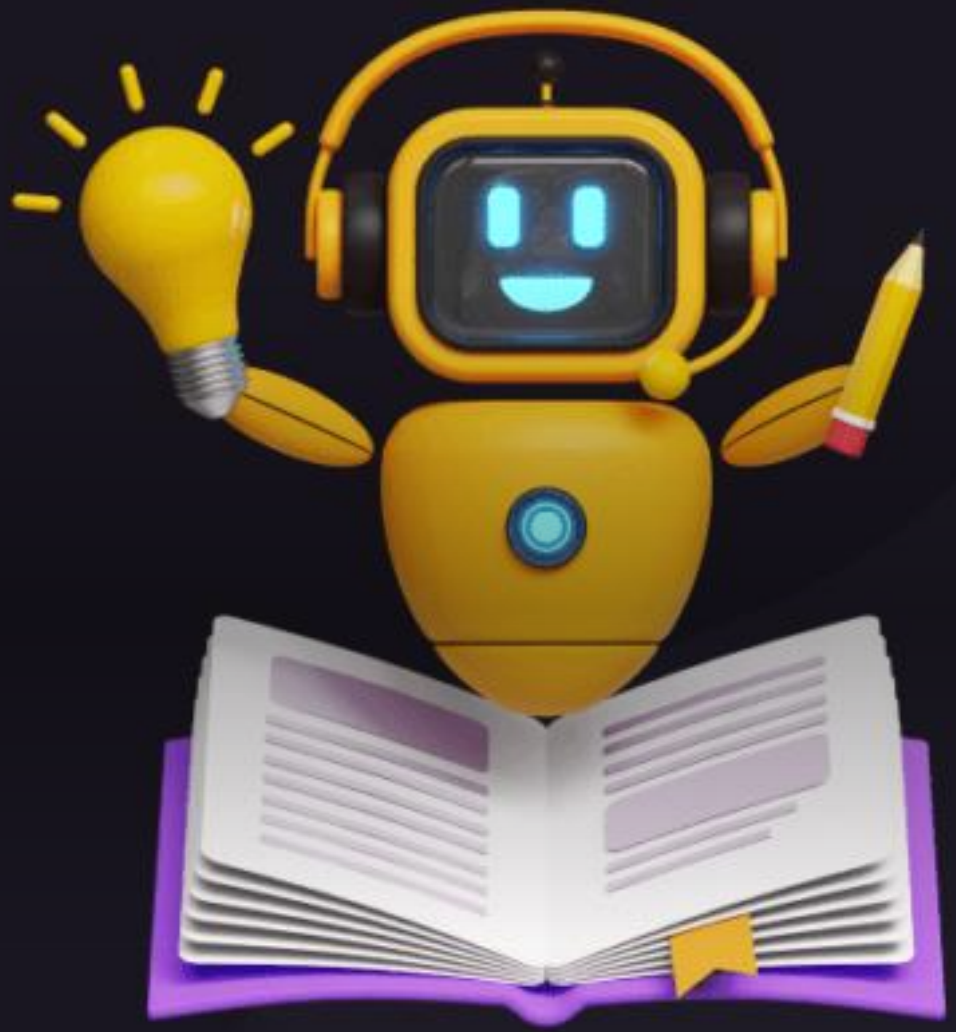


Retrieval Augmented Generation

05.06.2025

Jonas Wolber

What is Retrieval Augmented Generation (RAG)?

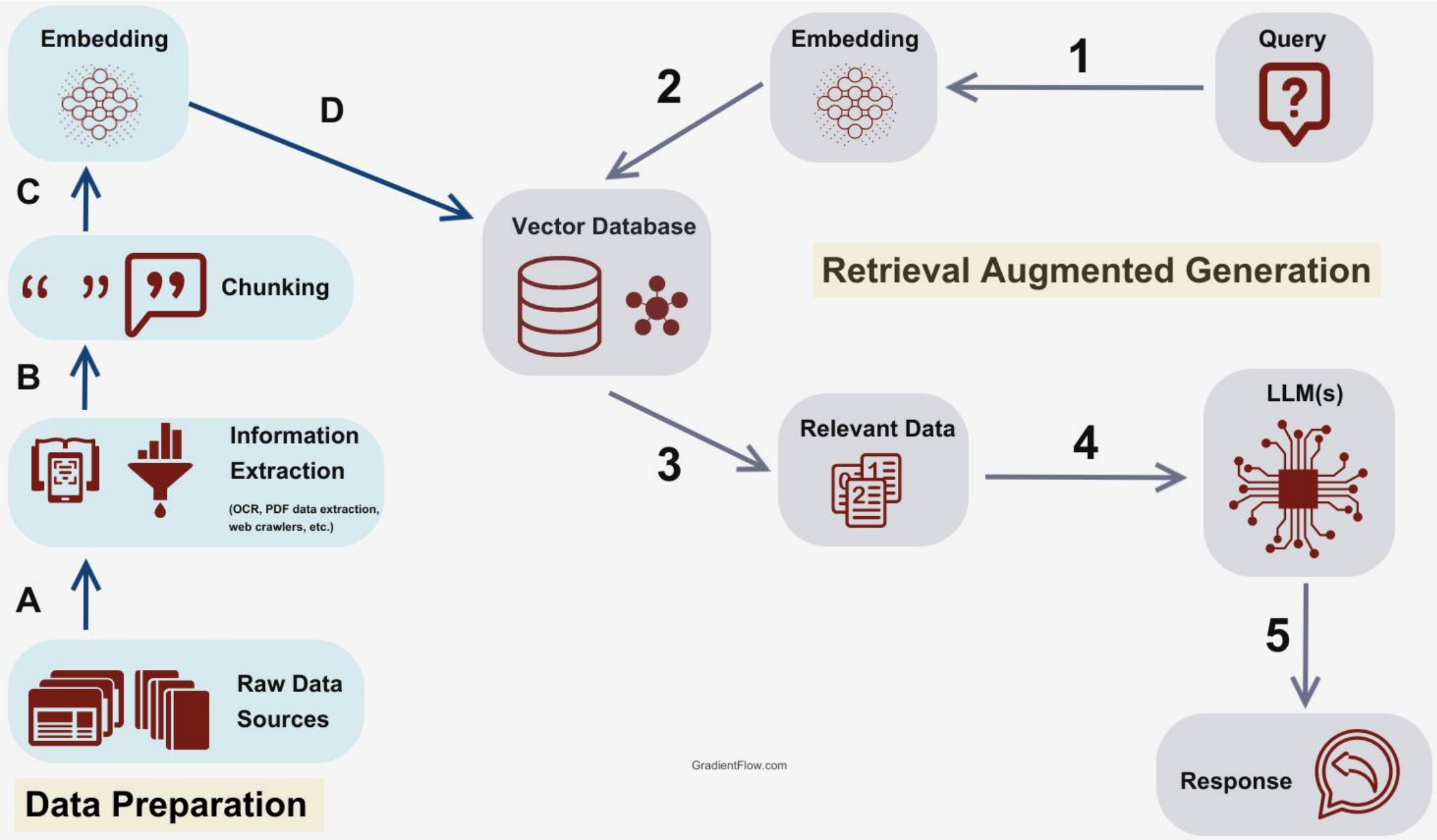


Image taken from: fusionchat.ai/news/10-ways-rag-enhances-ai-systems

Word embeddings and vector space

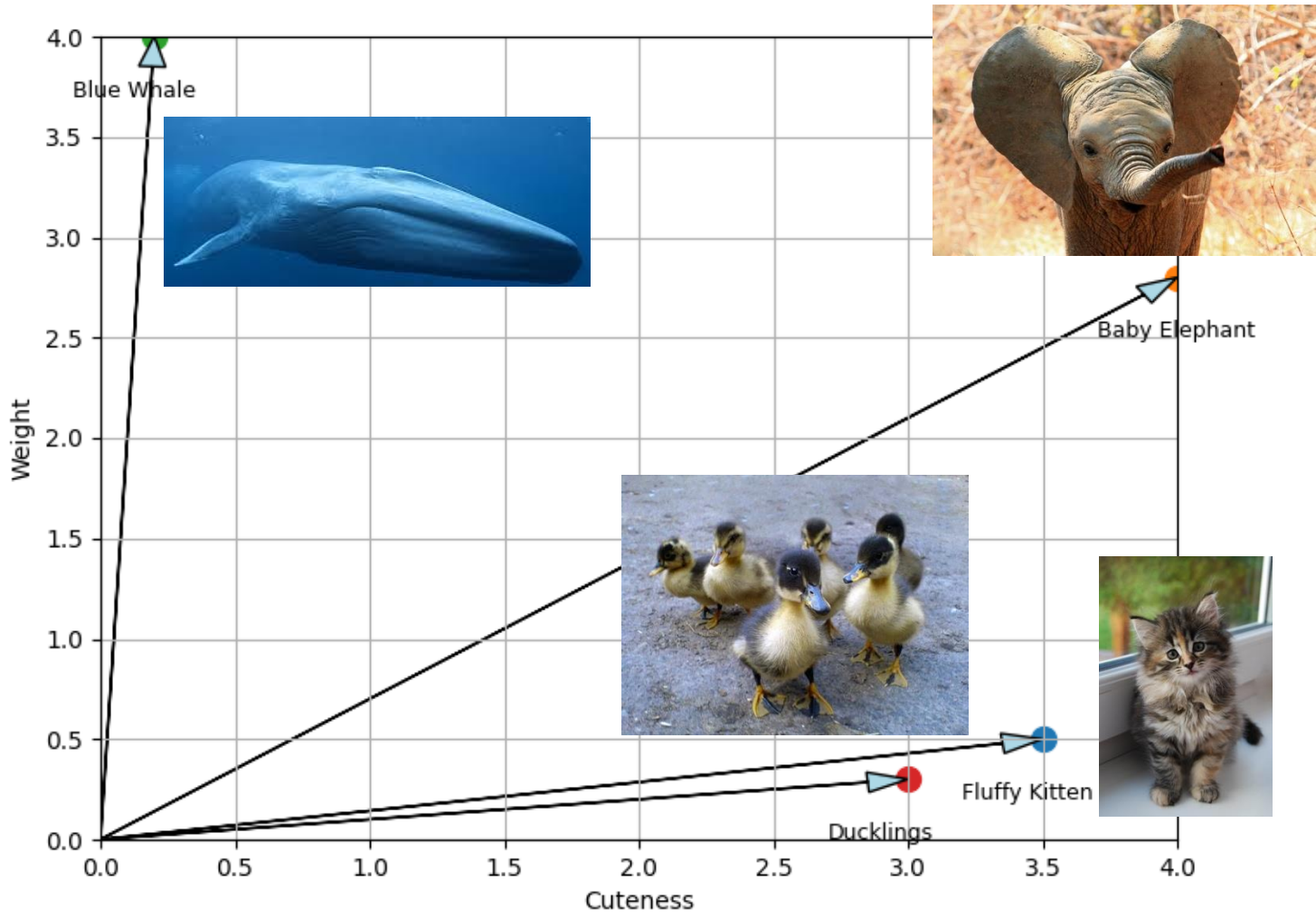
Text
"Center for Computational Life Sciences"

Tokenization

Tokens
“Cen“, “ter“, “for“, “Compu“, “tational“, “Life“, “Scien“, “ces“

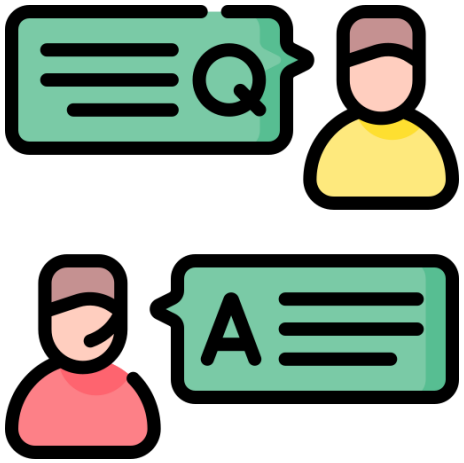
Embedding

Embeddings
[0.016539961, -0.025933826, 0.048515484, 0.03760145, -0.04732014, -0.002327353, ...]



Applications of RAG

Question Answering



Document Summarization

Coding Assistants

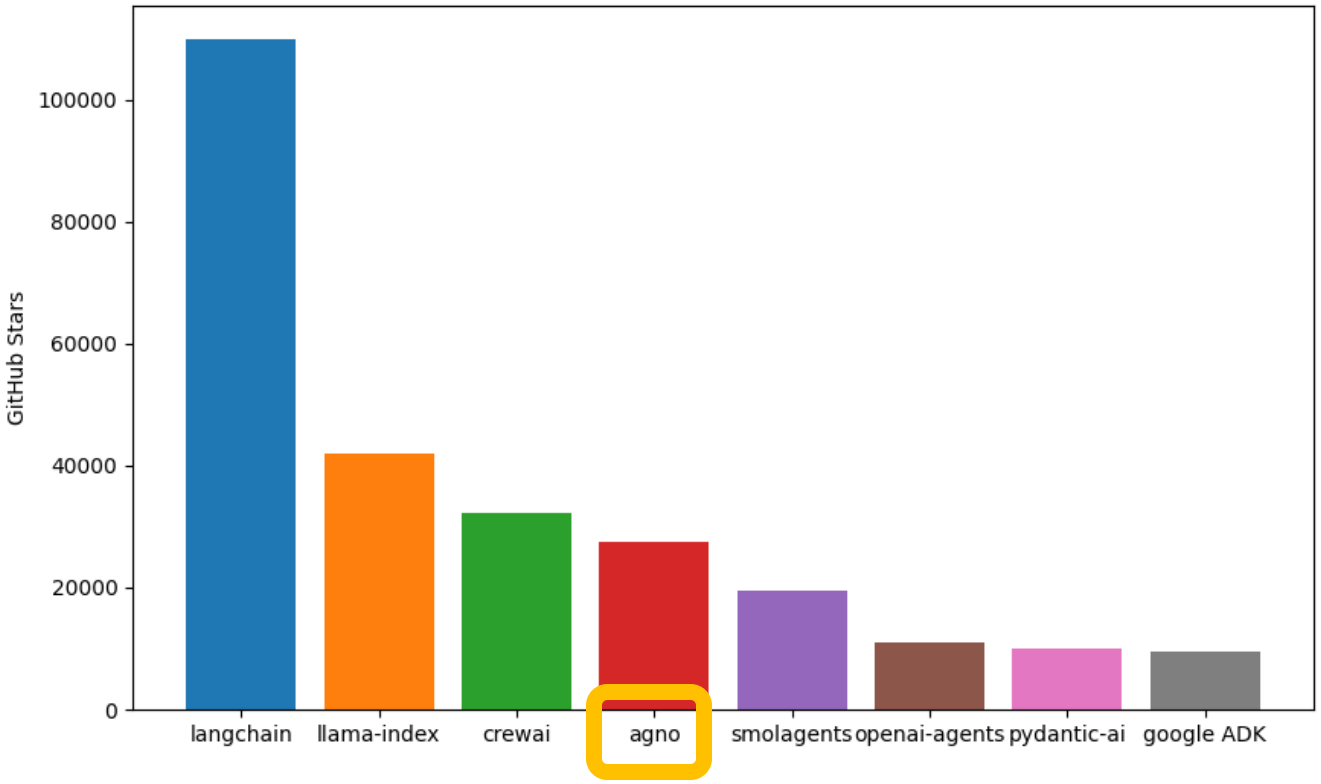
Enterprise Knowledge Management

Patient Diagnostics

Scientific Research

Icons made by Freepik, Juicyfish, Dewi Sari, Vectorsmarket from www.flaticon.com

Frameworks for LLMs, RAG and Agentic AI



agno

- Beginner-friendly
- Lightweight
- Fast
- Open-source
- Flexible

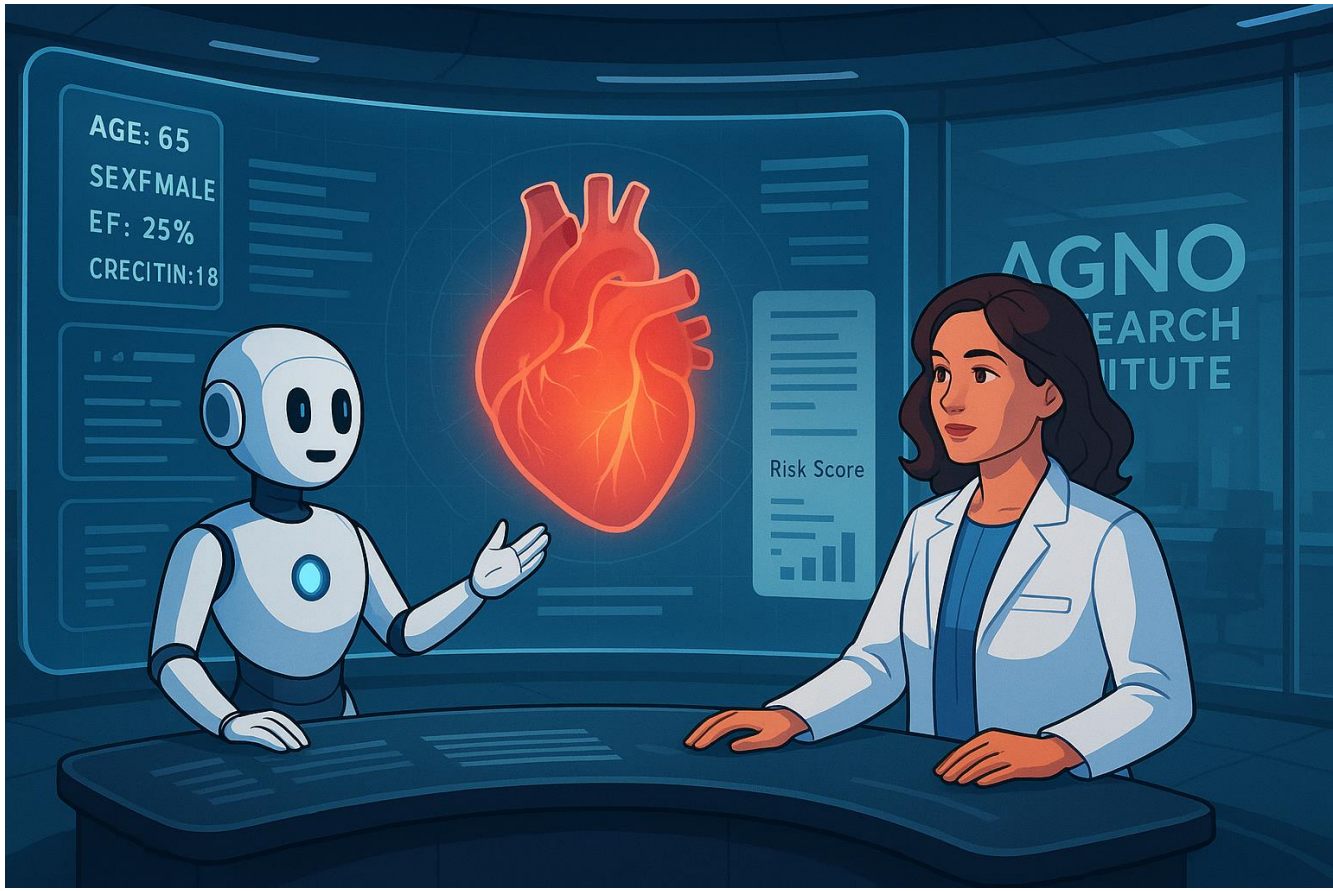
Building our first RAG app

Task

Dr. Cardia, a brilliant but overwhelmed cardiologist has a patient with troubling symptoms: fatigue, chest pain, shortness of breath. His vitals and lab results are confusing. The risk of **heart failure** looms, but Dr. Cardia isn't sure which features are most predictive for this patient's condition—or what the latest research says about key predictors and outcomes.

She found a very interesting **scientific article** about the risk factors of heart failure but she does not have time to read the whole article.

She asks up to come up with a **RAG agent** that can quickly and accurately look up information from this article and provide meaningful responses.



Setting up the Python environment

1. Setting up the environment

```
(base) jwolber@Mac datathon % python3 -m venv ccls_datathon
source ccls_datathon/bin/activate
```

2. Install dependencies

```
(ccls_datathon) (base) jwolber@Mac datathon % pip install -r requirements.txt
Collecting agno
  Using cached agno-1.5.6-py3-none-any.whl (802 kB)
```

3. Set up Azure OpenAI

```
from agno.models.azure import AzureOpenAI
from agno.agent import Agent
import os

model_name = "gpt-4.1-nano"
api_version="2025-04-01-preview"
endpoint =
api_key =

os.environ["AZURE_OPENAI_API_KEY"] = api_key
os.environ["AZURE_OPENAI_ENDPOINT"] = endpoint
os.environ["OPENAI_API_VERSION"] = api_version
```

4. Test

```
agent = Agent(
    model=AzureOpenAI(id=model_name),
    description="You are an enthusiastic news reporter with a flair for storytelling!",
)
res = agent.run("Tell me about a breaking news story from Aachen.")
print(res.content)
```