

Large Language Models for Devs

***from zero to your first LLM
application***



Open Source Day 2024

Stefano Fiorucci



Stefano Fiorucci

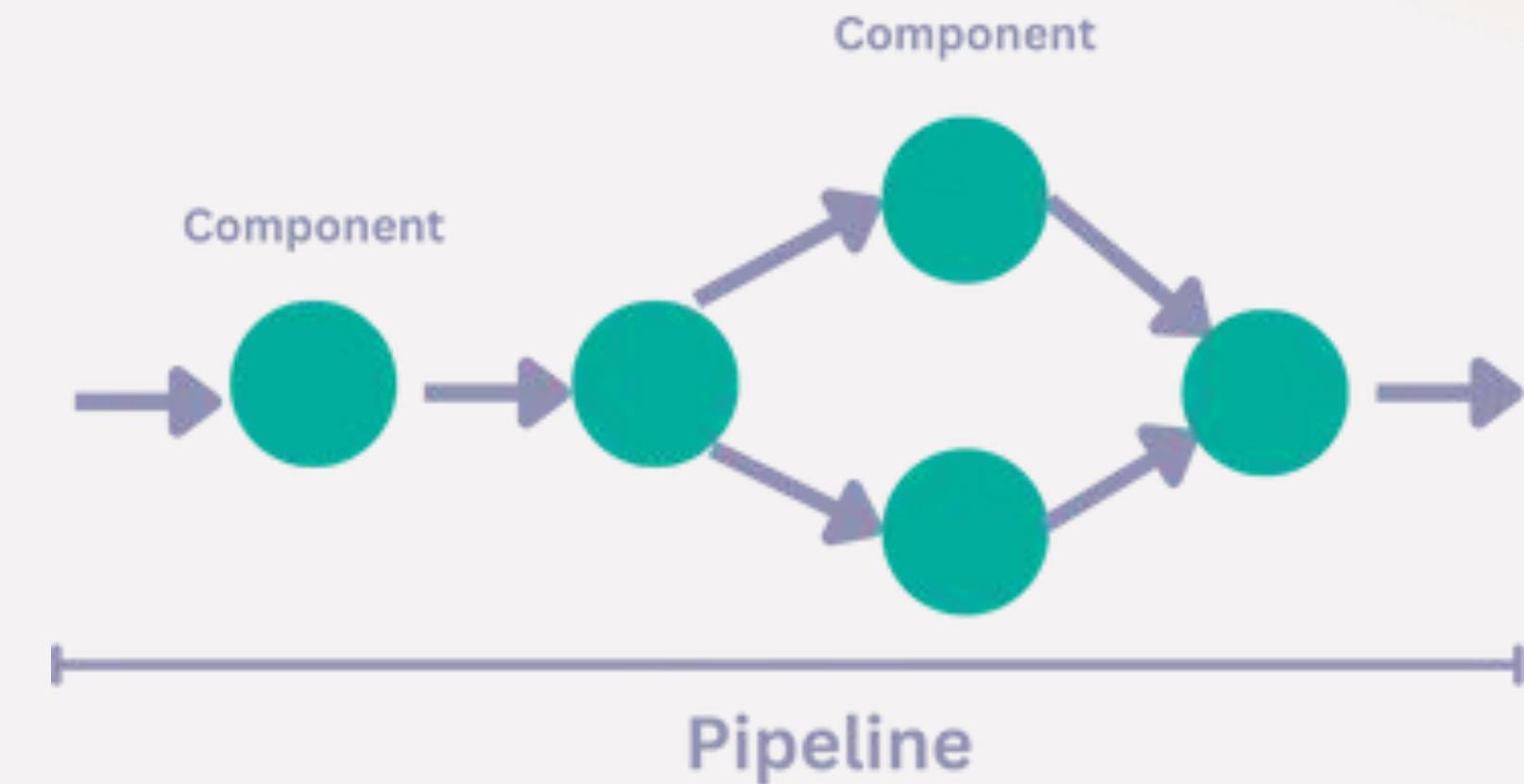
Software/NLP Engineer
deepset

- NLP Engineer, Craftsman and Explorer 
- LinkedIn: [Stefano Fiorucci](#)
- GitHub: [@anakin87](#)
-  Open source LLM Framework: Haystack



Starred 12.8k ▾

- Open-source orchestration framework to build LLM applications
- Use cases: RAG, question answering, semantic search, chatbots...
- Building blocks: Components & Pipelines

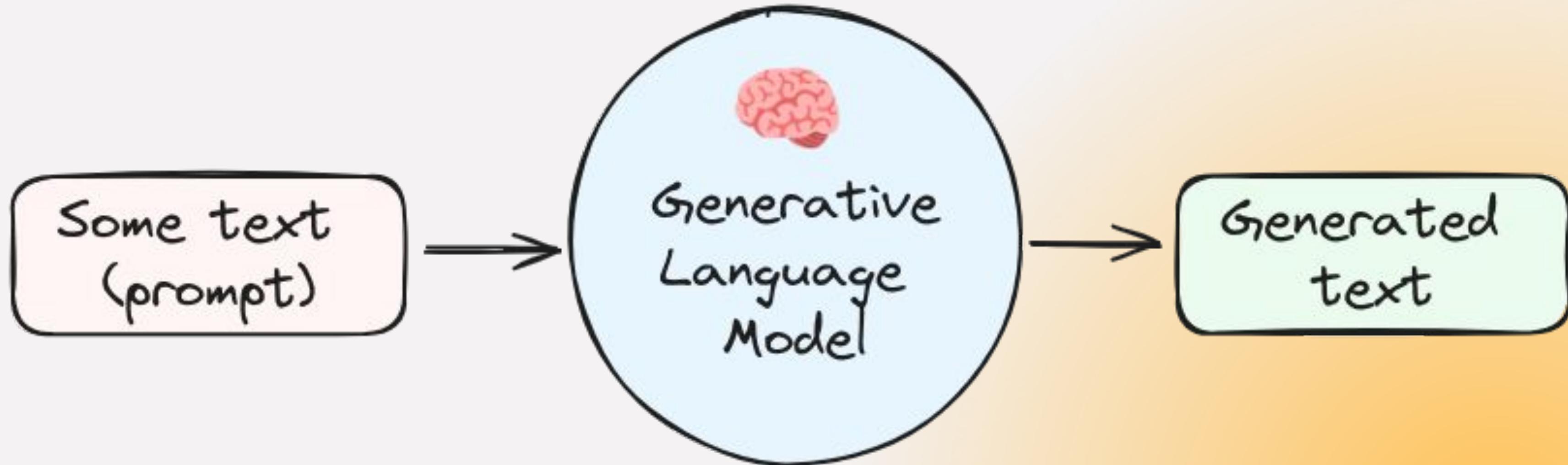


Why/Agenda

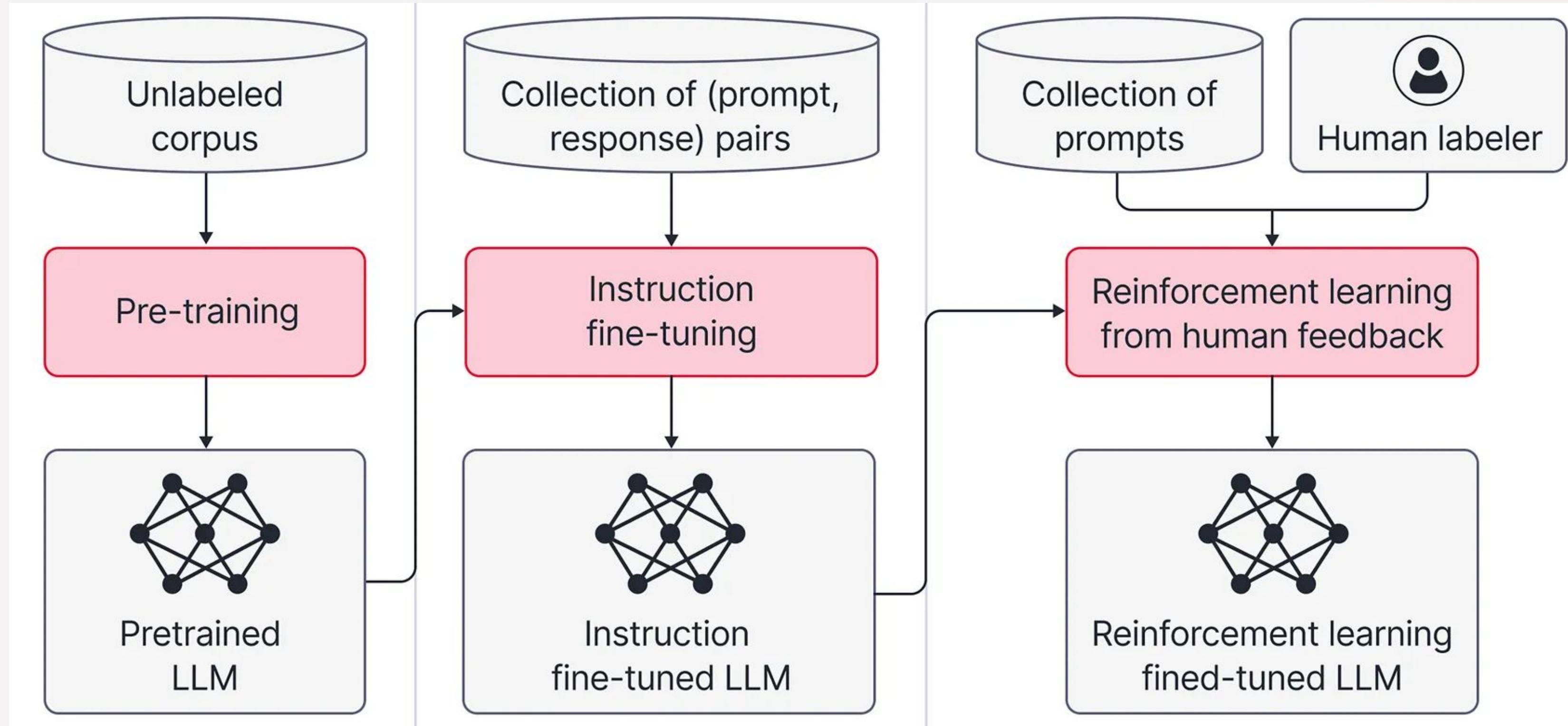
- ◆ Intro
- ◆ Proprietary and open models
- ◆ Chat
- ◆ Prompt engineering
- ◆ Common challenges with LLMs
- ◆ RAG
- ◆ Retrieval
- ◆ Deploy a LLM application
- ◆ Beyond RAG...



What is a Large Language Model?



How are LLMs trained?



Let's start!

```
! pip install haystack-ai

from haystack.components.generators import OpenAIGenerator

generator = OpenAIGenerator()
print(generator.run(prompt="Where is Florence?"))

# {'replies': ['Florence is a city located
# in the Tuscany region of Italy.'], ...}
```



Switch to local open LLMs

```
curl -fsSL https://ollama.com/install.sh | sh
```

```
ollama serve
```

```
ollama pull mistral
```



```
! pip install ollama-haystack
```

```
from haystack_integrations.components.generators.ollama import OllamaGenerator
```

```
generator = OllamaGenerator(model="mistral")
```

```
print(generator.run(prompt="Where is Florence?"))
```

```
# {'replies': ['Florence is a city in the region of Tuscany...'], ...}
```

Install Ollama and pull an open model

Chat with Mistral



```
from haystack_integrations.components.generators.ollama \
    import OllamaChatGenerator
from haystack.dataclasses import ChatMessage

generator = OllamaChatGenerator(model="mistral")

messages = []

while True:
    msg = input("Enter your message or Q to exit\n👤 ")
    if msg=="Q":
        break
    messages.append(ChatMessage.from_user(msg))
    response = generator.run(messages=messages)
    assistant_resp = response['replies'][0]
    print("🤖 "+assistant_resp.content)
    messages.append(assistant_resp)
```



Chat with Mistral



👤 can you help me planning a trip in Florence?

🤖 Absolutely! Florence, Italy is known for its rich history, stunning art, and beautiful architecture. Here's a suggested itinerary for a 3-day trip to Florence:

Day 1:

1. Begin your day at the Accademia Gallery to see Michelangelo's famous statue of David....

👤 nice. can you provide me with a list of things I should take with me to Italy for the trip?

🤖 Certainly! Here's a list of essential items to pack for your trip to Italy, keeping in mind that Florence has a Mediterranean climate with warm summers and cool winters:

1. Clothing: Pack lightweight, breathable clothing for the summer months, and bring layers for cooler evenings or winter travel. Consider bringing a rain jacket or umbrella as well....



Prompt engineering

From [OpenAI guide](#):

- Write clear instructions
- Provide reference text
- Split complex tasks into simpler subtasks
- Give the model time to "think"
- Use external tools
- Test changes systematically



promptingguide.ai



Common challenges with LLMs

👤 Who won the 2024 Italian Super Cup?

🤖 **GPT3.5-TURBO** Unfortunately, I do not have information on the winner of the 2024 Italian Super Cup as it is a fictional event that has not occurred yet.

🤖 **MISTRAL-7B** I cannot provide an answer to that question as I don't have access to real-time information and the Italian Super Cup for the year 2024 has not yet taken place...



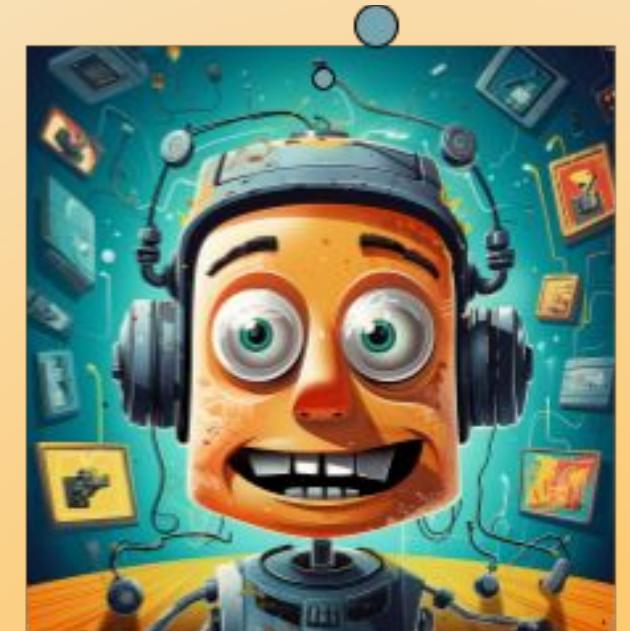
Their knowledge is not up-to-date

Common challenges with LLMs

👤 Who is Stefano Fiorucci?

🤖 **GPT3.5-TURBO** Stefano Fiorucci is an Italian fashion designer and the founder of the luxury fashion brand Fiorucci....

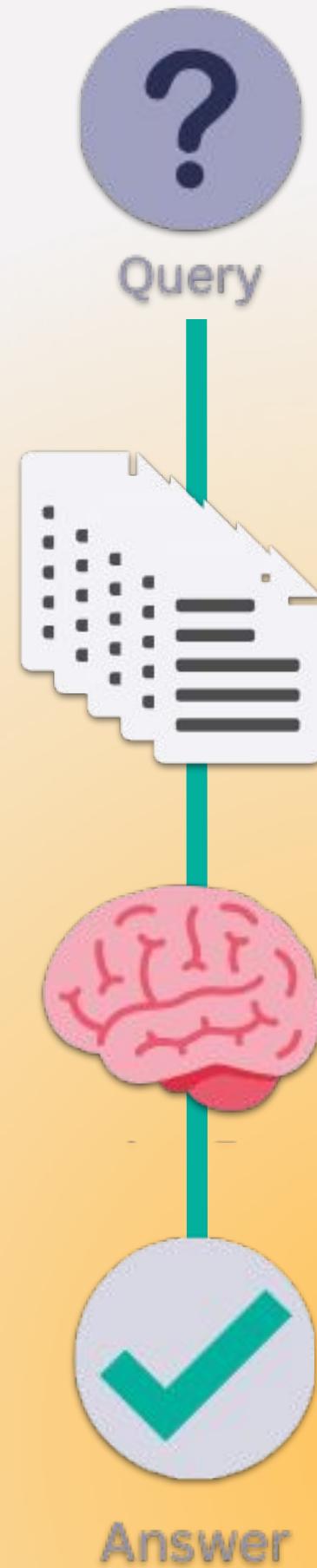
🤖 **MISTRAL-7B** Stefano Fiorucci is an Italian artist, born on February 23, 1980. He is primarily known for his digital art and graphic design works...



Hallucinations

Meet RAG!

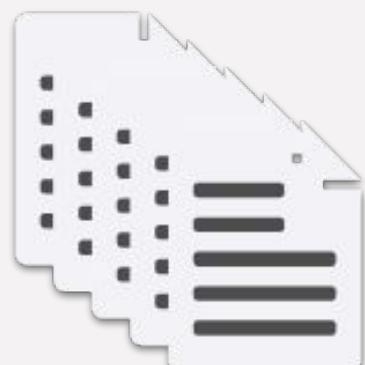
- *Retrieval Augmented Generation*
 - ◆ improving text generation by injecting knowledge into the prompt
- Direct prompt as a “closed book exam”
 - ◆ either the LLM knows, or it makes up the answer it considers most likely
- RAG as an “open book exam”
 - ◆ the LLMs receives all the information it needs to answer the question and it only has to re-elaborate it



Naive RAG 1/3 - Create the Knowledge Base

```
from haystack import Document

documents = [
    Document(content="Italian Super Cup is an annual super cup tournament in Italian football."),
    Document(content="The tournament was established in 1988 and the first title was won by AC Milan."),
    Document(content="In 2024, the tournament was won by Inter.")
]
```



Naive RAG 2/3 - Prompt Builder

```
from haystack.components.builders.prompt_builder import PromptBuilder

prompt_template = """
Given these documents, answer the question.

Documents:
{% for doc in documents %}
    {{ doc.content }}
{% endfor %}
Question: {{question}}
Answer:
"""

prompt_builder = PromptBuilder(template=prompt_template)
```



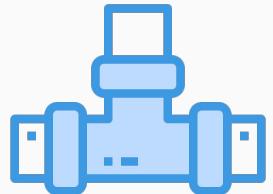
Naive RAG 3/3 - Pipeline

```
from haystack import Pipeline

rag_pipeline = Pipeline()
rag_pipeline.add_component("prompt_builder", prompt_builder)
rag_pipeline.add_component("llm", OllamaGenerator(model="mistral"))
rag_pipeline.connect("prompt_builder", "llm")

# Ask a question
question = "Who won the Italian Super Cup in 2024?"
res = rag_pipeline.run({"question": question,
                       "documents": documents})

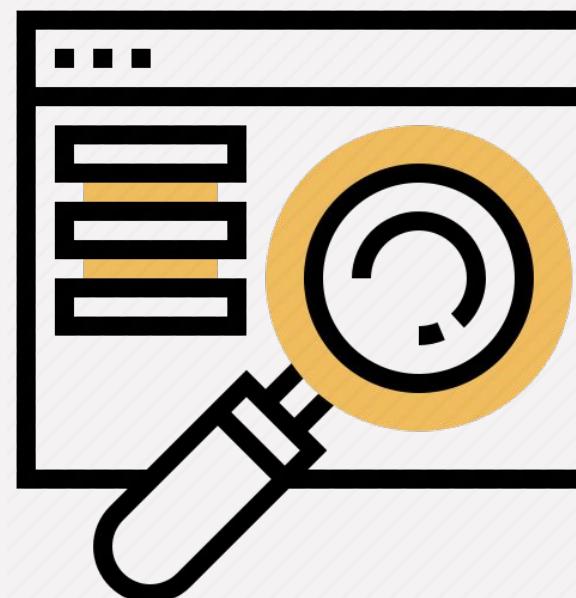
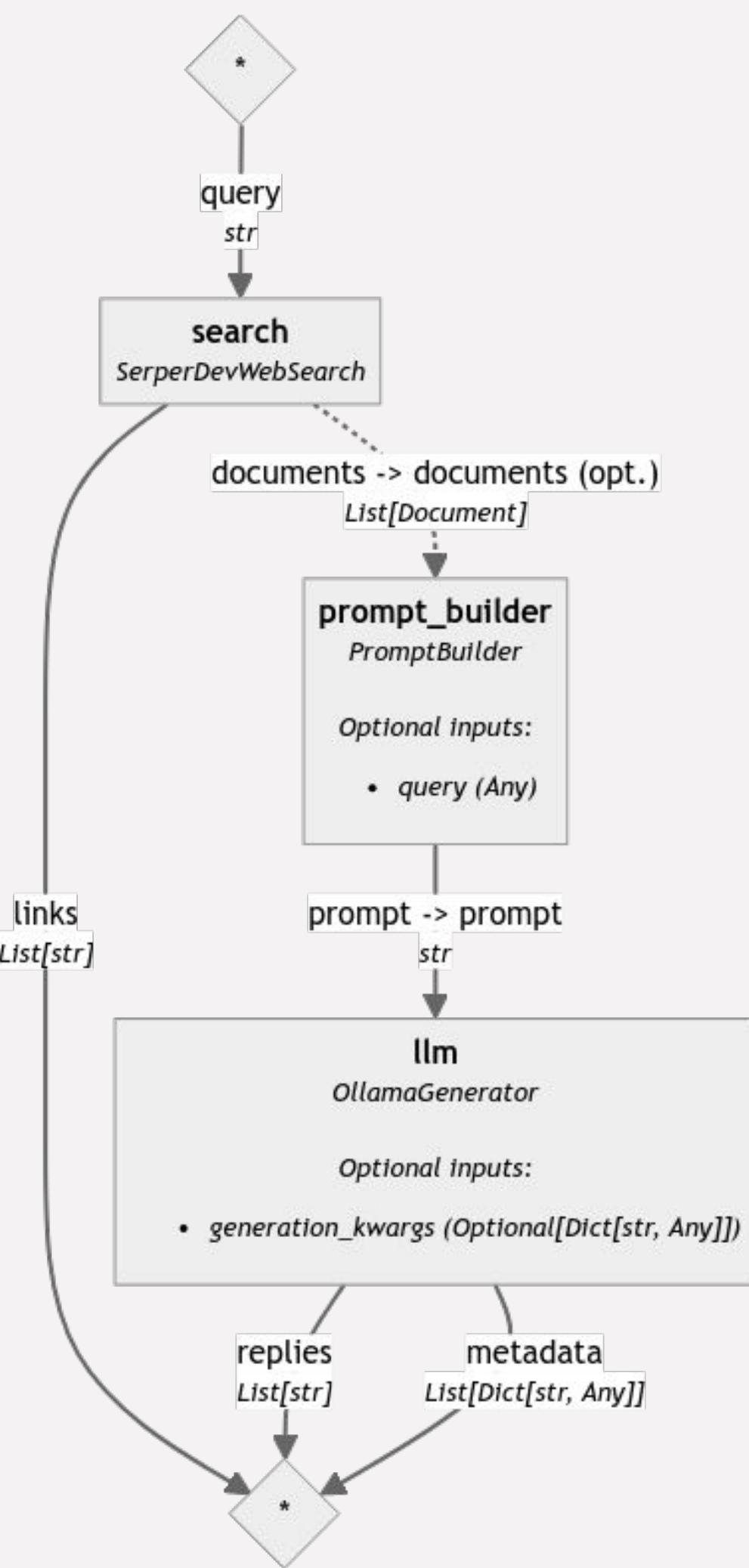
print(res["llm"]["replies"])
```



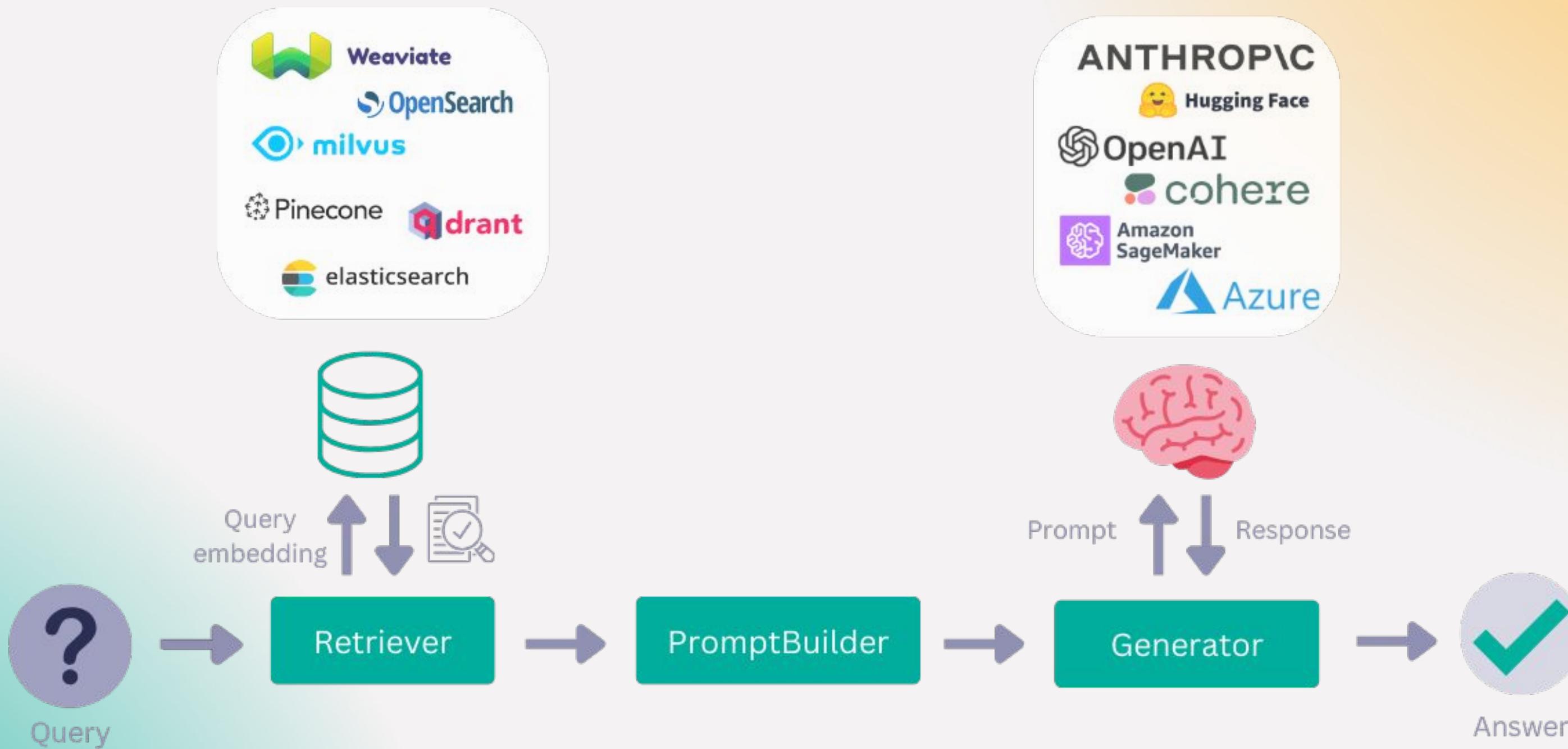
['Inter won the Italian Super Cup in 2024.]



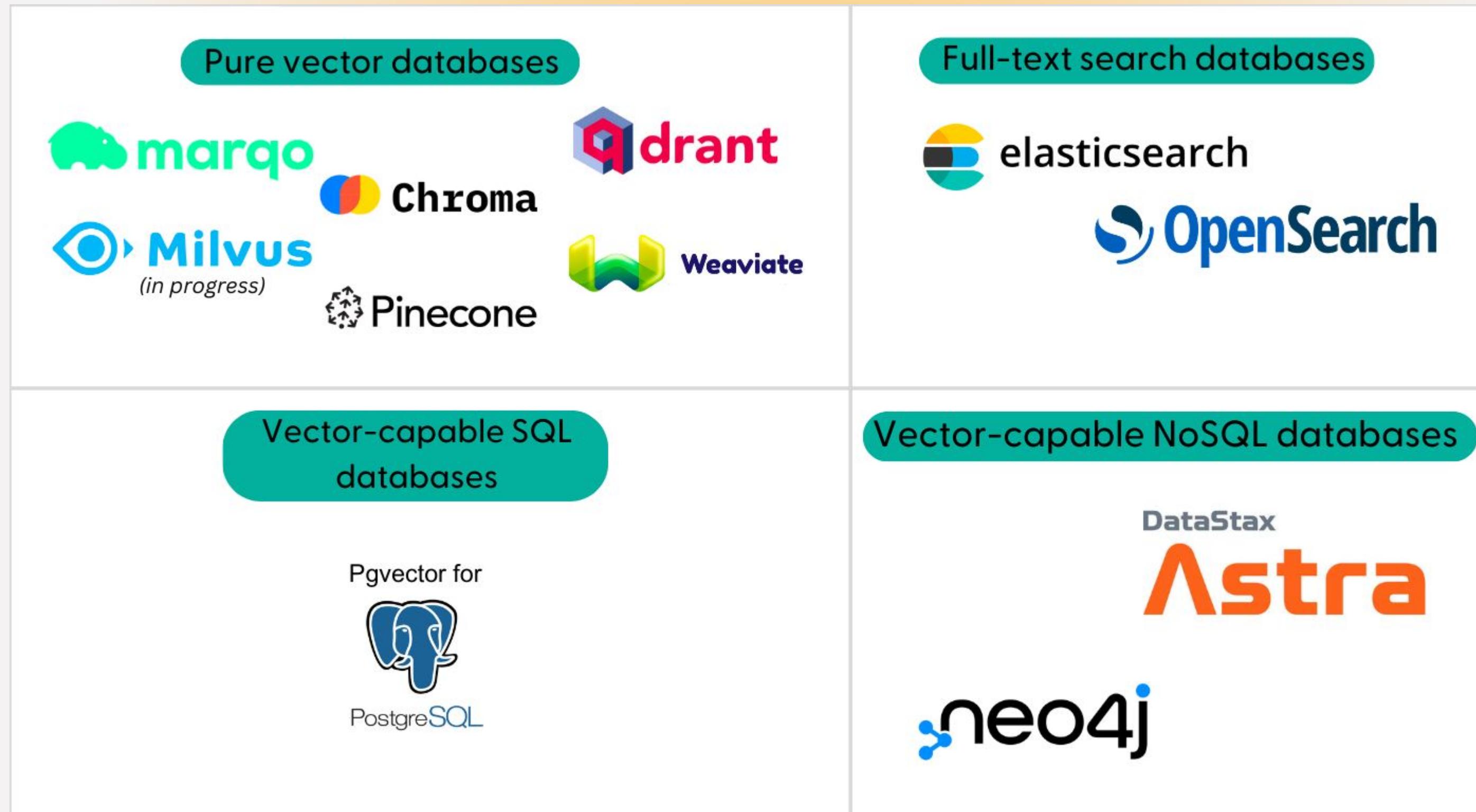
Web RAG



RAG Pipeline



Document Stores



Keyword-based Retrieval (BM25)

	a	and	cheerful	dessert	i	icecream	is	like	season	summer	warm
I like icecream.	0	0	0	0	1	1	0	1	0	0	0
Icecream is a summer dessert.	1	0	0	1	0	1	1	0	0	1	0
Summer is a warm and cheerful season.	1	1	1	0	0	0	1	0	1	1	1

Bag-of-words

	a	and	cheerful	dessert	i	icecream	is	like	season	summer	warm
I like icecream.	0	0	0	0	0.62	0.47	0	0.62	0	0	0
Icecream is a summer dessert.	0.42	0	0	0.55	0	0.42	0.42	0	0	0.42	0
Summer is a warm and cheerful season.	0.32	0.42	0.42	0	0	0	0.32	0	0.42	0.32	0.42

TF-IDF

BM25 - Indexing

```
import glob
from haystack_integrations.document_stores.weaviate.document_store \
    import WeaviateDocumentStore
from haystack import Pipeline
from haystack.components.converters import TextFileToDocument
from haystack.components.preprocessors import DocumentSplitter
from haystack.components.writers import DocumentWriter

document_store = WeaviateDocumentStore(url="http://localhost:8080")

pipe = Pipeline()
pipe.add_component("converter", TextFileToDocument())
pipe.add_component("splitter", DocumentSplitter())
pipe.add_component("writer", DocumentWriter(document_store))
pipe.connect("converter", "splitter")
pipe.connect("splitter", "writer")

file_paths = glob.glob("data/*.txt")
pipe.run({"sources": file_paths})
```

Install Weaviate
[see the docs](#)



RAG with BM25

```
rag_pipeline = Pipeline()
rag_pipeline.add_component(
    name="retriever", instance=WeaviateBM25Retriever(document_store, top_k=5)
)
rag_pipeline.add_component(
    instance=PromptBuilder(template=prompt_template), name="prompt_builder"
)
rag_pipeline.add_component(instance=OllamaGenerator(model="mistral"), name="llm")
rag_pipeline.connect("retriever", "prompt_builder.documents")
rag_pipeline.connect("prompt_builder", "llm")

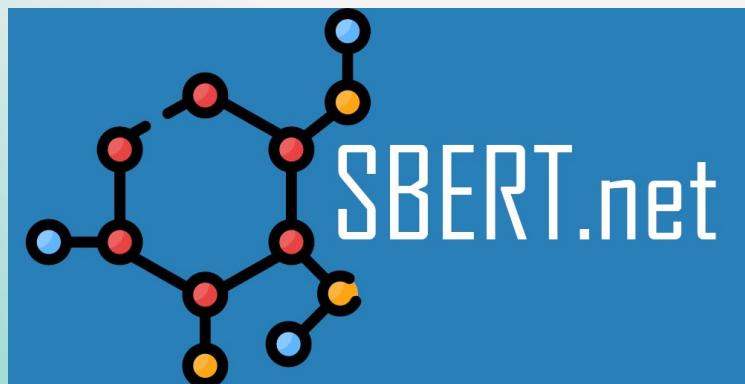
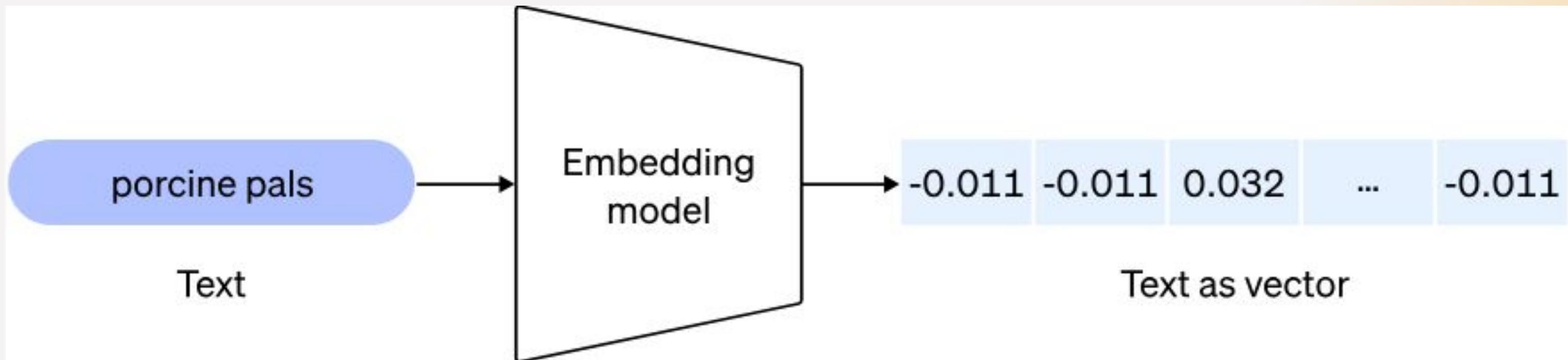
question = "who is Samuel Thomas Fender?"
print(rag_pipeline.run(
{
    "retriever": {"query": question},
    "prompt_builder": {"question": question},
}
))

```



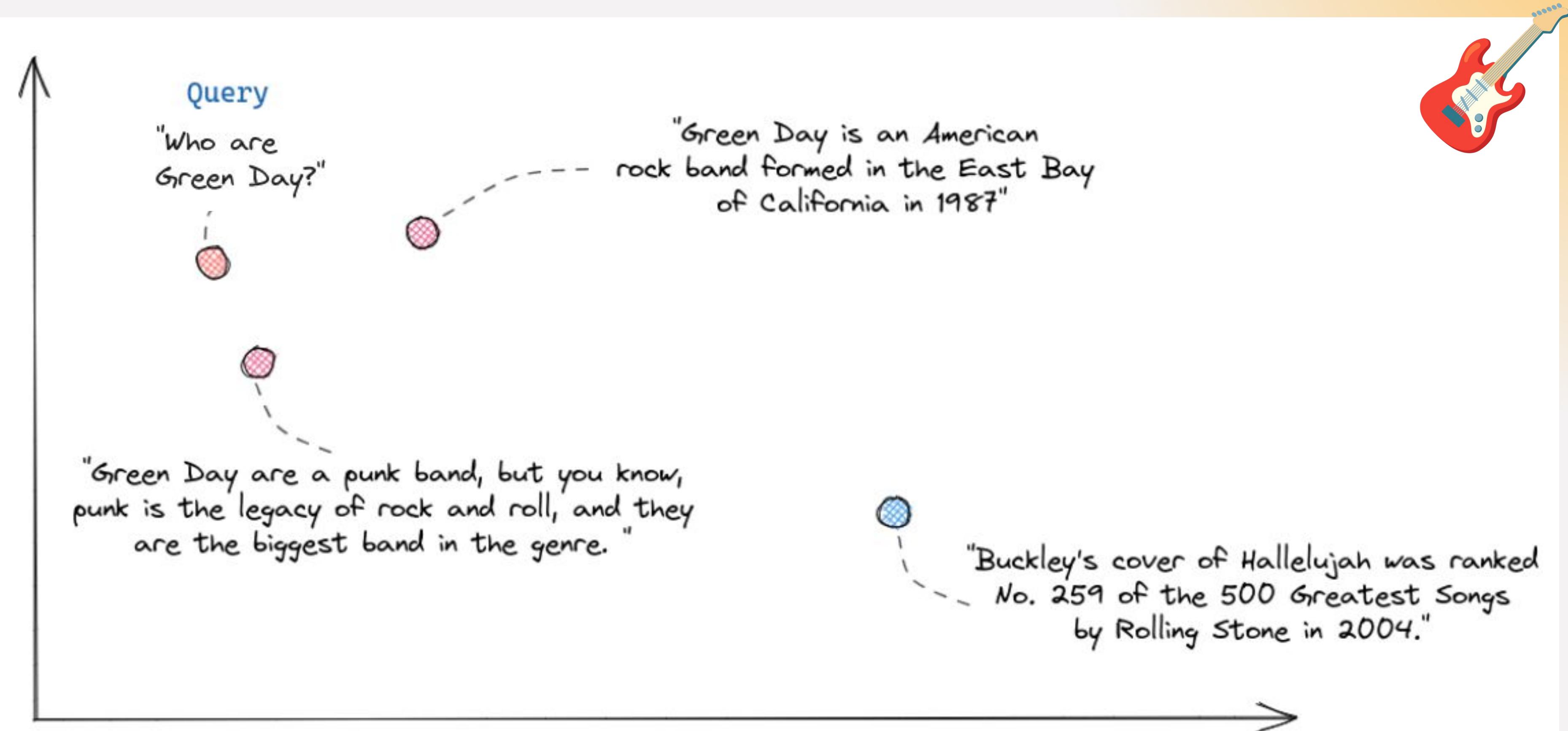
...Samuel Thomas Fender is an English singer, songwriter, and musician...

Embedding models



Sentence Transformers

Embedding Retrieval

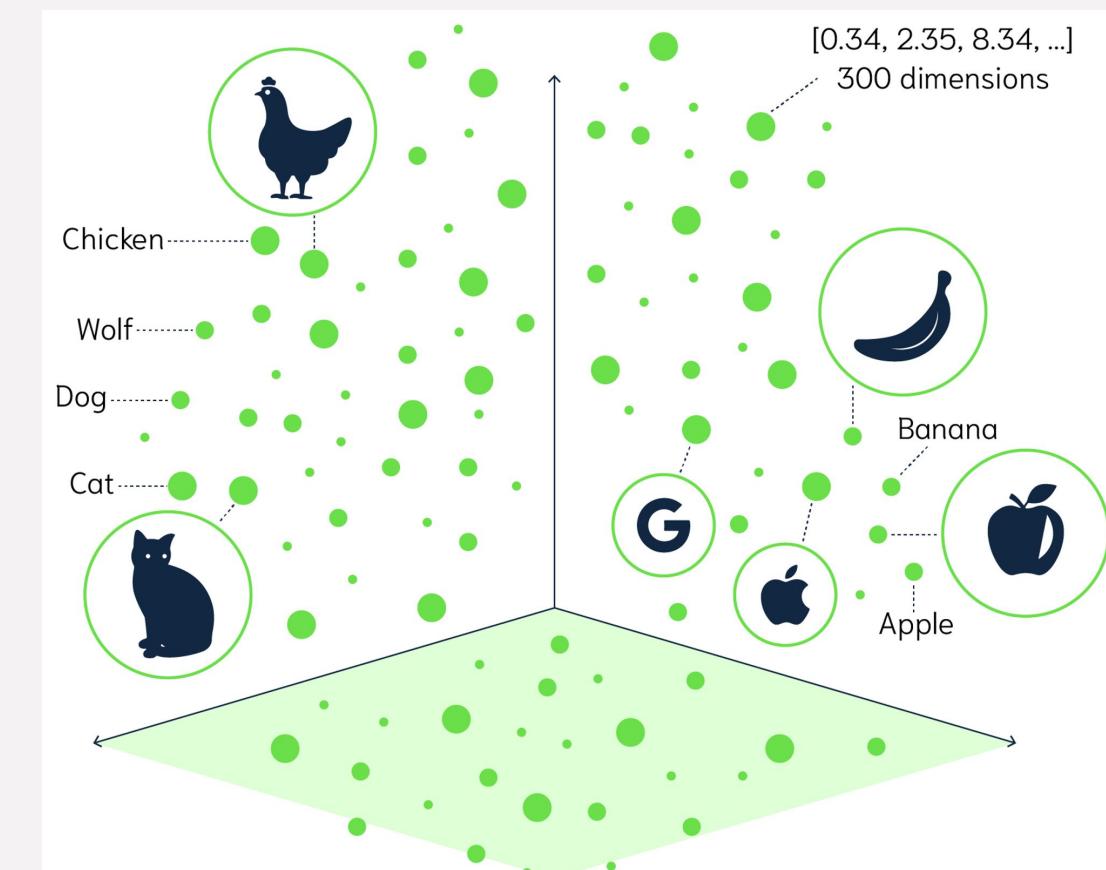


Indexing with embeddings

```
document_store = WeaviateDocumentStore(url="http://localhost:8080")

pipe = Pipeline()
pipe.add_component("converter", TextFileToDocument())
pipe.add_component("splitter", DocumentSplitter())
pipe.add_component("embedder", SentenceTransformersDocumentEmbedder())
pipe.add_component("writer", DocumentWriter(document_store))
pipe.connect("converter", "splitter")
pipe.connect("splitter", "embedder")
pipe.connect("embedder", "writer")

file_paths = glob.glob("data/*.txt")
pipe.run({"sources": file_paths})
```



RAG with embeddings

```
document_store = WeaviateDocumentStore(url="http://localhost:8080")

rag_pipeline = Pipeline()
rag_pipeline.add_component("text_embedder", SentenceTransformersTextEmbedder())
rag_pipeline.add_component(
    name="retriever", instance=WeaviateEmbeddingRetriever(document_store=document_store, top_k=5)
)
rag_pipeline.add_component(
    instance=PromptBuilder(template=prompt_template), name="prompt_builder"
)
rag_pipeline.add_component(instance=OllamaGenerator(model="mistral"), name="llm")
rag_pipeline.connect("text_embedder", "retriever")
rag_pipeline.connect("retriever", "prompt_builder.documents")
rag_pipeline.connect("prompt_builder", "llm")

question = """Audioslave was formed by members of two bands.  
Can you name the bands and summarize the Audioslave sound with a short bulleted list?"""
print(rag_pipeline.run(
{
    "text_embedder": {"text": question},
    "prompt_builder": {"question": question},
}
))

```



RAG with embeddings - result

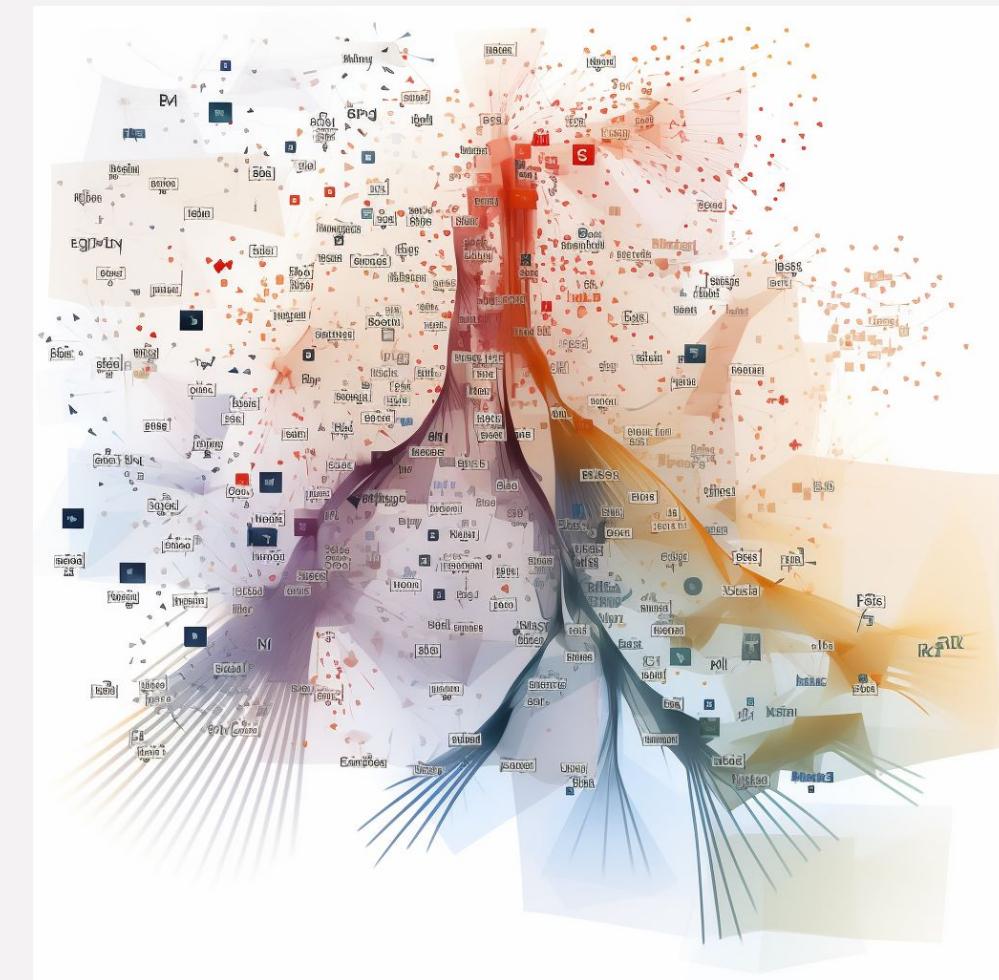
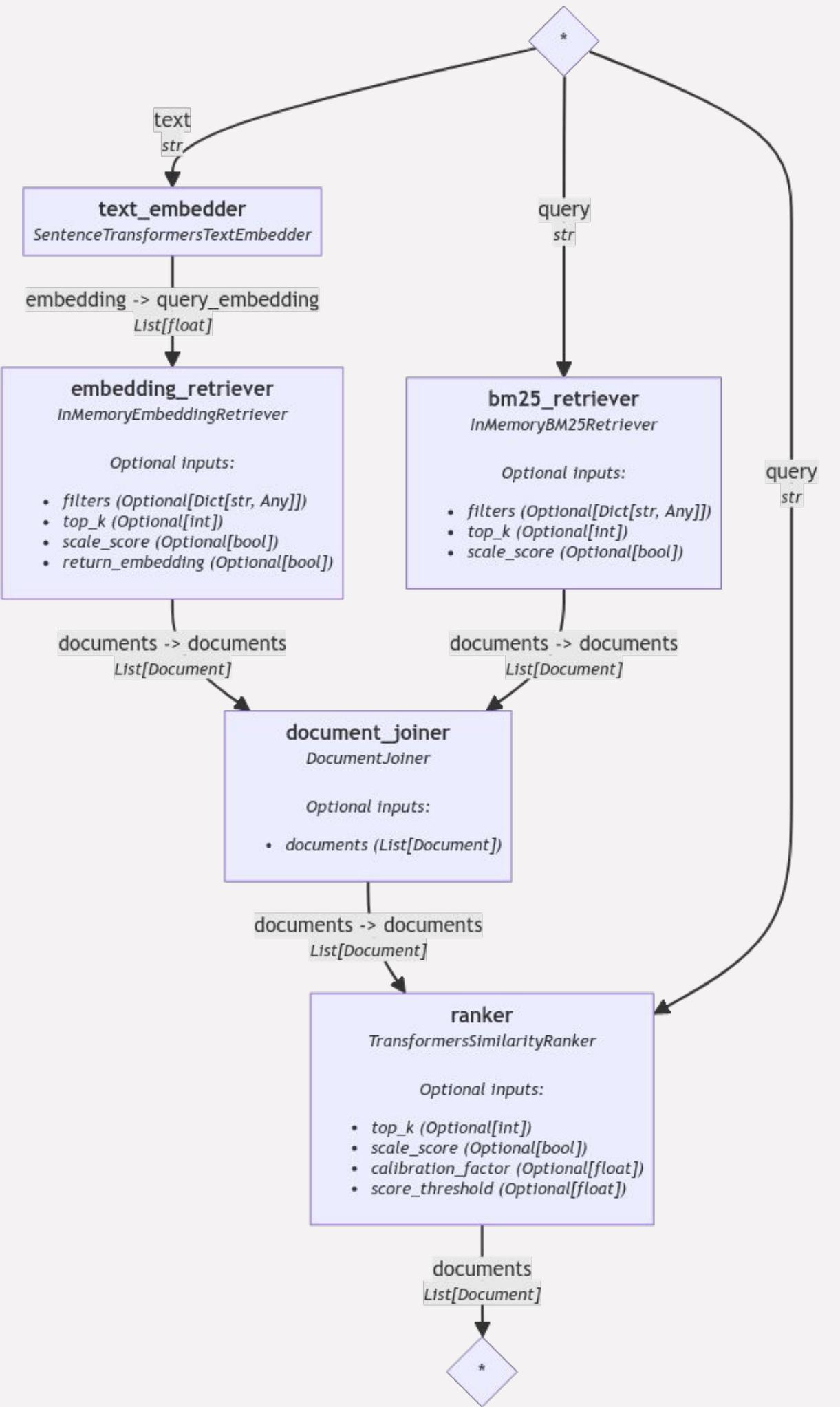
Audioslave was formed by members of Soundgarden and Rage Against the Machine.

The band's sound can be described as follows:

- A fusion of grunge, alternative rock, and heavy metal
- Chris Cornell's powerful vocals leading the way
- Tom Morello and Brad Wilk's innovative rhythmic grooves from Rage Against the Machine
- Heavy use of distortion and dynamic contrasts
- Socially conscious lyrics, but also exploring personal themes
- A strong live presence with energetic performances.



Hybrid Retrieval



Choose a LLM inference solution



Proprietary models

providers: OpenAI, Azure, Google, Anthropic, Cohere, Mistral...
Amazon Bedrock: models from different providers

Open models hosted by providers

Shared Hosted models
 Privately Hosted models

HF Inference API
Cloud providers: Anyscale, Deep Infra, Fireworks, Lemonfox.ai, OctoAI, Together AI...

Open models on-premise

Local experimentation
 Serving in production

GPU: HF Transformers
CPU (+GPU): llama.cpp, Ollama...

vLLM, HF TGI...

Deployment

- Pipelines can be serialized to YAML
- They can be served using & [Hayhooks](#)

```
pip install hayhooks

hayhooks run --pipelines-dir .

"""

INFO:      Deployed pipeline: embed_rag
INFO:      Deployed pipeline: embed_indexing
INFO:      Started server process [13537]
INFO:      Waiting for application startup.
INFO:      Application startup complete.
INFO:      Uvicorn running on http://localhost:1416 (Press CTRL+C to quit) """
```

Deployment: Hayhooks in action

default

POST /embed_rag Embed Rag

Parameters

No parameters

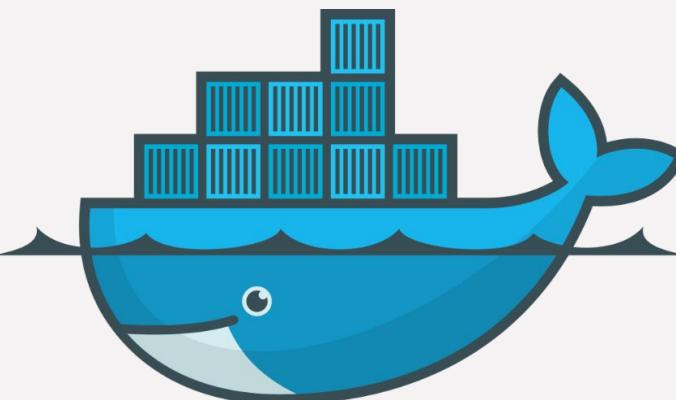
Request body required

application/json

Example Value | Schema

```
{  
    "llm": {  
        "generation_kwargs": {}  
    },  
    "prompt_builder": {  
        "question": ""  
    },  
    "retriever": {  
        "filters": {},  
        "top_k": 0,  
        "distance": 0,  
        "certainty": 0  
    },  
    "text_embedder": {  
        "text": "string"  
    }  
}
```

FastAPI



Beyond RAG...



在意、英、法三国语言的多语种RAG
from a 🎧 podcast

haystack by deepset OpenAI Whisper Adrantly 8X7



Haystack Cookbook

Get the summaries of latest top
Hacker News posts ❤️

How many of the top posts should I summarize?

Get summaries

0

5

Summary: A new tool for writing Pkl code has been released, offering plugins and extensions for IntelliJ, Visual Studio Code, and Neovim. Language Server Protocol support is coming soon. URL: <https://pkl-lang.org/index.html>

Summary: Mozilla has commissioned a report detailing how Microsoft prevents effective browser choice on Windows. The report documents how Microsoft places its own browser, Edge, at the center of its operating system and weaponizes Windows' user interface design to undermine people selecting rival browsers. URL: <https://research.mozilla.org/browser-competition/over-the-edge-the-use-of-design-tactics-to-undermine-browser-choice/>

Beyond RAG...

ARTIFICIAL INTELLIGENCE / MICROSOFT / BUSINESS

Sam Altman to return as CEO of OpenAI



Sam Altman speaks during the OpenAI DevDay event on November 6th, 2023. Photo by Justin Sullivan / Getty Images

/ After an attempted coup by OpenAI's board that lasted five days, Altman is returning alongside co-founder Greg Brockman.

By Nilay Patel and Alex Heath
Nov 22, 2023, 7:03 AM GMT+1

| 202 Comments (202 New)

Sam Altman will return as CEO of OpenAI, overcoming an attempted boardroom coup that sent the company into chaos over the past several days. Former president Greg Brockman, who quit in protest of Altman's firing, will return as well.

The company said in a statement late Tuesday that it has an "agreement in principle" for Altman to return alongside a new board composed of Bret Taylor, Larry Summers, and Adam D'Angelo. D'Angelo is a holdover from the previous board that initially fired Altman on Friday. He remains on this new board to give the previous board some representation, we're told.

Information extraction via LLMs

people: ['Sam Altman', 'Greg Brockman', 'Bret Taylor', 'Larry Summers', 'Adam D'Angelo', 'Ilya Sutskever', 'Emmett Shear']

companies: ['OpenAI', 'Microsoft', 'Thrive Capital']

summary: Sam Altman will return as CEO of OpenAI, overcoming an attempted boardroom coup that sent the company into chaos over the past several days.

topics: ['OpenAI', 'Artificial intelligence', 'Machine learning', 'Computer vision', 'Natural language processing']

about_animals: False

about_ai: True



Haystack Cookbook

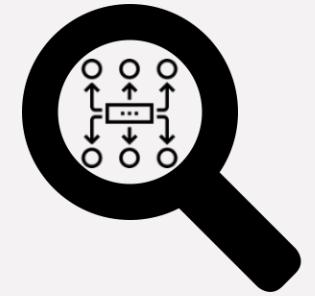
There is much more!



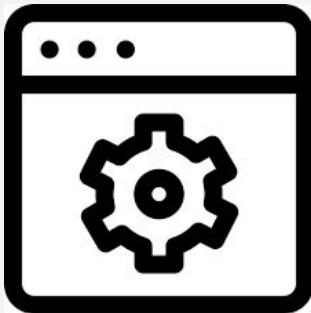
File
conversion



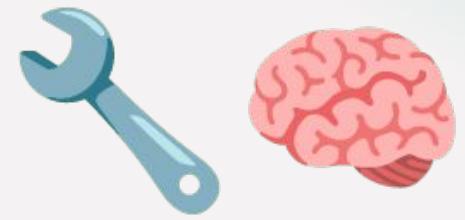
Evaluation



Observability

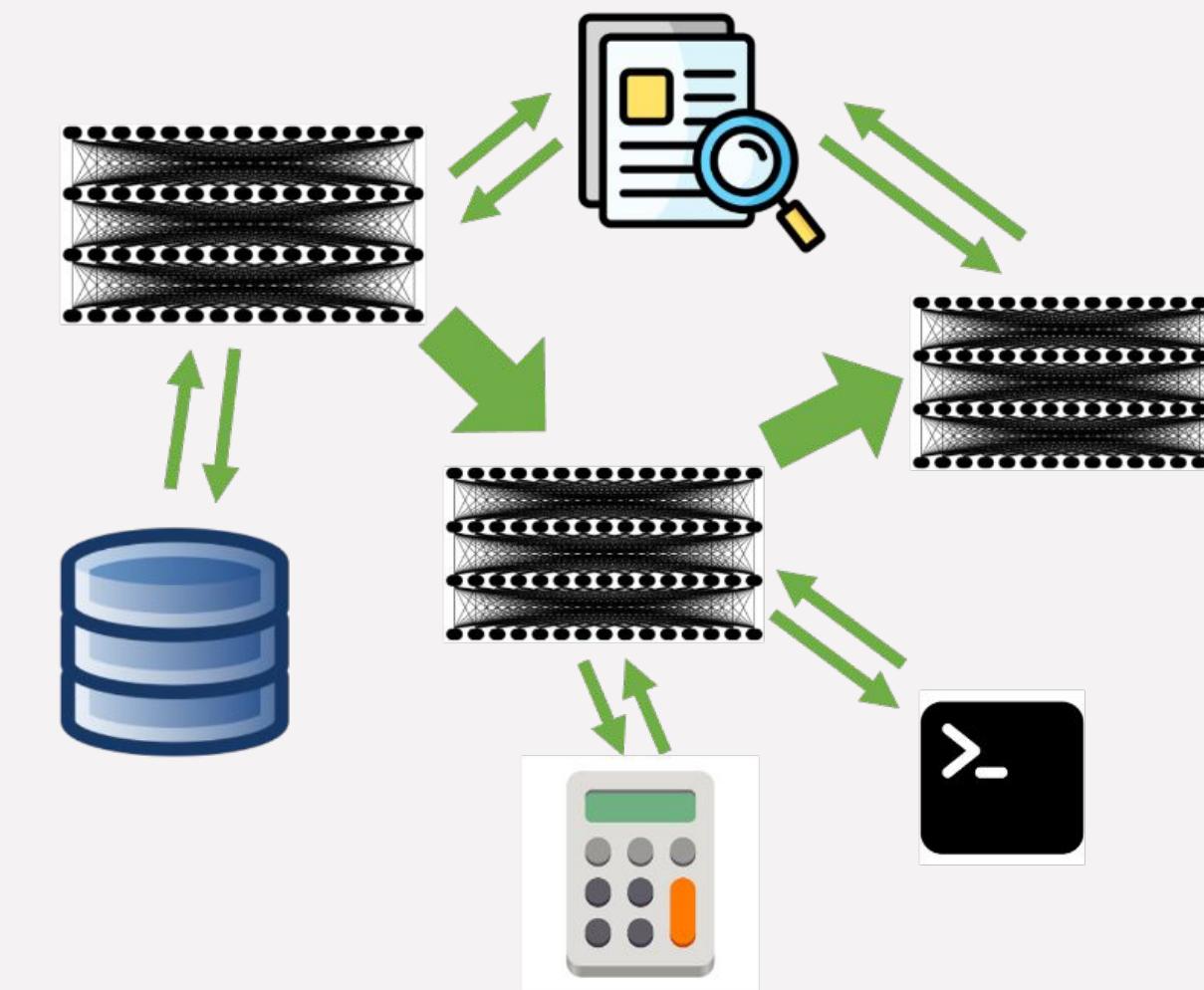
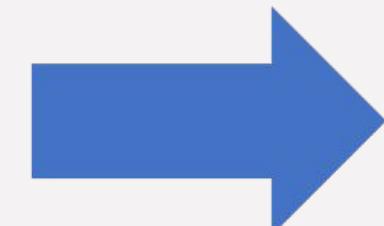
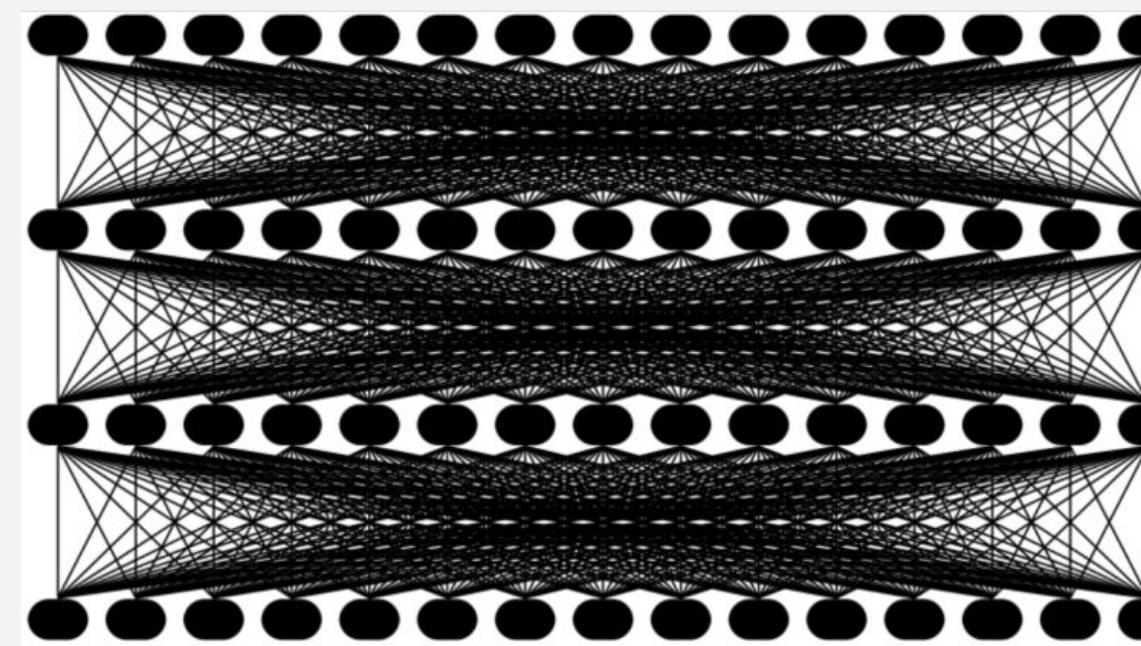


Function
calling



Agents

From models to Compound AI Systems



Berkeley AI Research blogpost

Thank you!

Check out 
[Haystack](#)



LinkedIn: [Stefano Fiorucci](#)
GitHub: [@anakin87](#)