Talk to your Documents

by **Plamen Mushkov** Oracle ACE Associate

APEXappLab.dev

blog.APEXappLab.dev

Plamen **M**ushkov

@plamen_9

insum
A Talan Company

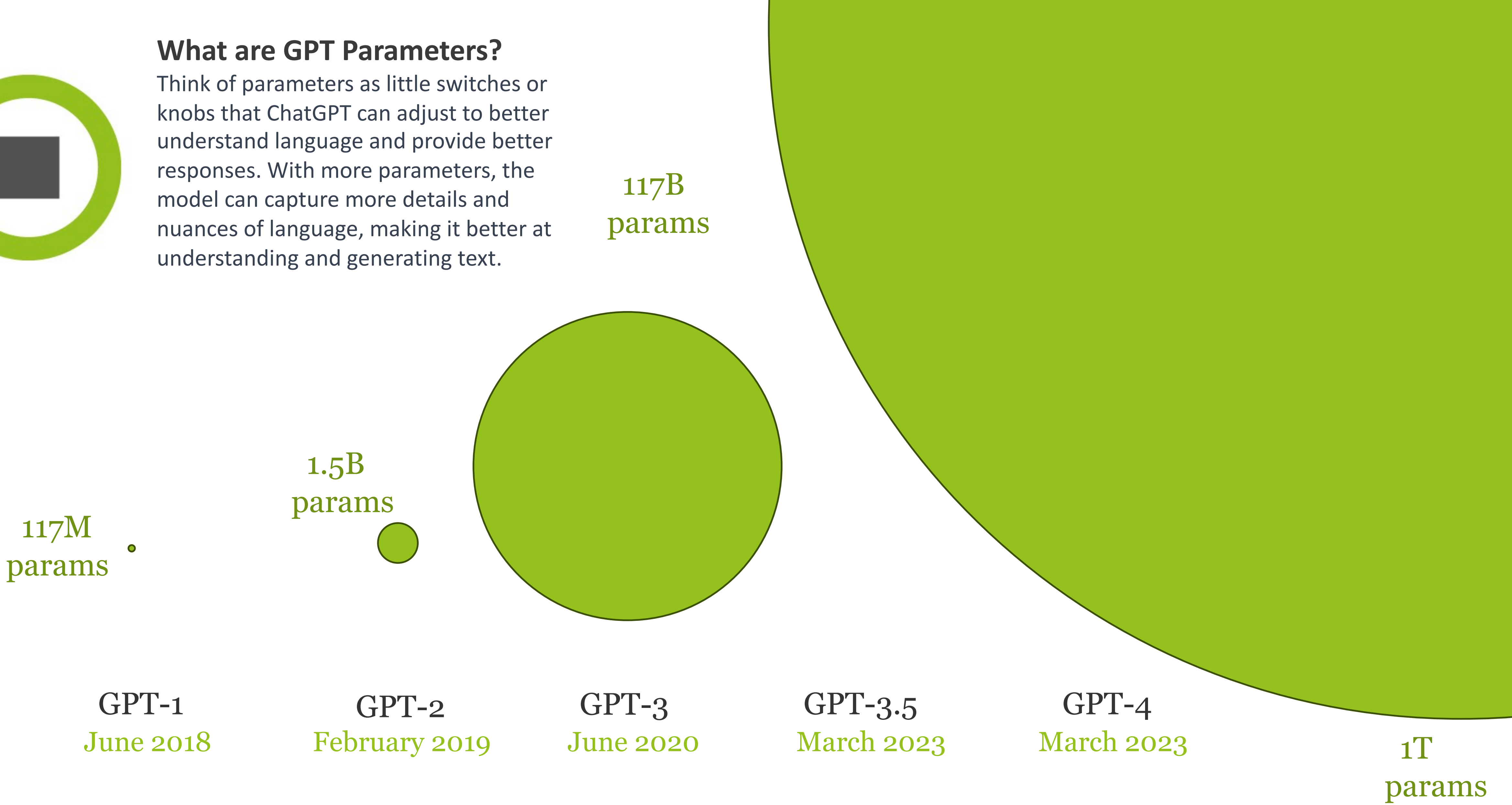
Intro

30 November 2022

ChatGPT

What are GPT Parameters?

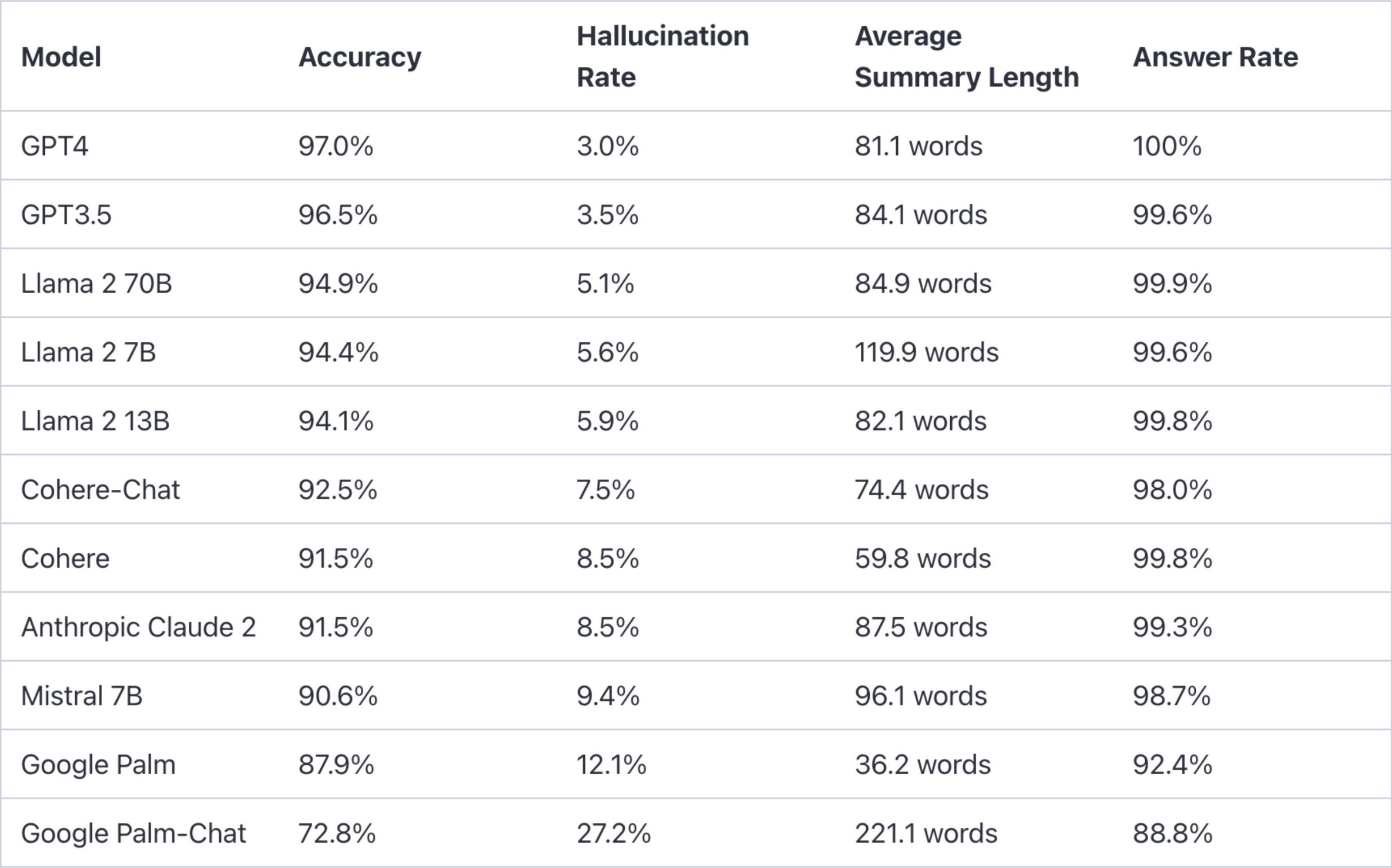
Think of parameters as little switches or knobs that ChatGPT can adjust to better understand language and provide better responses. With more parameters, the model can capture more details and nuances of language, making it better at understanding and generating text.



Comparing LLMs

	ChatGPT **OpenAI**	Llama 2 **Meta**	Gemini **Google**	Claude **Anthropic**	Cohere
Open source	No	Yes	No	No	
Context	GPT-3.5 / GPT-4 Turbo 4k / 128k tokens	4k to 32k	32k	100k	4k
Price	GPT4 starting at $30/1m tokens	Free	$20 per month for the Ultra model	Claude Instant / Claude 2 $5.5/1m / $32/1m	Cohere medium/Cohere Xlarge $2/1m / $15/1m
Parameters	GPT-3.5 / GPT-4 175b / 1t	7b / 13b/ 70b	Nano – 1.8b / 3.25b	137b	Cohere medium/Cohere Xlarge 6.1b / 52.4b
Notes			Gemma 2B and Gemma 7B are Google's open-source models		

Comparing LLMs



Model	Accuracy	Hallucination Rate	Average Summary Length	Answer Rate
GPT4	97.0%	3.0%	81.1 words	100%
GPT3.5	96.5%	3.5%	84.1 words	99.6%
Llama 2 70B	94.9%	5.1%	84.9 words	99.9%
Llama 2 7B	94.4%	5.6%	119.9 words	99.6%
Llama 2 13B	94.1%	5.9%	82.1 words	99.8%
Cohere-Chat	92.5%	7.5%	74.4 words	98.0%
Cohere	91.5%	8.5%	59.8 words	99.8%
Anthropic Claude 2	91.5%	8.5%	87.5 words	99.3%
Mistral 7B	90.6%	9.4%	96.1 words	98.7%
Google Palm	87.9%	12.1%	36.2 words	92.4%
Google Palm-Chat	72.8%	27.2%	221.1 words	88.8%

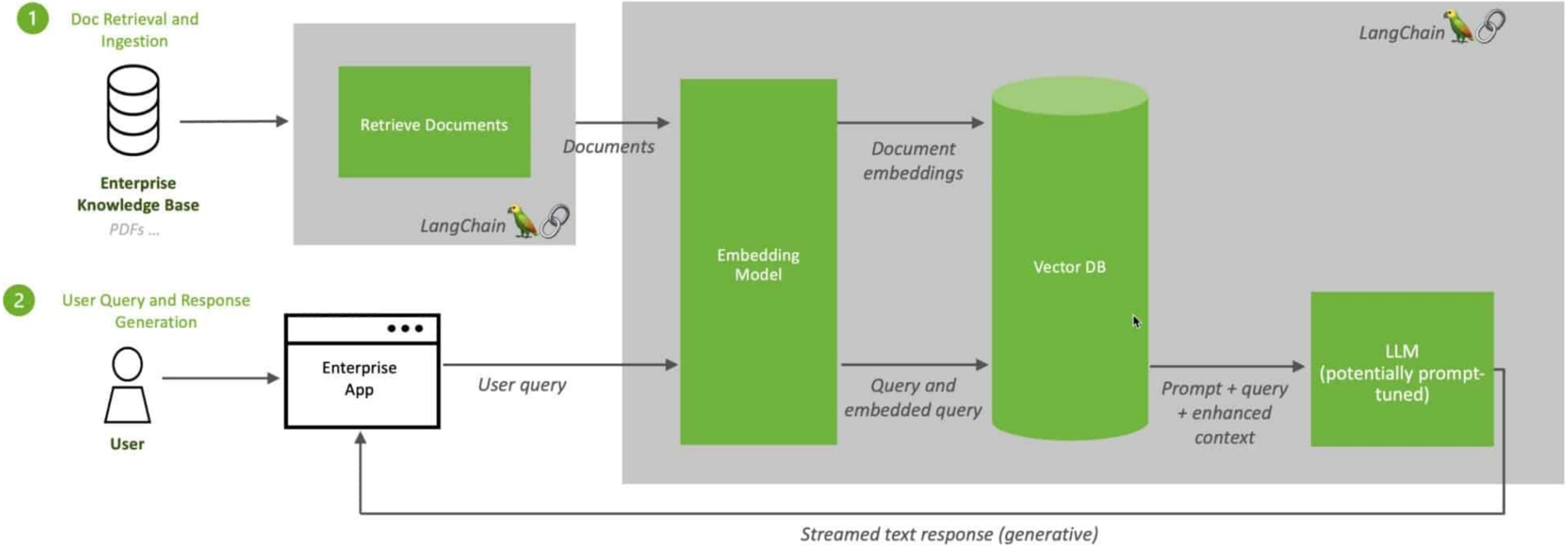
Retrieval Augmented Generation

(RAG)

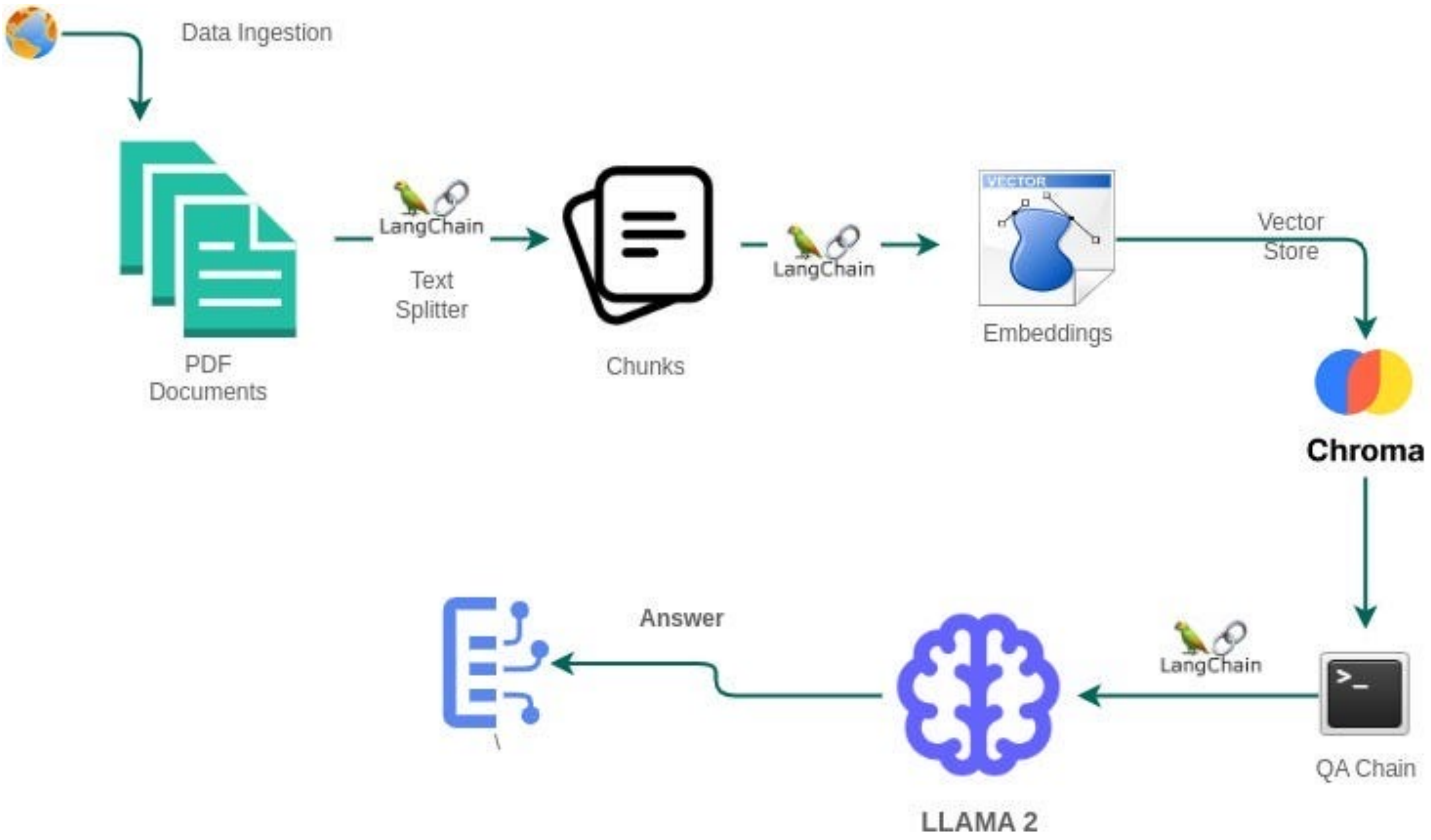
Why do we need a RAG and not just LLM?

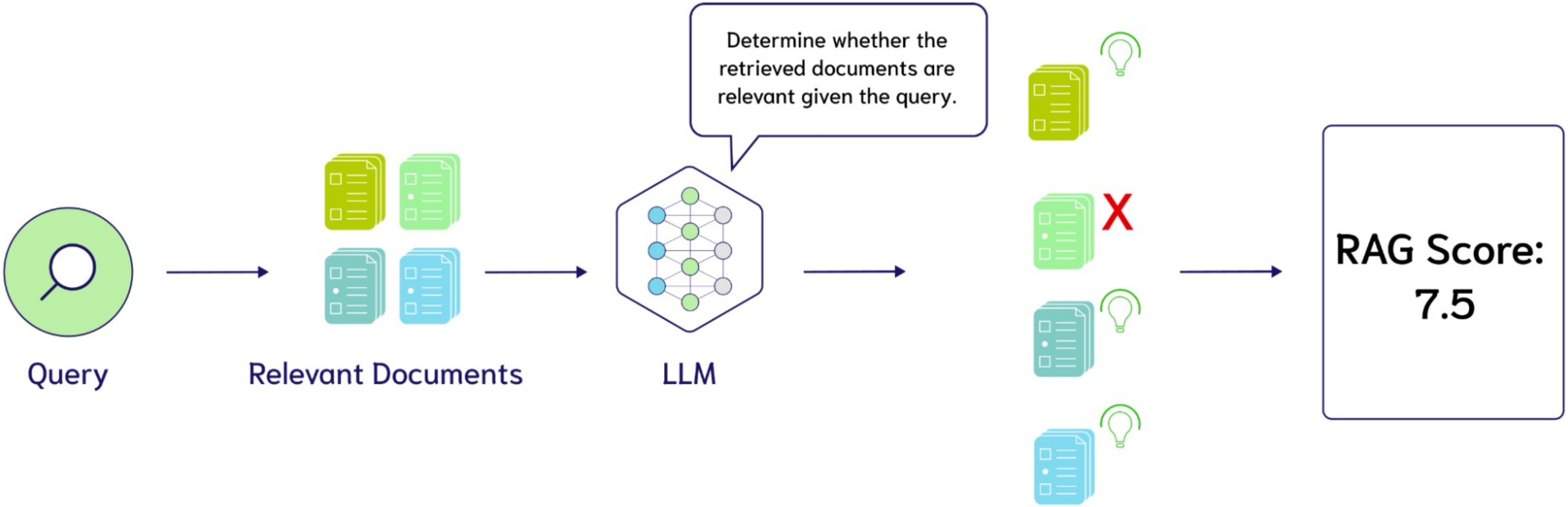
- RAG allows LLMs to use DOMAIN SPECIFIC data, that is otherwise unknown to them
- RAG gives the LLM a long-term memory
- RAG allows pointing to the specific source of information, unlike LLMs
- Information is always up-to-date (as long as you keep your Vector database updated)
- Significantly reduces hallucinations
- RAG makes using LLMs cost-effective – less tokens are used for setting the context
- You don’t need to set the context of LLMs over and over again
- Allows using a smaller models which require less resources

Retrieval Augmented Generation (RAG) Sequence Diagram



1
Doc Retrieval and Ingestion
Enterprise Knowledge Base
PDFs ...
Retrieve Documents
LangChain
Documents
Embedding Model
Document embeddings
Vector DB
LangChain
2
User Query and Response Generation
User
Enterprise App
User query
Query and embedded query
Prompt + query + enhanced context
LLM (potentially prompt-tuned)
Streamed text response (generative)


Data Ingestion
PDF Documents
LangChain
Text Splitter
Chunks
LangChain
VECTOR
Embeddings
Vector Store
Chroma
QA Chain
LangChain
LLAMA 2
Answer



Determine whether the retrieved documents are relevant given the query.

Query

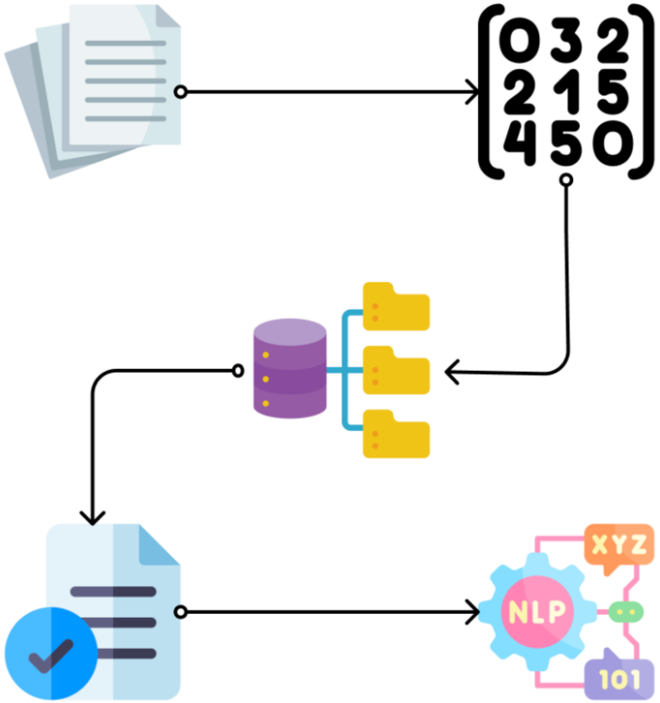
Relevant Documents

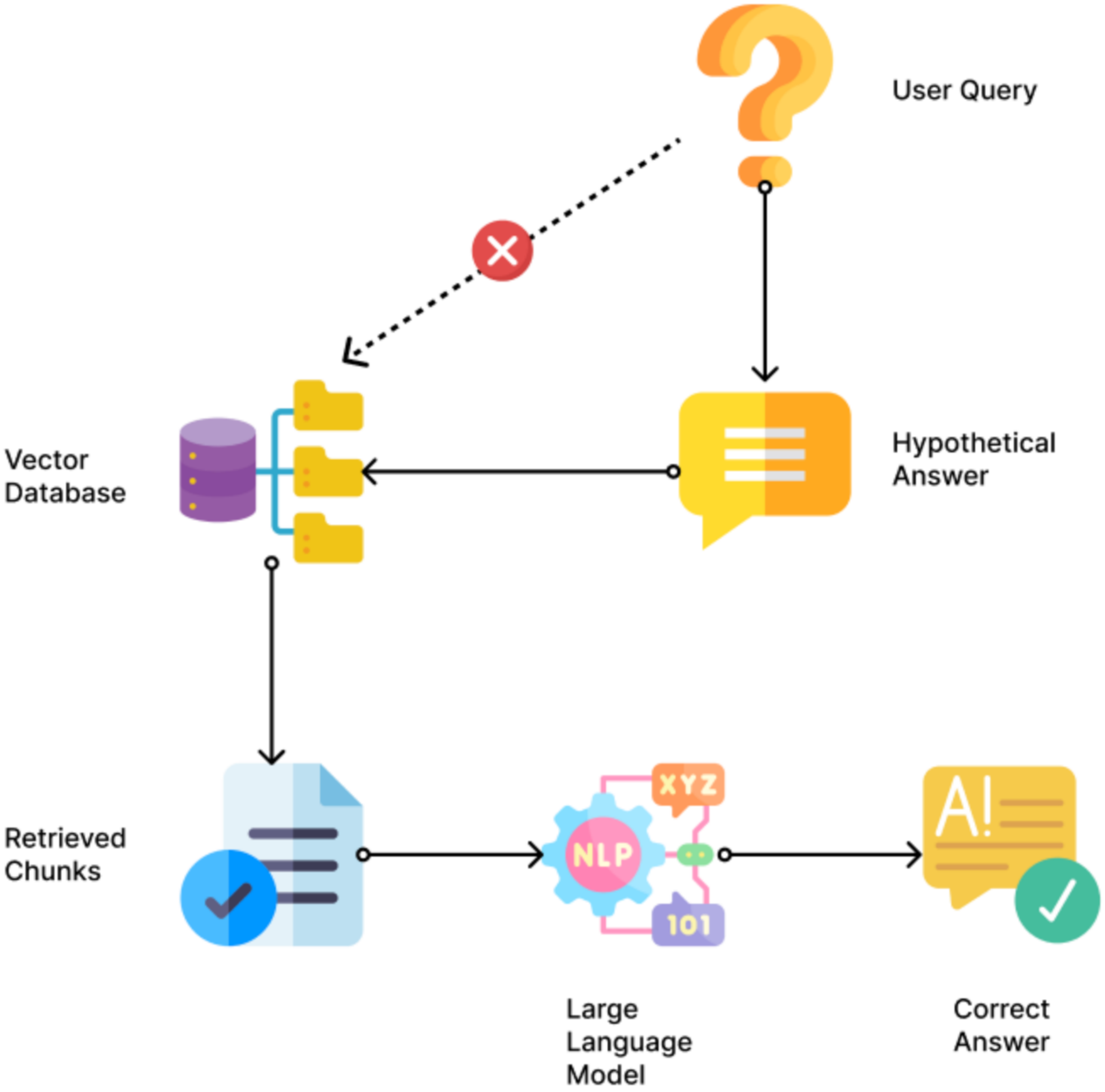
LLM

RAG Score: 7.5

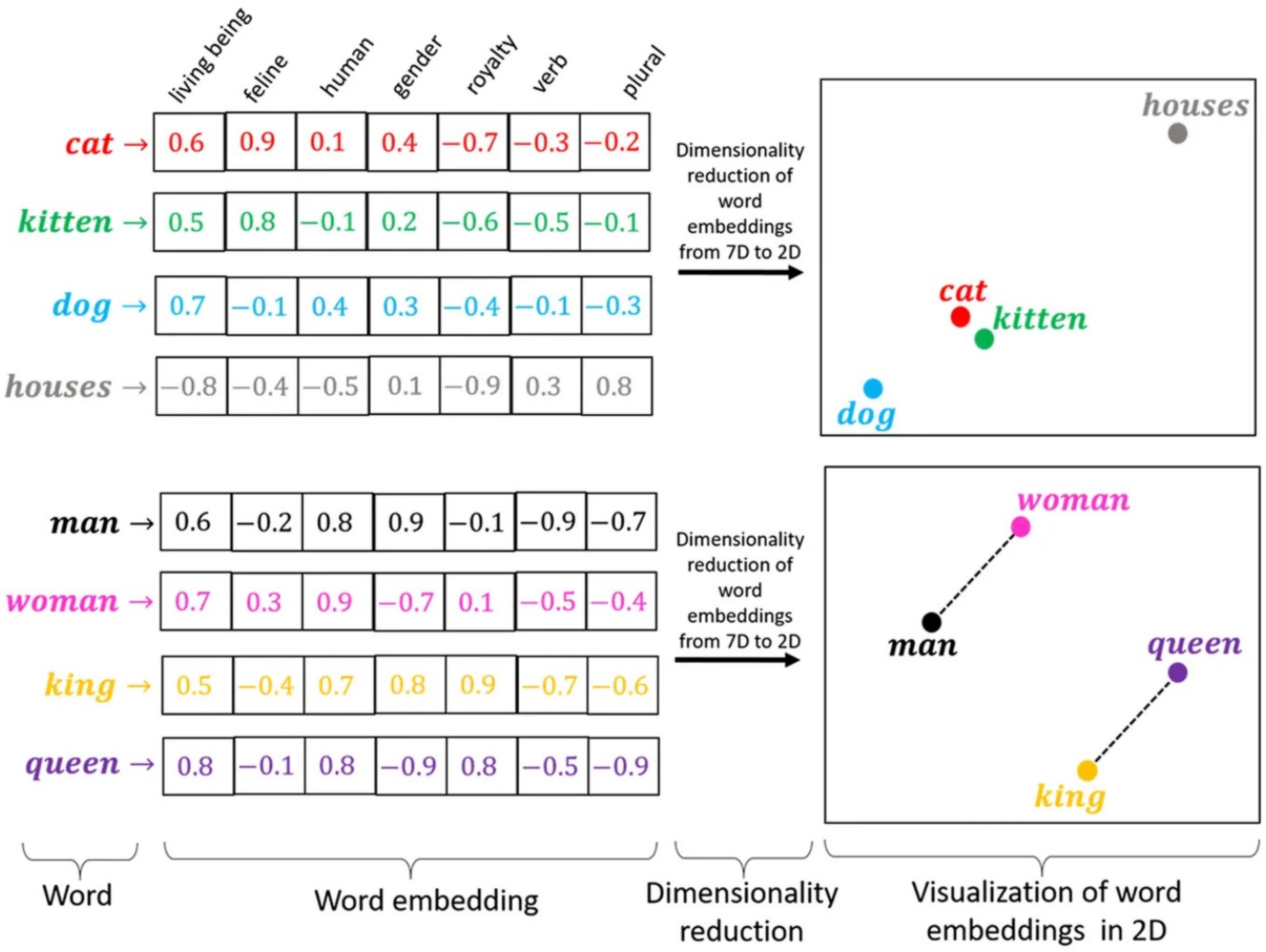
Vector Databases

Vector Databases





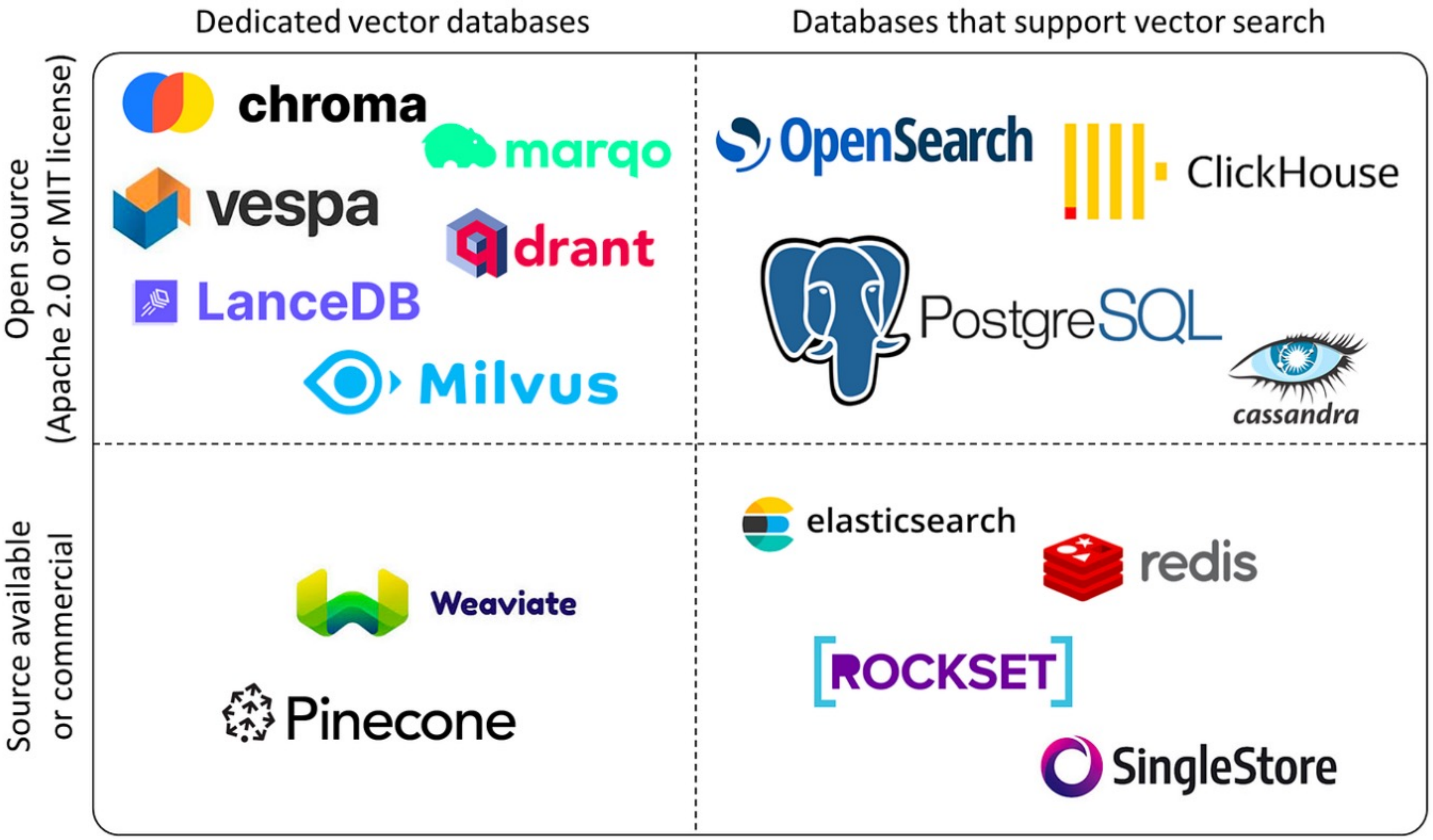
Vector Databases



	living being	feline	human	gender	royalty	verb	plural
cat →	0.6	0.9	0.1	0.4	−0.7	−0.3	−0.2
kitten →	0.5	0.8	−0.1	0.2	−0.6	−0.5	−0.1
dog →	0.7	−0.1	0.4	0.3	−0.4	−0.1	−0.3
houses →	−0.8	−0.4	−0.5	0.1	−0.9	0.3	0.8
man →	0.6	−0.2	0.8	0.9	−0.1	−0.9	−0.7
woman →	0.7	0.3	0.9	−0.7	0.1	−0.5	−0.4
king →	0.5	−0.4	0.7	0.8	0.9	−0.7	−0.6
queen →	0.8	−0.1	0.8	−0.9	0.8	−0.5	−0.9

Dimensionality reduction of word embeddings from 7D to 2D
houses
cat
kitten
dog
Dimensionality reduction of word embeddings from 7D to 2D
woman
man
queen
king
Word
Word embedding
Dimensionality reduction
Visualization of word embeddings in 2D

Vector Databases



Dedicated vector databases

Databases that support vector search

Open source (Apache 2.0 or MIT license)

Source available or commercial

LangChain 🦜🔗

LangChain

- Orchestration Framework
- Opensource
- Helps integrate LLMs into applications
- Workflow and decision making
- API wrapper
- Offers Python and JS libraries

LangChain

150+

- 001

Document Loaders

60+

- 002

Vector Stores

50+

- 003

Embedding Models

GitHub

CODE

JSON

API

MARKDOWN

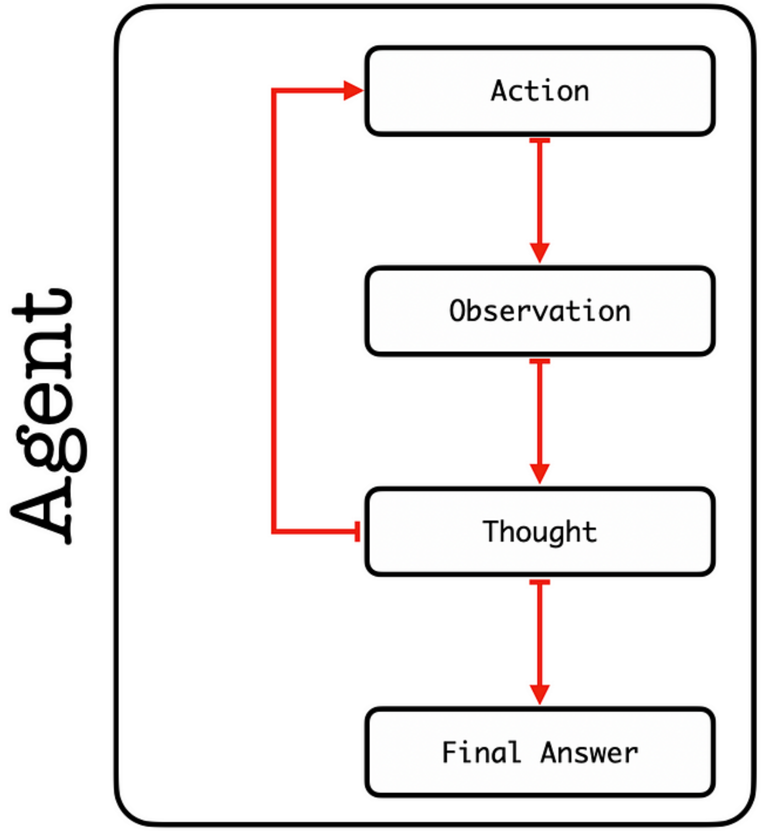
HTML

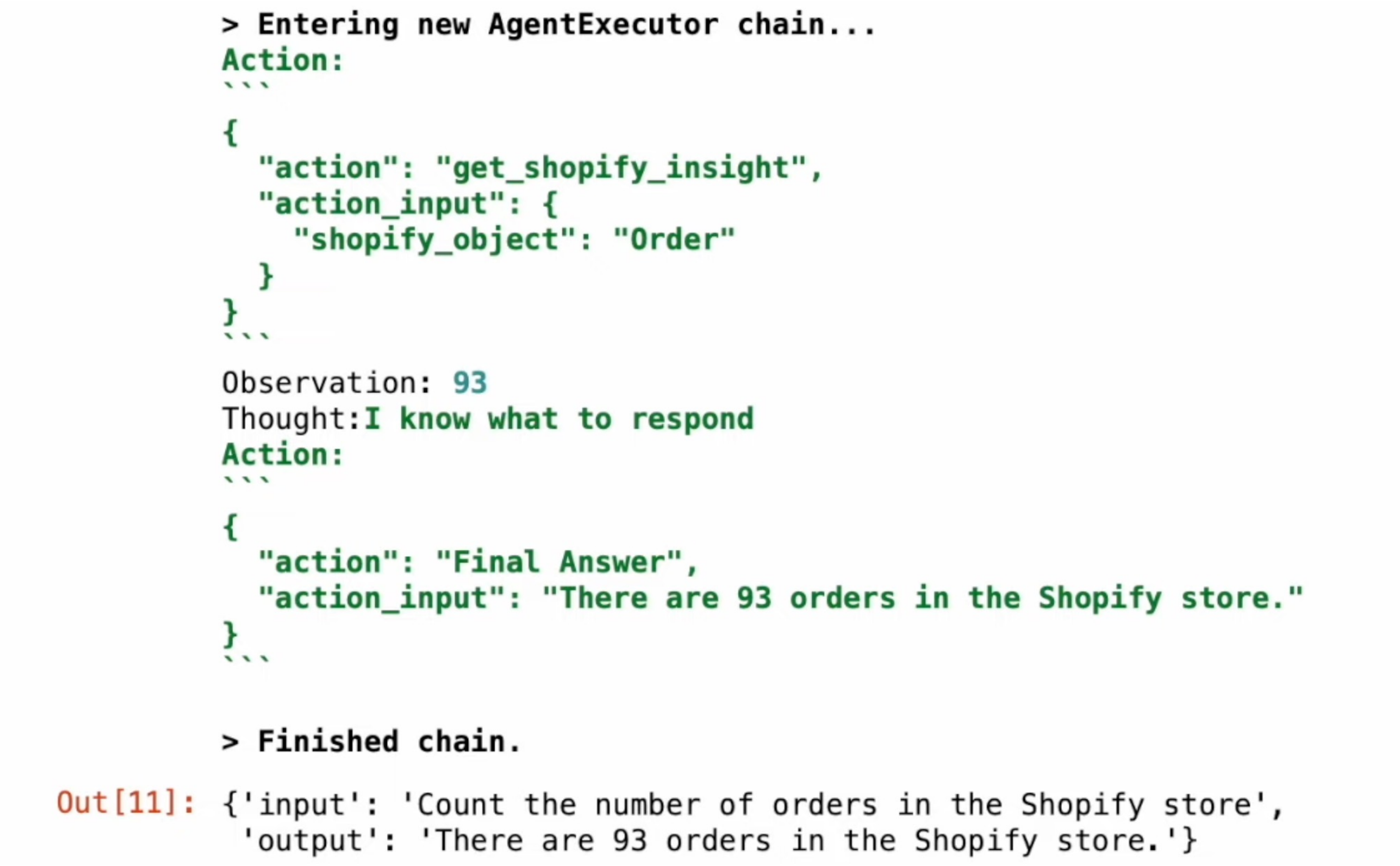
TEXT

SQL

PDF

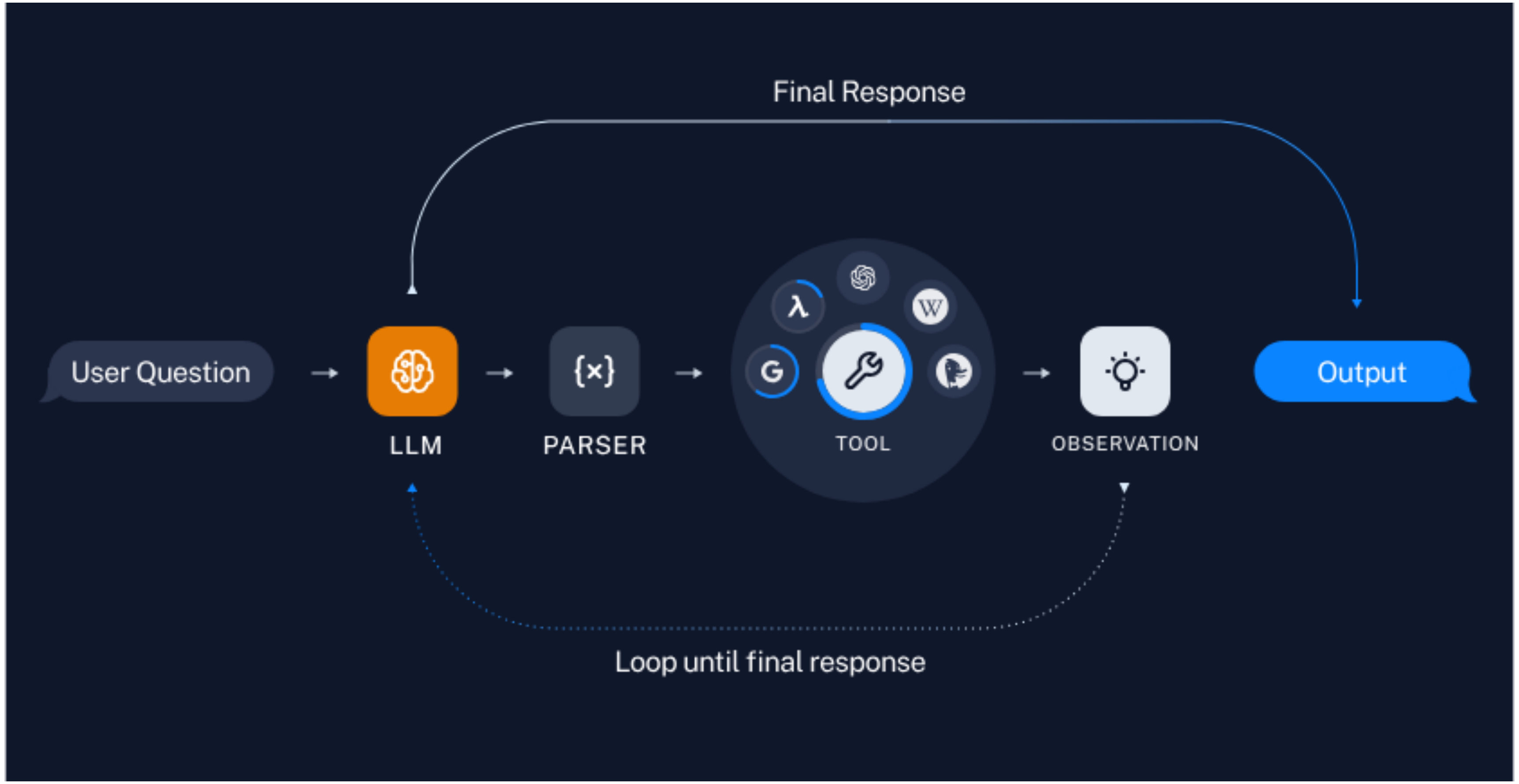
LangChain



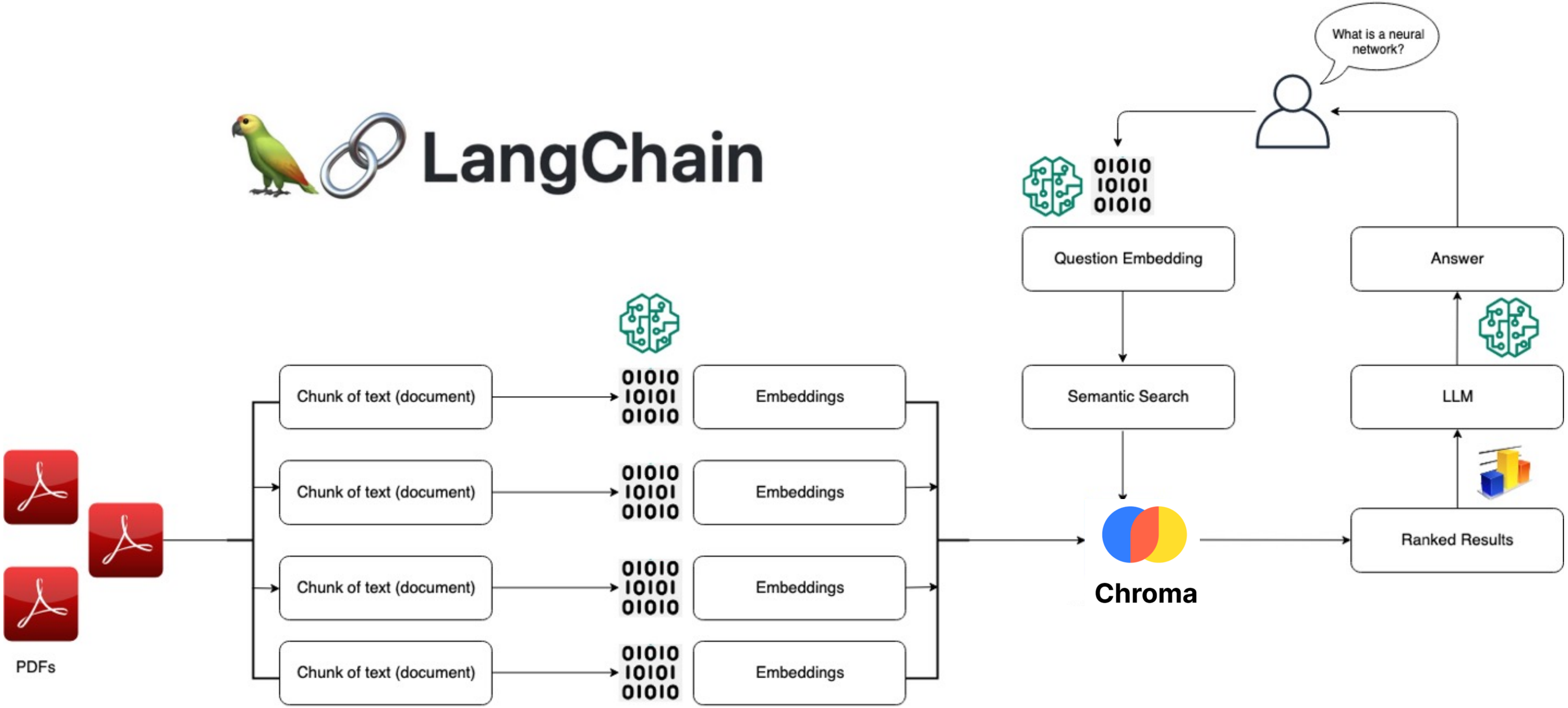


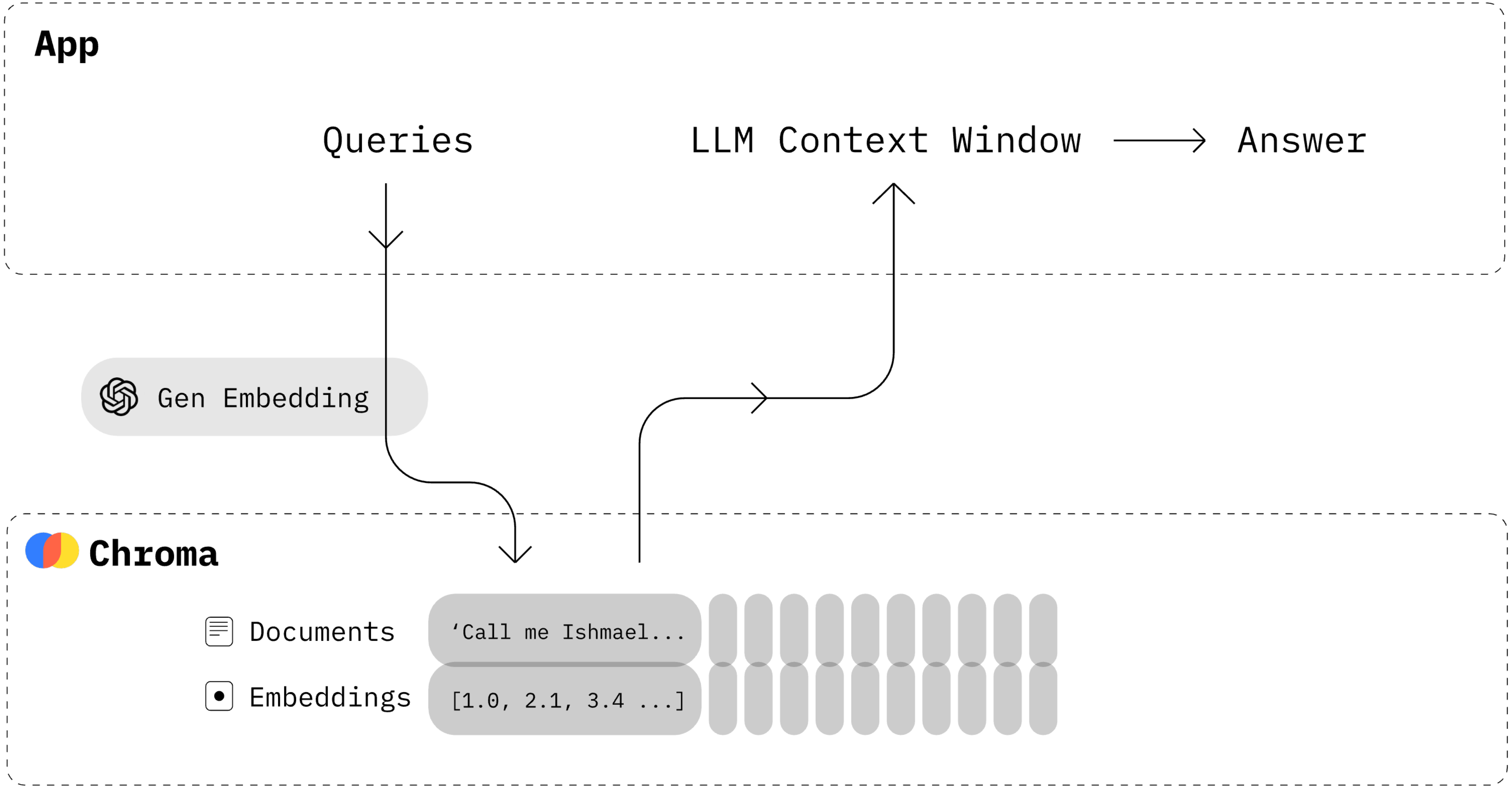
```
> Entering new AgentExecutor chain...
Action:
```
{
  "action": "get_shopify_insight",
  "action_input": {
    "shopify_object": "Order"
  }
}
```
Observation: 93
Thought:I know what to respond
Action:
```
{
  "action": "Final Answer",
  "action_input": "There are 93 orders in the Shopify store."
}
```

> Finished chain.
Out[11]: {'input': 'Count the number of orders in the Shopify store',
          'output': 'There are 93 orders in the Shopify store.'}
```

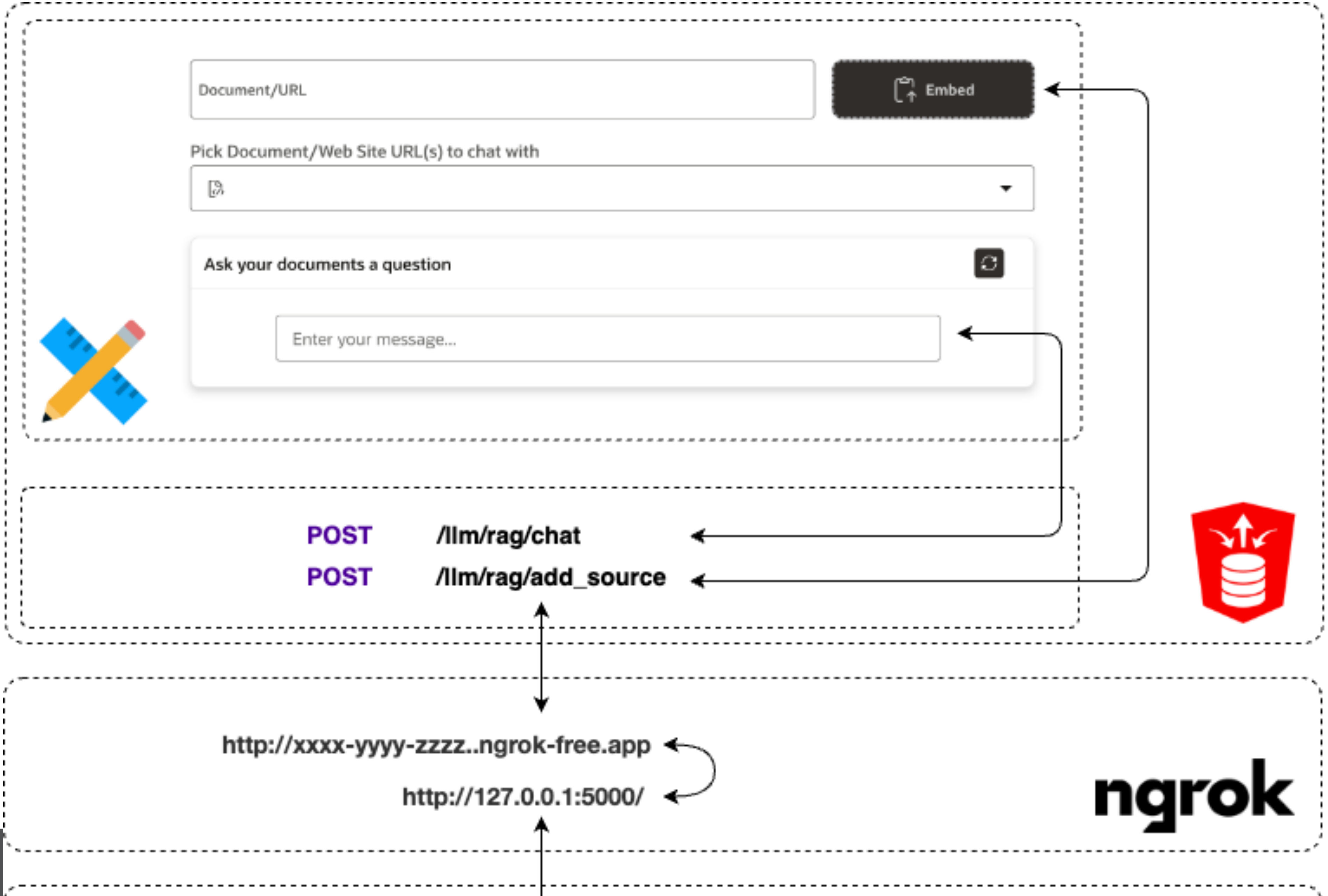
LangChain


Final Response
User Question
LLM
{x}
PARSER
TOOL
OBSERVATION
Output
Loop until final response

My setup


LangChain
What is a neural network?
Question Embedding
Answer
Semantic Search
LLM
Ranked Results
Chroma
Chunk of text (document)
Chunk of text (document)
Chunk of text (document)
Chunk of text (document)
01010
10101
01010
Embeddings
Embeddings
Embeddings
Embeddings
PDFs


App
Queries
LLM Context Window
Answer
Gen Embedding
Chroma
Documents
'Call me Ishmael...
Embeddings
[1.0, 2.1, 3.4 ...]

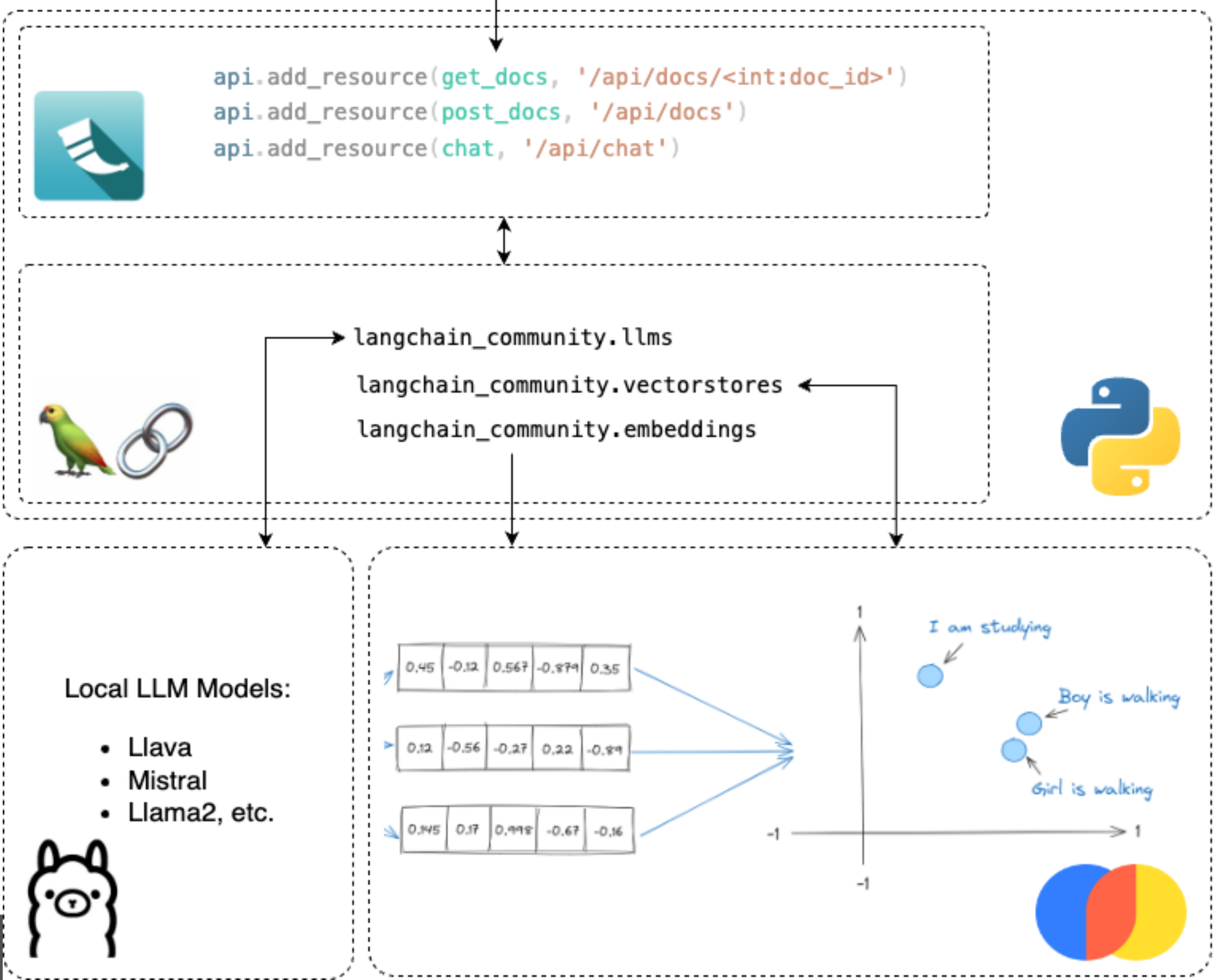

Document/URL
Embed
Pick Document/Web Site URL(s) to chat with
Ask your documents a question
Enter your message...
ngrok

POST /llm/rag/chat

POST /llm/rag/add_source

http://xxxx-yyyy-zzzz..ngrok-free.app

http://127.0.0.1:5000/



```
api.add_resource(get_docs, '/api/docs/<int:doc_id>')
api.add_resource(post_docs, '/api/docs')
api.add_resource(chat, '/api/chat')
```

```
langchain_community.llms
langchain_community.vectorstores
langchain_community.embeddings
```

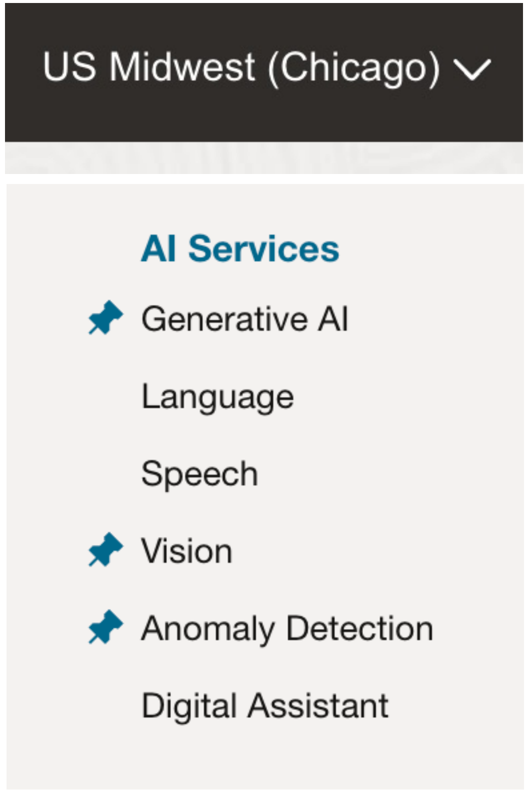
Local LLM Models:

- Llava
- Mistral
- Llama2, etc.

Oracle in the AI era

OCI Generative AI Service

now



Models:

cohere

Cohere Command 52b and 6b parameter sizes (the XL and light models)

Llama 2

Meta Llama 2 70b parameter model

Embeddings:

cohere

Embed English and English Light V3
Embed Multilingual and Multilingual Light V3

Features:

- Text generation
- Text summarization
- Text embeddings

- Fine tuning custom models
- Management, Inference and Agents* APIs
- LangChain integration
- Playground for trying out models and parameters

OCI Generative AI Service

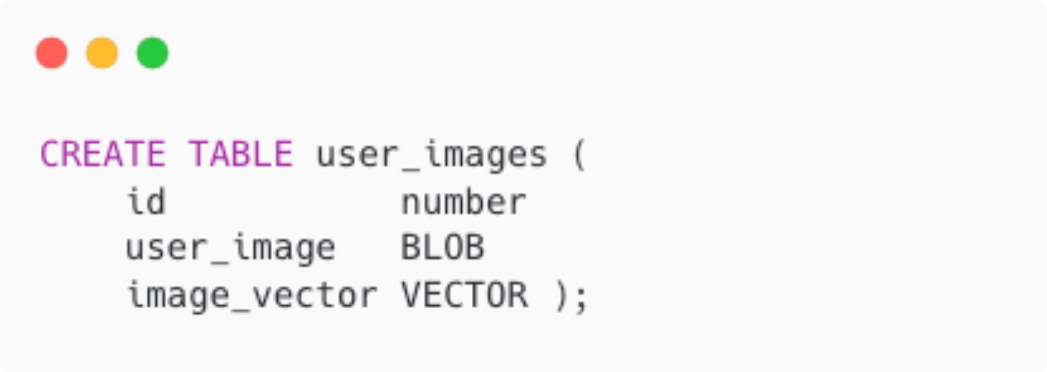
Coming in 23.4

- AI Vector Search, including:
  - New vector data type
  - Vector indexes
  - Vector search SQL operators that enable the Oracle Database to store the semantic content of documents, images, and other unstructured data as vectors, and use these to run fast similarity queries.

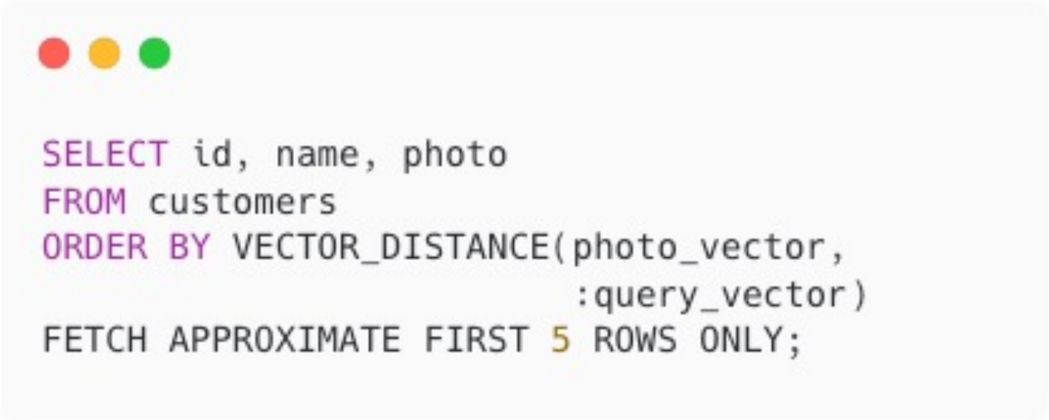
- LlamaIndex support
- Support for defining vector types in PLSQL
- ONNX Runtime in the Oracle database
  - This allows you to run ML models in the database
  - Enables the SQL embed() function to create vectors in the database

OCI Generative AI Service

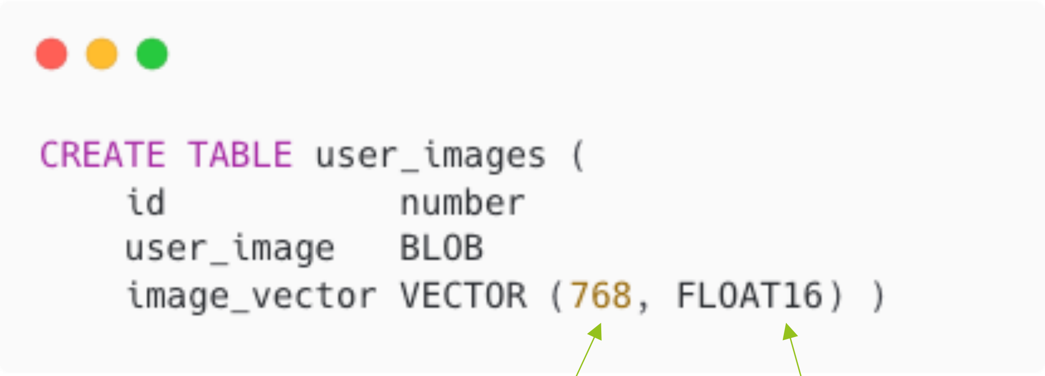
Coming in 23.4



```
CREATE TABLE user_images (
    id          number
    user_image  BLOB
    image_vector VECTOR );
```

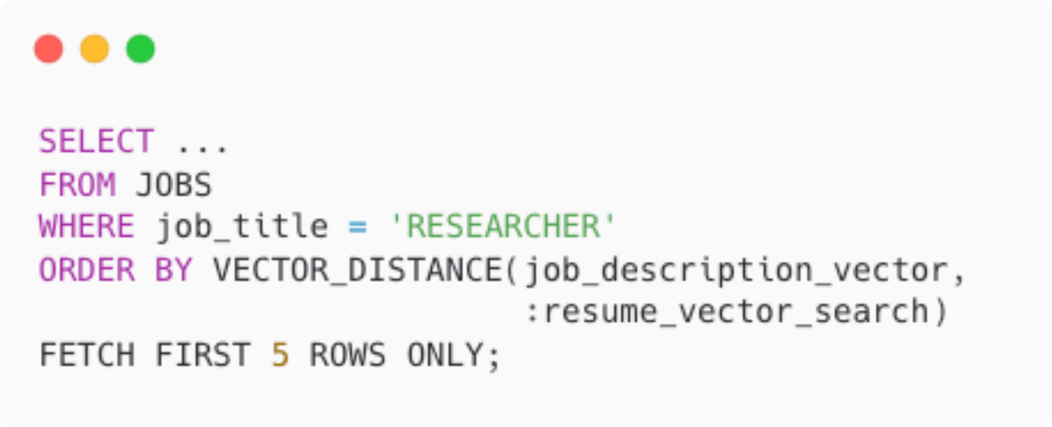


```
SELECT id, name, photo
FROM customers
ORDER BY VECTOR_DISTANCE(photo_vector,
                         :query_vector)
FETCH APPROXIMATE FIRST 5 ROWS ONLY;
```



```
CREATE TABLE user_images (
    id          number
    user_image  BLOB
    image_vector VECTOR (768, FLOAT16) )
```

# of dimensions **format**



```
SELECT ...
FROM JOBS
WHERE job_title = 'RESEARCHER'
ORDER BY VECTOR_DISTANCE(job_description_vector,
                         :resume_vector_search)
FETCH FIRST 5 ROWS ONLY;
```

Questions?