

**Improve search system**  
**Keyword search**  
**Semantic search**  
**Hybrid search**





Page

Messages

Notifications 3

Insights

Publishing Tools

Settings

Help ▾



somkiat.cc

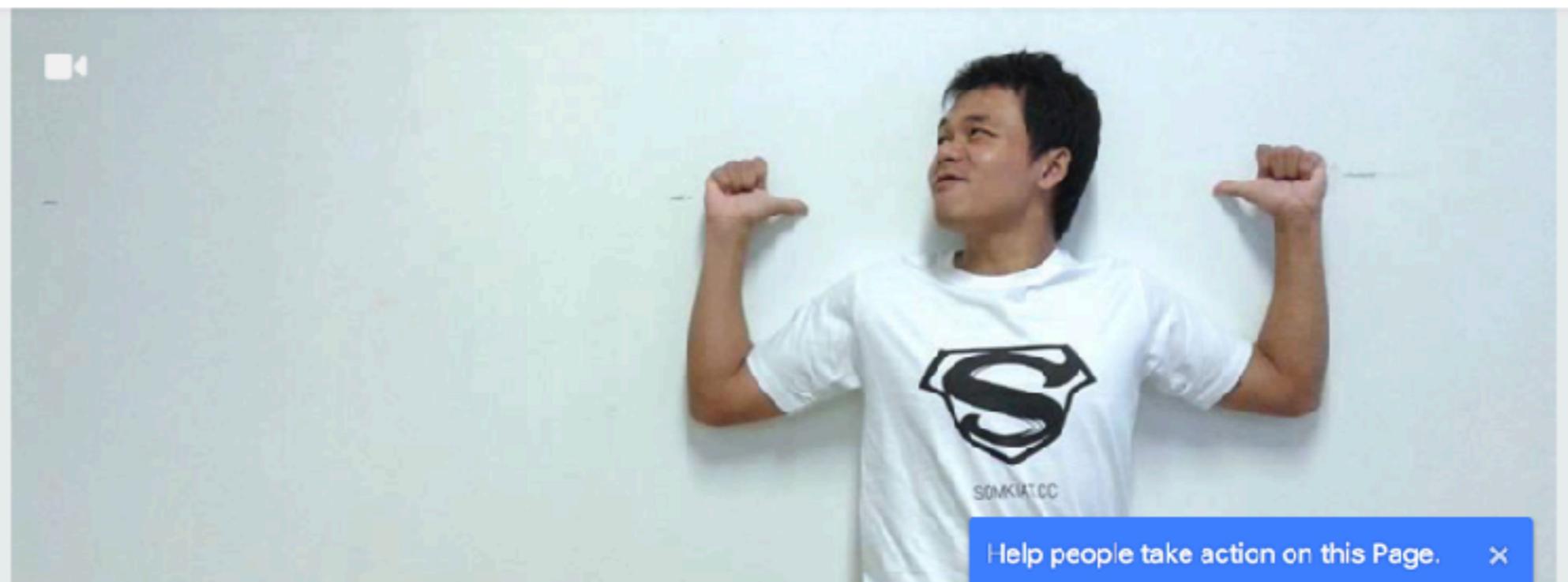
@somkiat.cc

Home

Posts

Videos

Photos



Liked

Following

Share

...

Help people take action on this Page. 

+ Add a Button



Workshop

3

© 2020 - 2026 Siam Chamnankit Company Limited. All rights reserved.

**[https://github.com/up1/  
workshop-semantic-search](https://github.com/up1/workshop-semantic-search)**



# Topics

Search types

Keyword vs Semantic vs Hybrid search

Vectorization and Embedding

Improve your search system

Workshop



# **Retrieval-Augmented Generation (RAG)**



# RAG

Enhances LLMs by retrieving external knowledge before generating response

Improve accuracy

Reduce hallucinations

Real-time knowledge updates



# Core Components

## Retriever

Fetch relevant documents from a knowledge base

## Generator

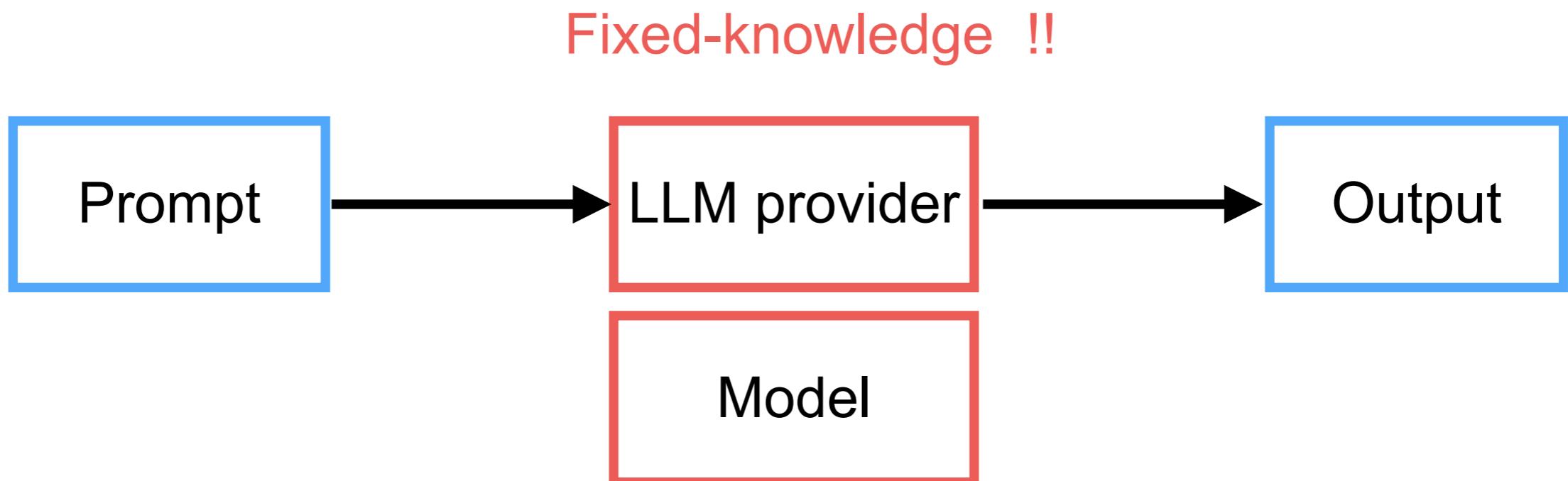
Uses the retrieved information to generate a response

## Enhancements

Different RAG architectures modify how retrieval and generation interact to improve performance

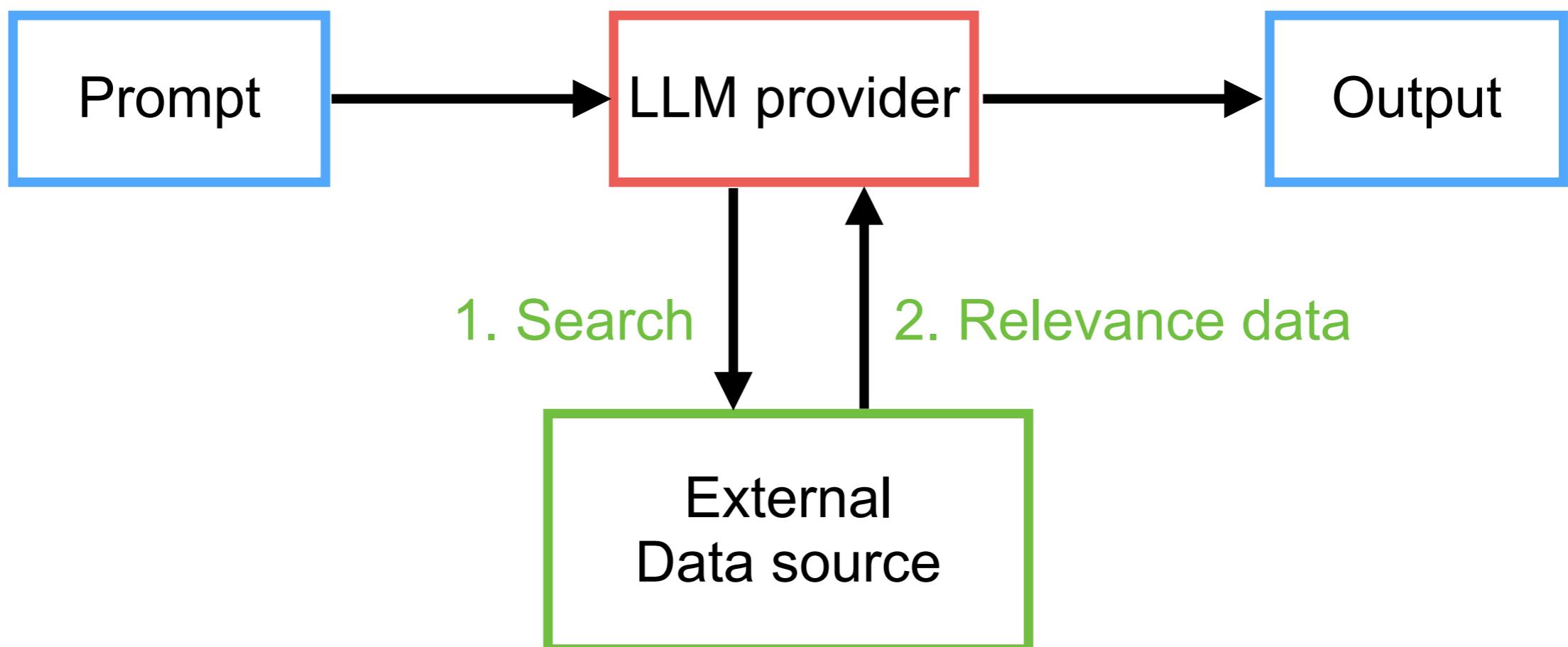


# Limitation of LLM



# RAG ?

Fixed-knowledge !!



# How to search/retrieve data ?

Data source

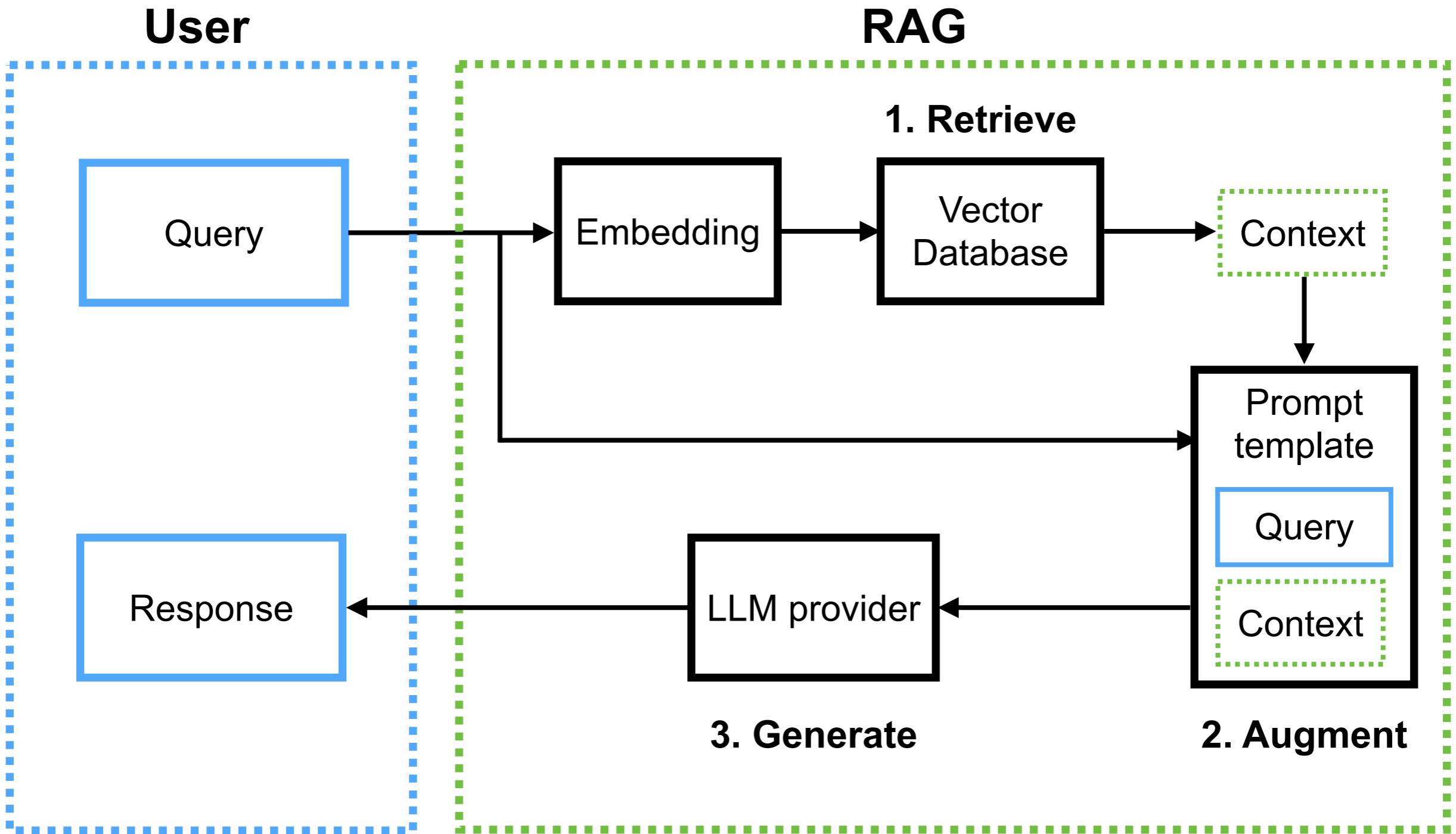
Data preparation

Search

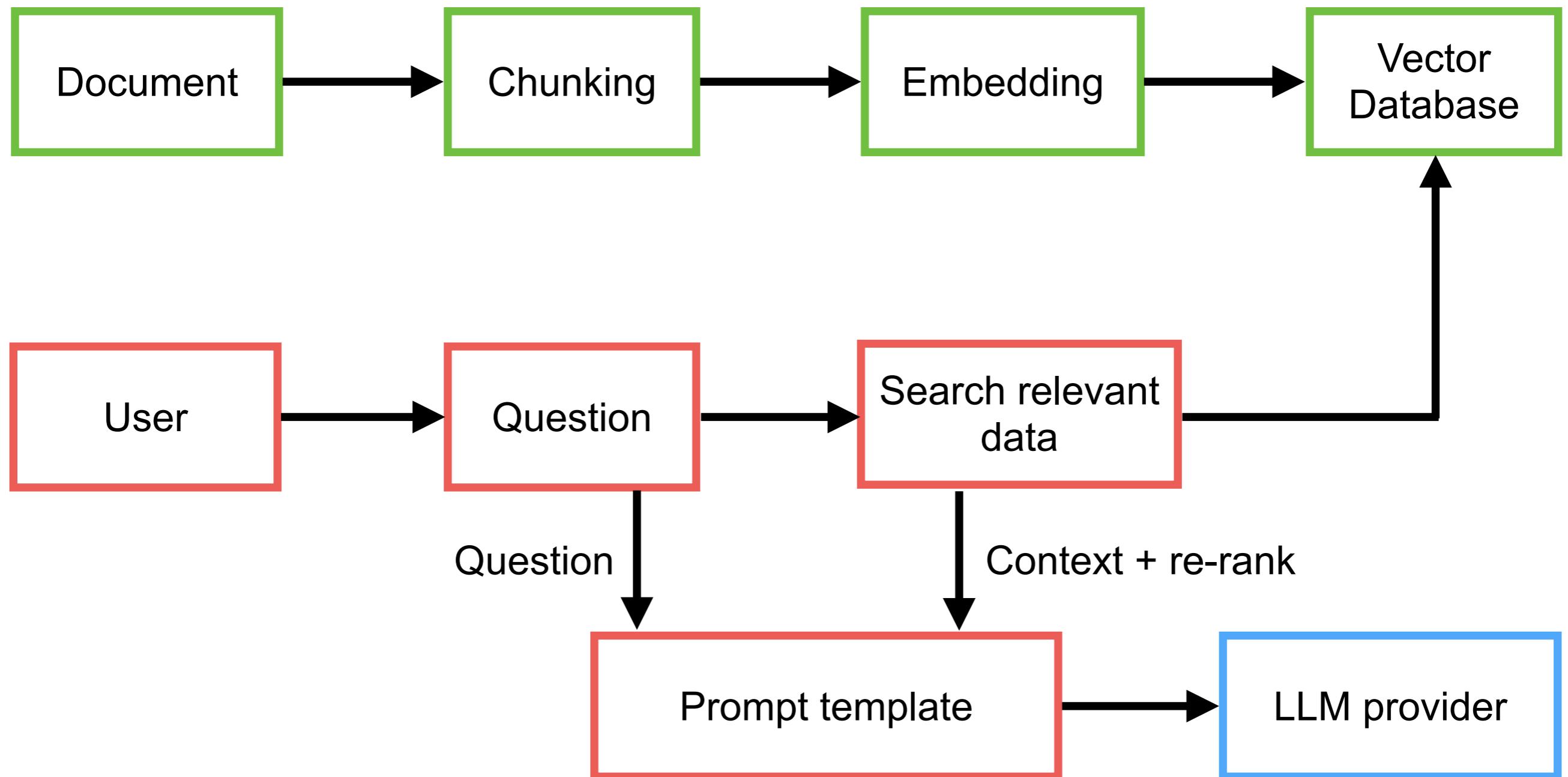
Search result



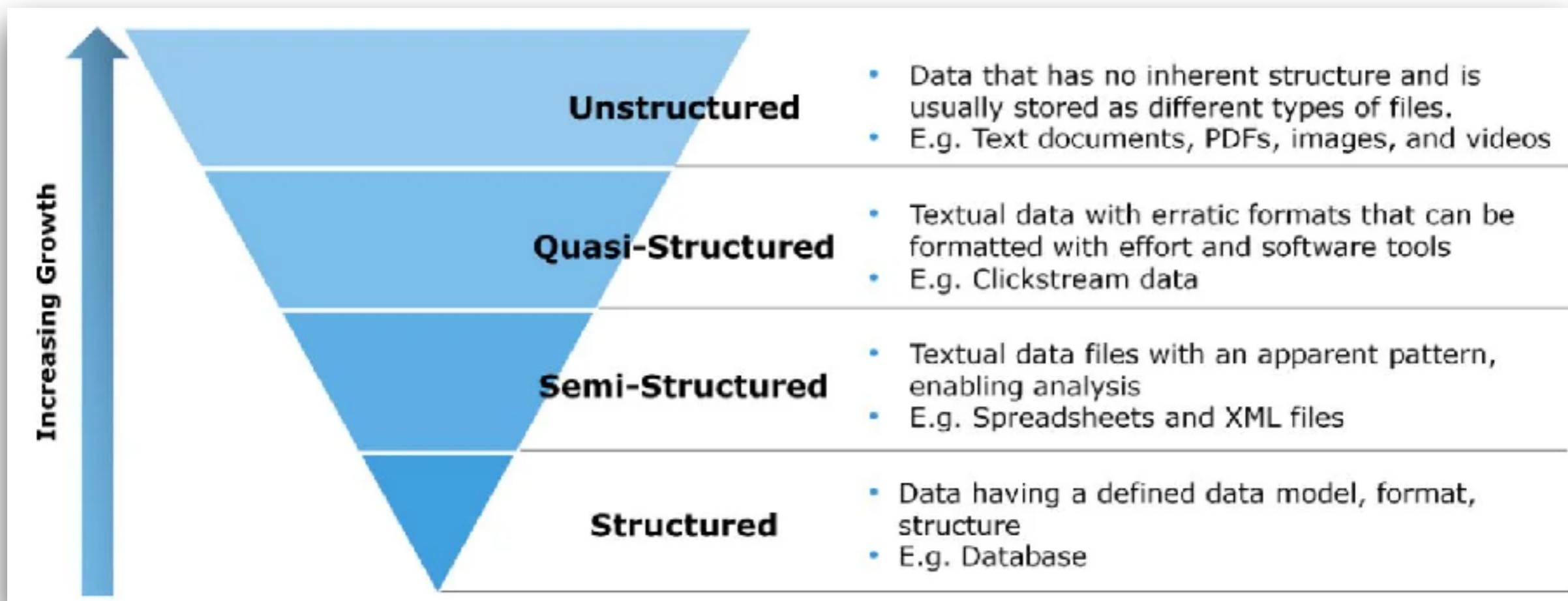
# Basic RAG Architecture



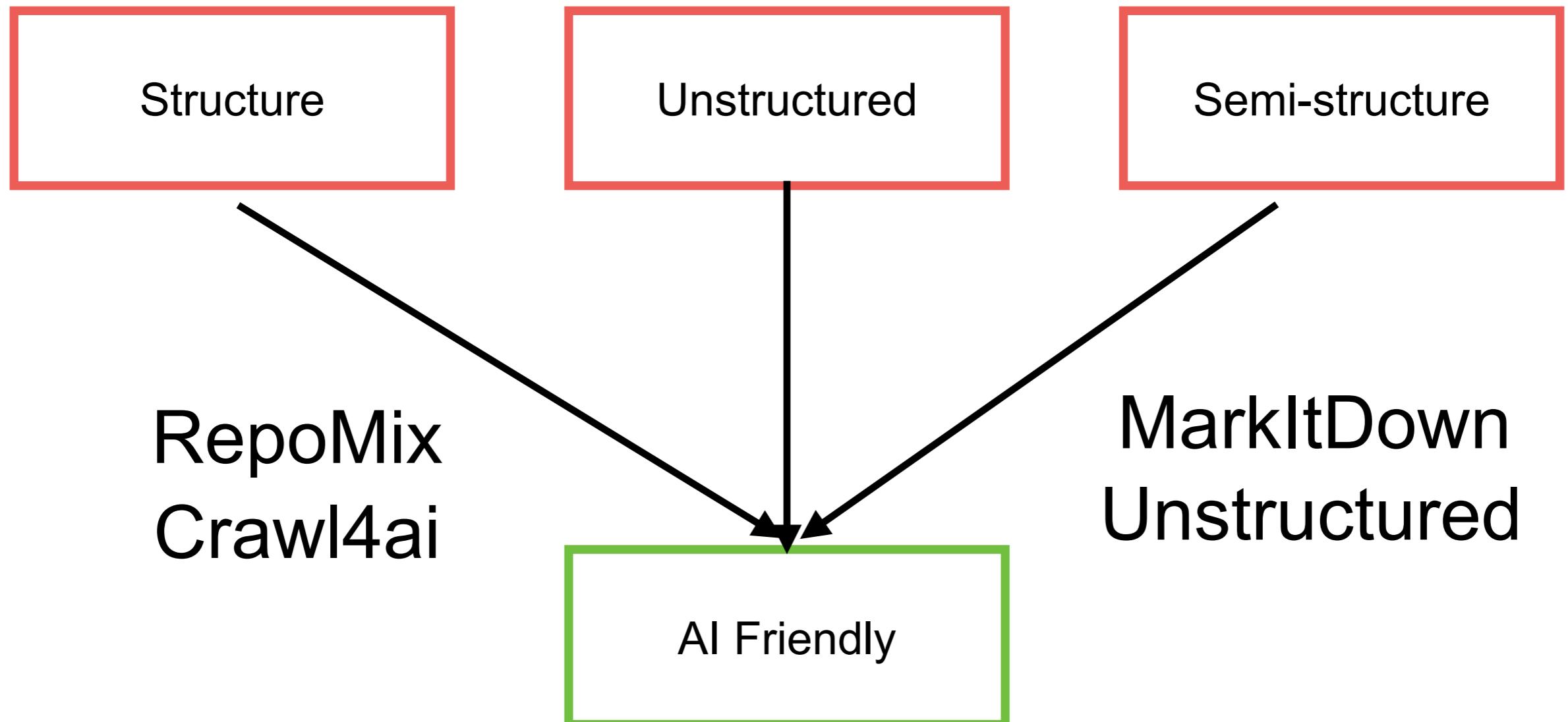
# RAG process



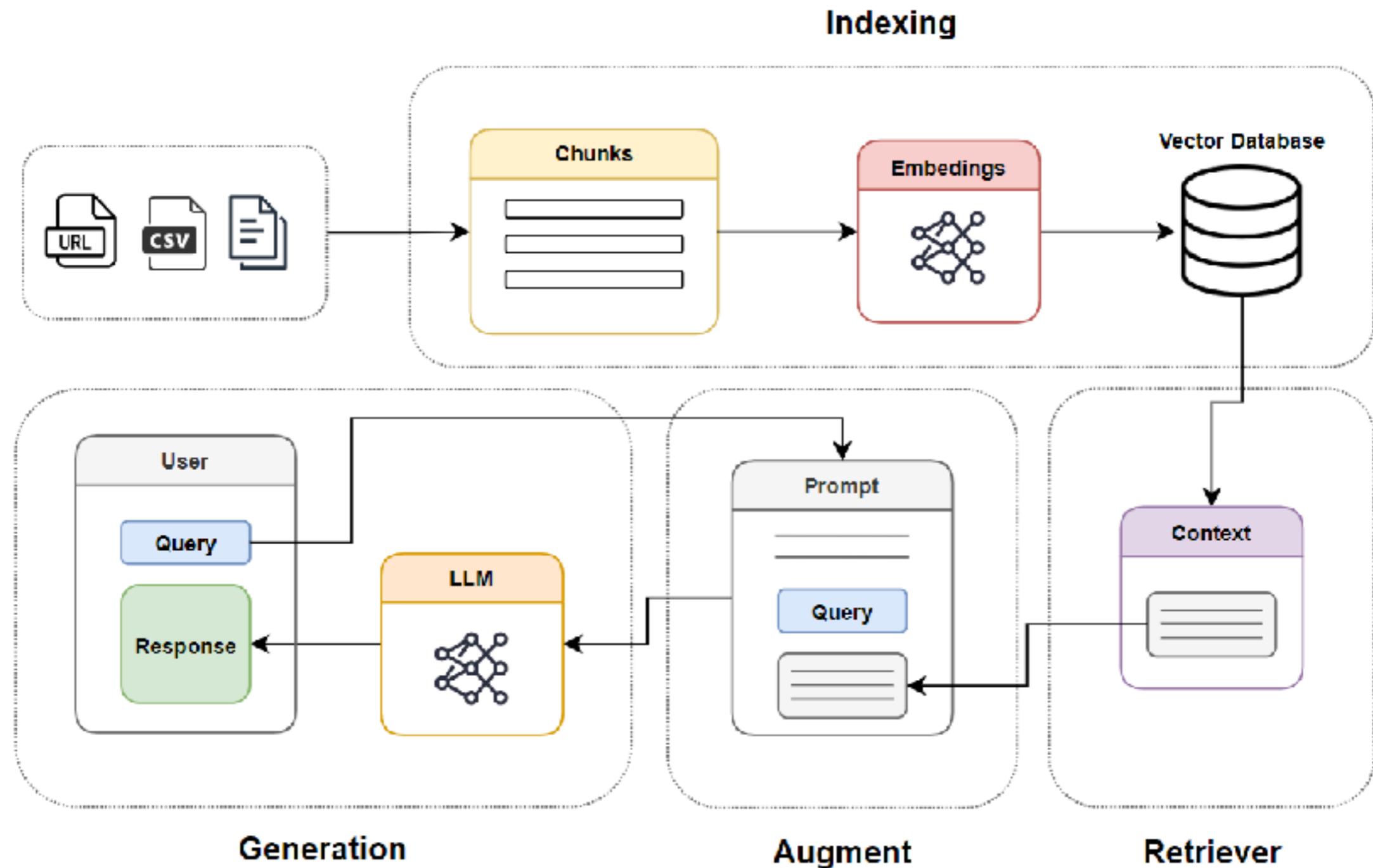
# Structures of Data ?



# Friendly Data for LLM/AI



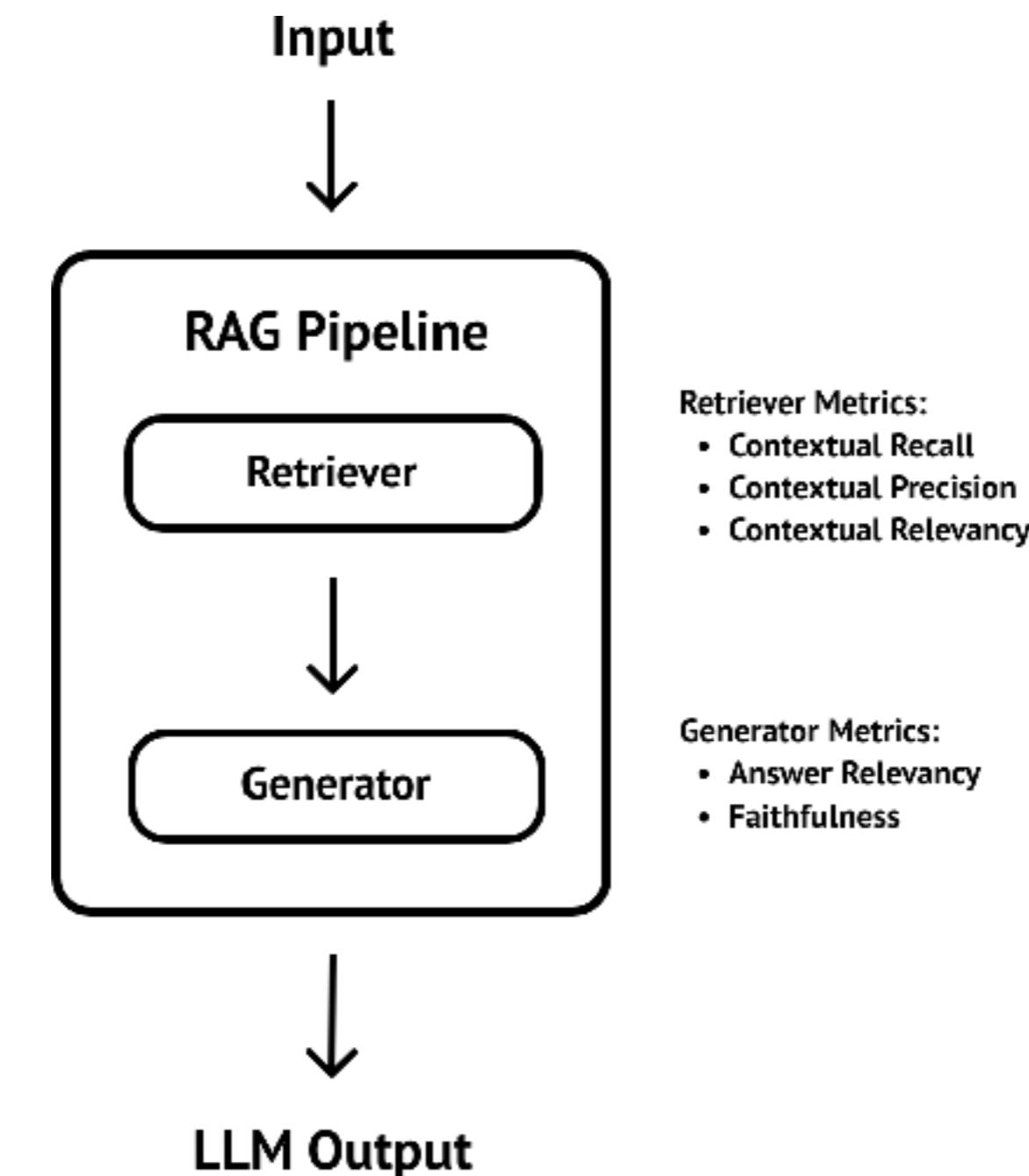
# RAG Cookbook (techniques)



<https://github.com/athina-ai/rag-cookbooks>



# RAG Evaluation !!



#### Retriever Metrics:

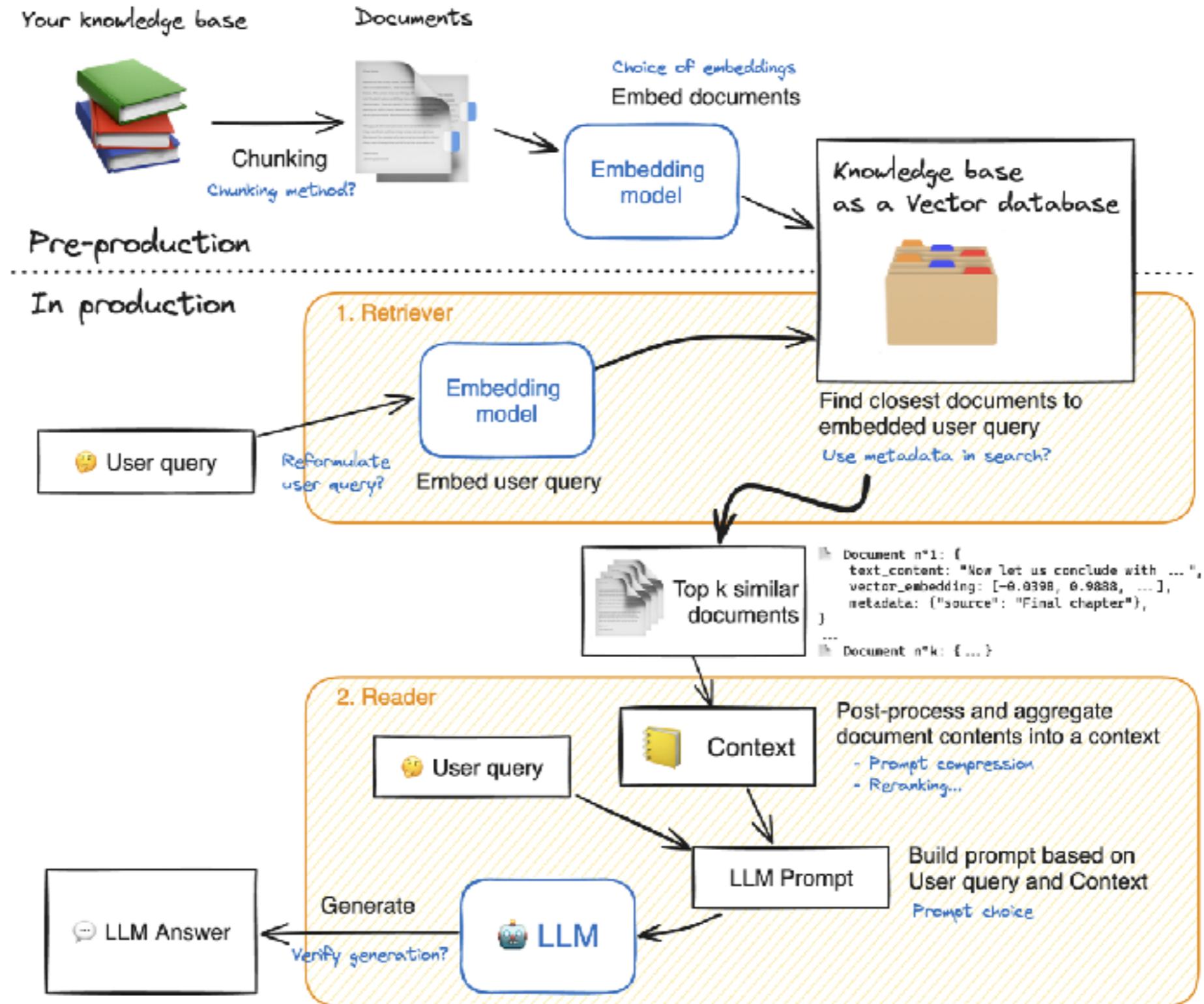
- Contextual Recall
- Contextual Precision
- Contextual Relevancy

#### Generator Metrics:

- Answer Relevancy
- Faithfulness

<https://www.deepeval.com/guides/guides-rag-evaluation>

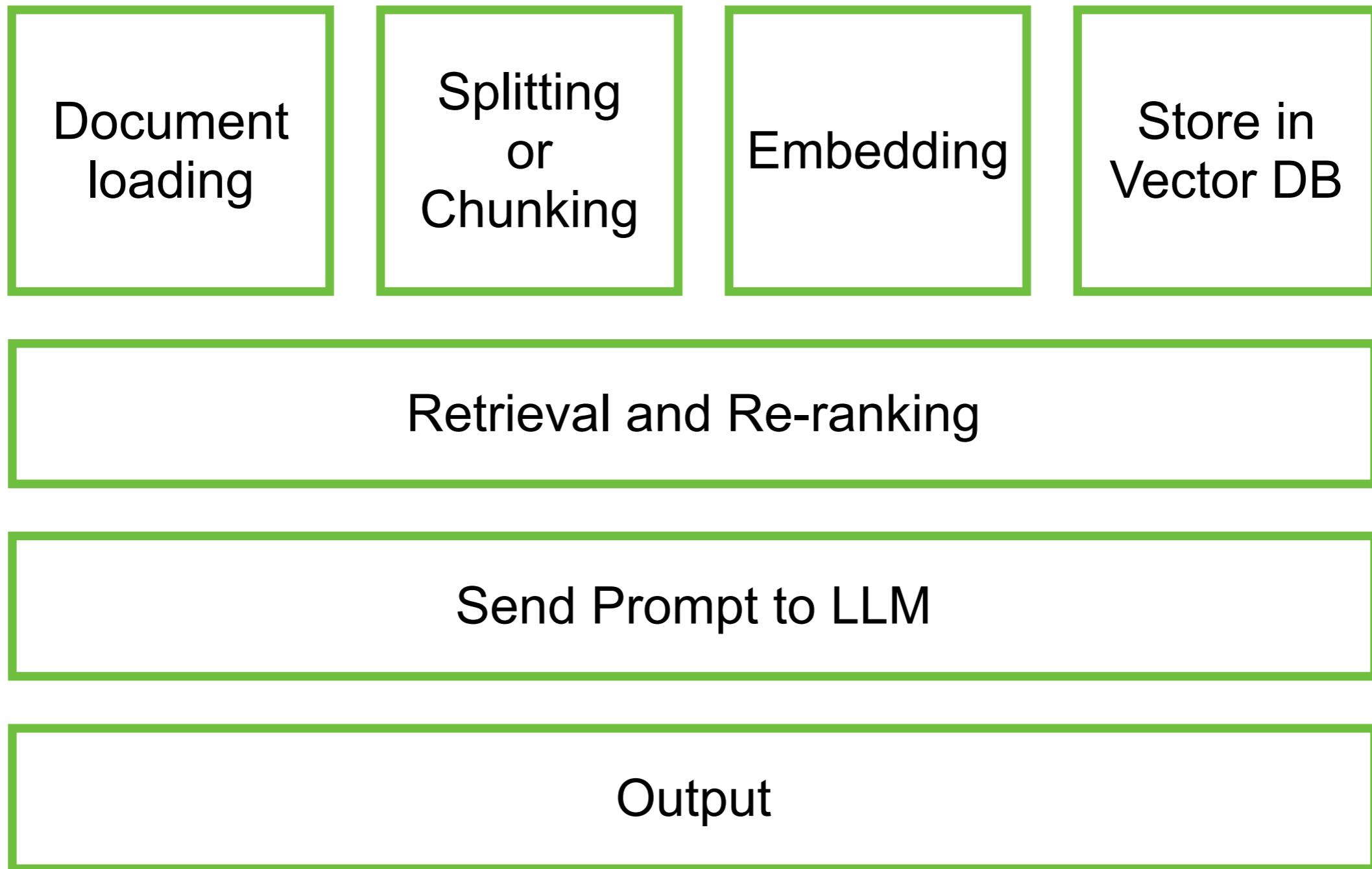




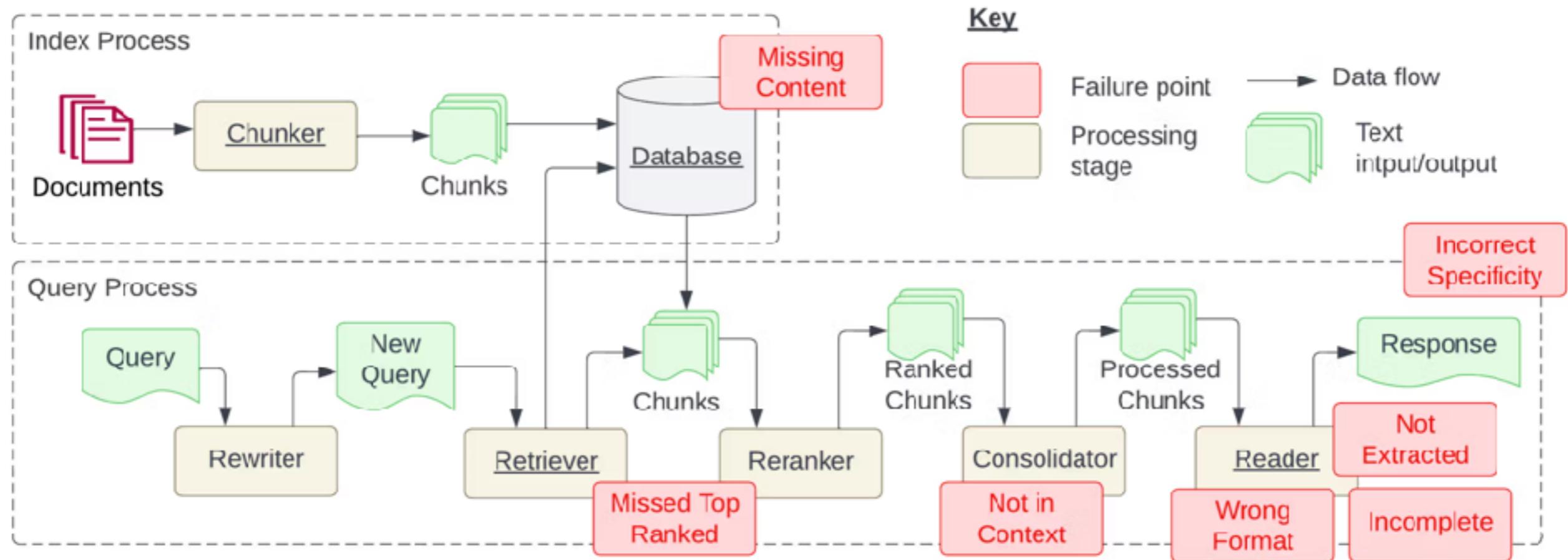
[https://huggingface.co/learn/cookbook/en/rag\\_evaluation](https://huggingface.co/learn/cookbook/en/rag_evaluation)



# RAG Implementation



# Failure Points of RAG



**Figure 1: Indexing and Query processes required for creating a Retrieval Augmented Generation (RAG) system. The indexing process is typically done at development time and queries at runtime. Failure points identified in this study are shown in red boxes. All required stages are underlined. Figure expanded from [19].**

<https://www.galileo.ai/blog/mastering-rag-how-to-architect-an-enterprise-rag-system>



# RAG is better

**But come with Cost !!**

More latency

Retrieval errors

Accuracy !!

More  
Complexity

Maintenance  
overhead



# RAG Techniques ?

Semantic  
chunking

Chunk size  
selector

Context chunk  
header

Adaptive RAG

Re-ranking

Graph TAG

<https://github.com/FareedKhan-dev/all-rag-techniques>



# Pre-Retrieval optimization ?

Preprocessing and cleansing data

Chunking strategies

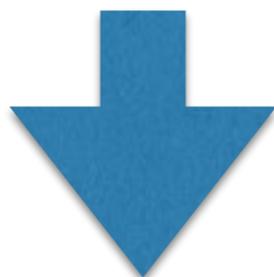
Add metadata

Embedding model selection

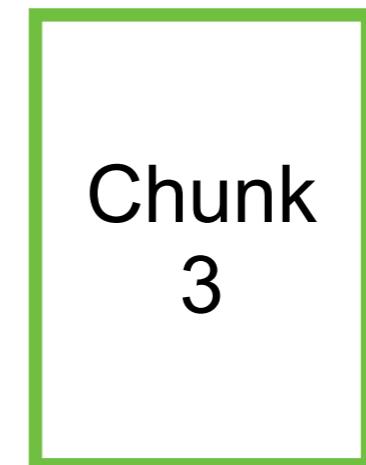
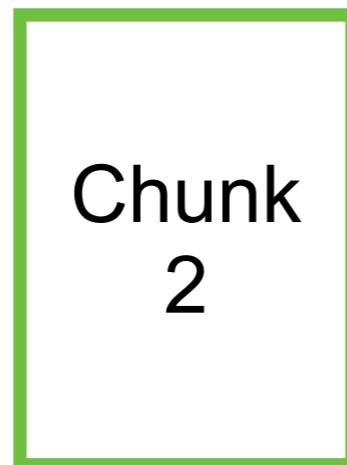
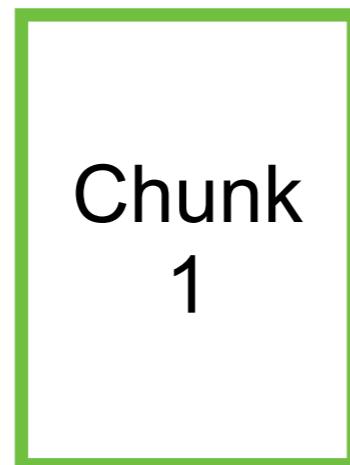


# Chunking strategies

Size of Document  
**> context window size**

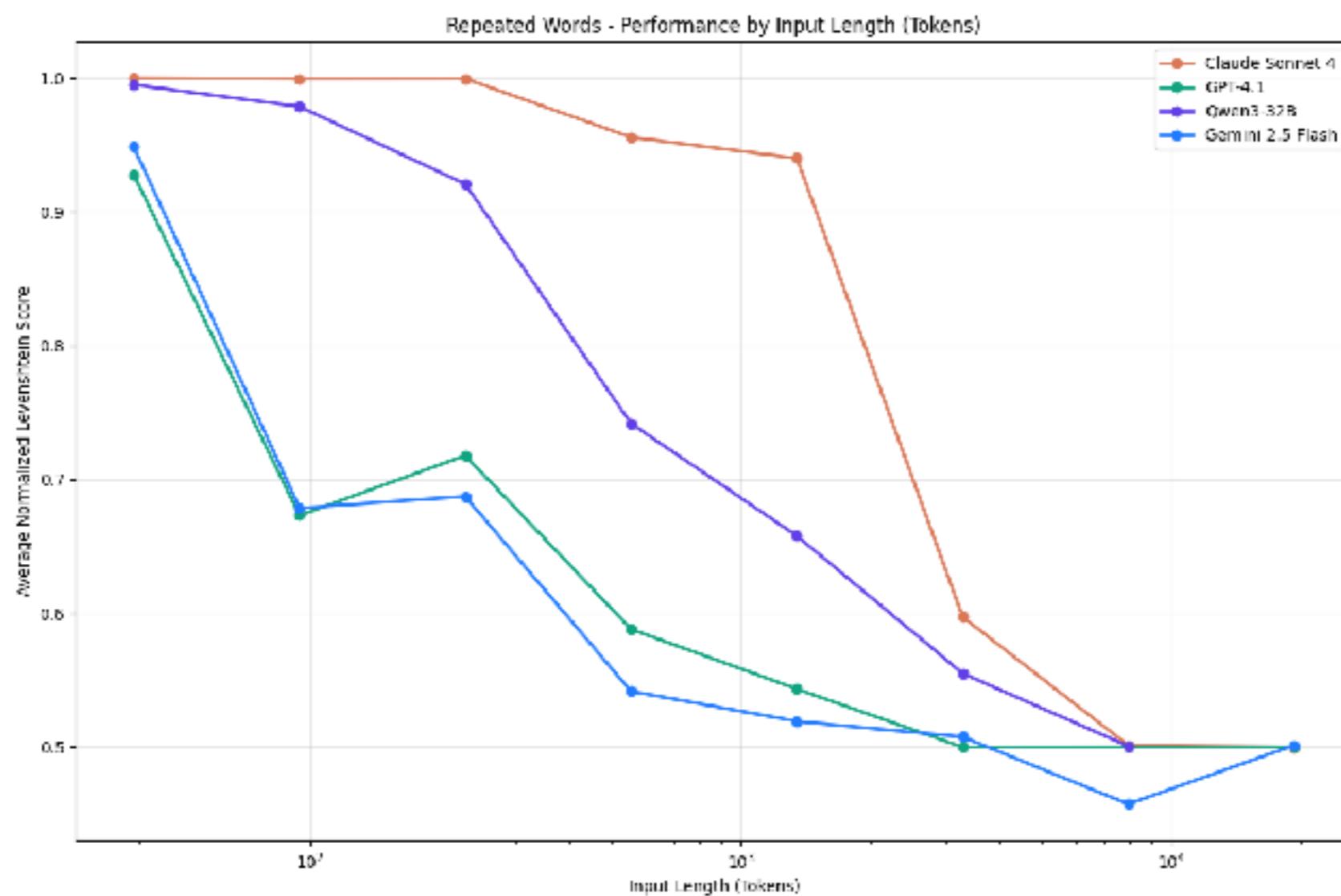


Chunking



# Context Window ?

Number of tokens in the context window increases,  
the model's ability to accurately recall information from that context decreases



<https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents>



# Chunking Strategies !!

Fixed size  
Recursive characters  
Document structure-based  
Semantic chunking  
Agentic chunking

...

<https://blog.dailydoseofds.com/p/5-chunking-strategies-for-rag>



# Good Chunking

Efficiency

Relevance

Context  
preservation

Improve content  
generation

Reduce noise



# Bad Chunking

Loss context

Redundancy

Inconsistency



# Chunking Visualization

## ChunkViz v0.1

Want to learn more about AI Engineering Patterns? Join me on [Twitter](#) or [Newsletter](#).

Language Models do better when they're focused.

One strategy is to pass a relevant subset (chunk) of your full data. There are many ways to chunk text.

This is an tool to understand different chunking/splitting strategies.

[Explain like I'm 5...](#)

One of the most important things I didn't understand about the world when I was a child is the degree to which the returns for performance are superlinear.

Teachers and coaches implicitly told us the returns were linear. "You get out," I heard a thousand times, "what you put in." They meant well, but this is rarely true. If your product is only half as good as your competitor's, you don't get half as many customers. You get no customers, and you go out of business.

It's obviously true that the returns for performance are superlinear in business. Some think this is a flaw of capitalism, and that if we changed the rules it would stop being true. But superlinear returns for performance are a feature of the world, not an artifact of rules we've invented. We see the same pattern in fame, power, military victories, knowledge, and ...

[Upload txt](#)

Splitter:  

Chunk Size:  

Chunk Overlap:  

Total Characters: 2658  
Number of chunks: 107  
Average chunk size: 24.8

One of the most important things I didn't understand about the world when I was a child is the degree to which the returns for performance are superlinear.

Teachers and coaches implicitly told us the returns were linear. "You get out," I heard a thousand times, "what you put in." They meant well, but this is rarely true. If your product is only half as good as your competitor's, you don't get half as many customers. You get no customers, and you go out of business.

<https://chunkviz.up.railway.app/>



# Retrieval optimization ?

Re-ranking

Hybrid search

Query  
transformation

Multi-vector  
embedding

Contextual  
retrieval

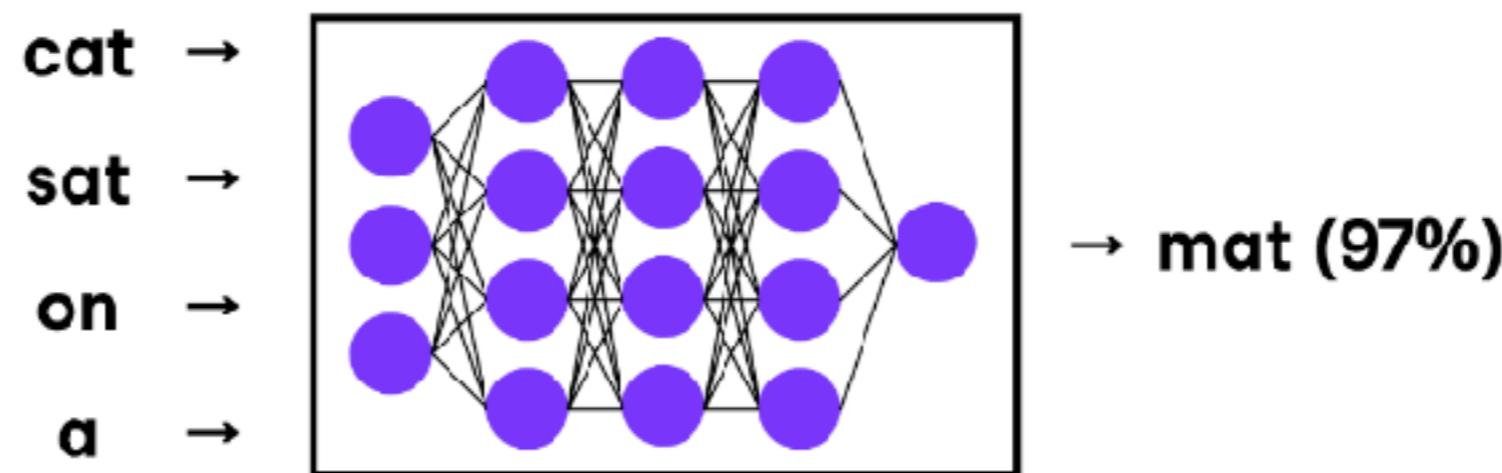
Vector database  
Selection



# Large Language Model (LLM)

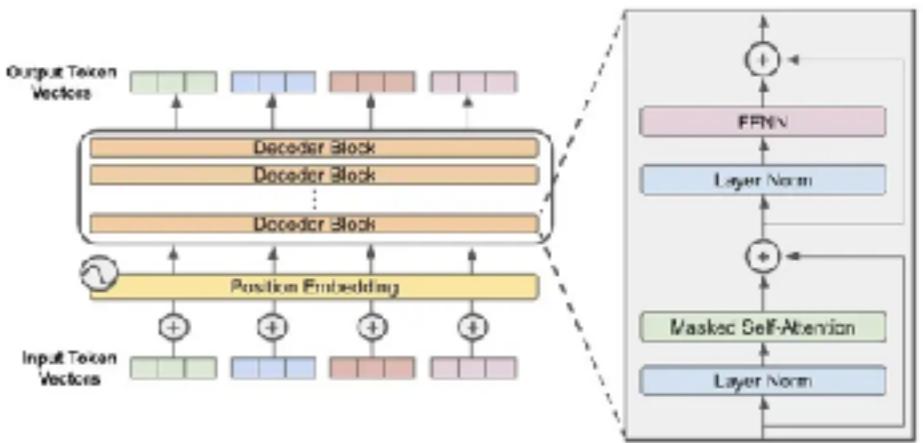
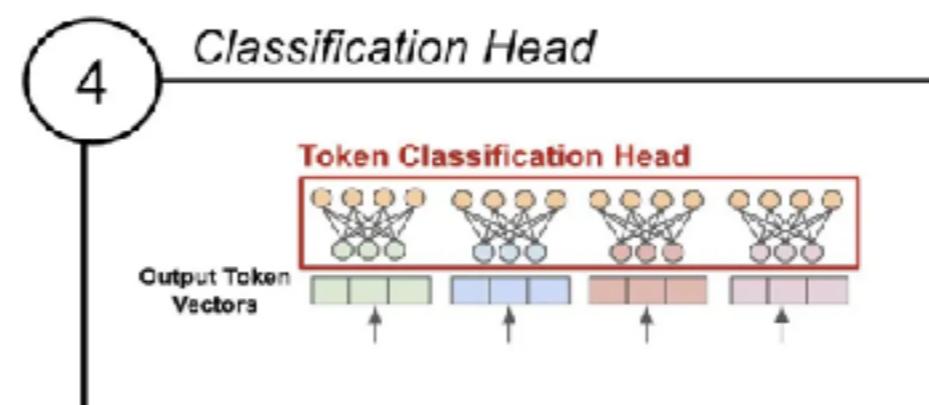
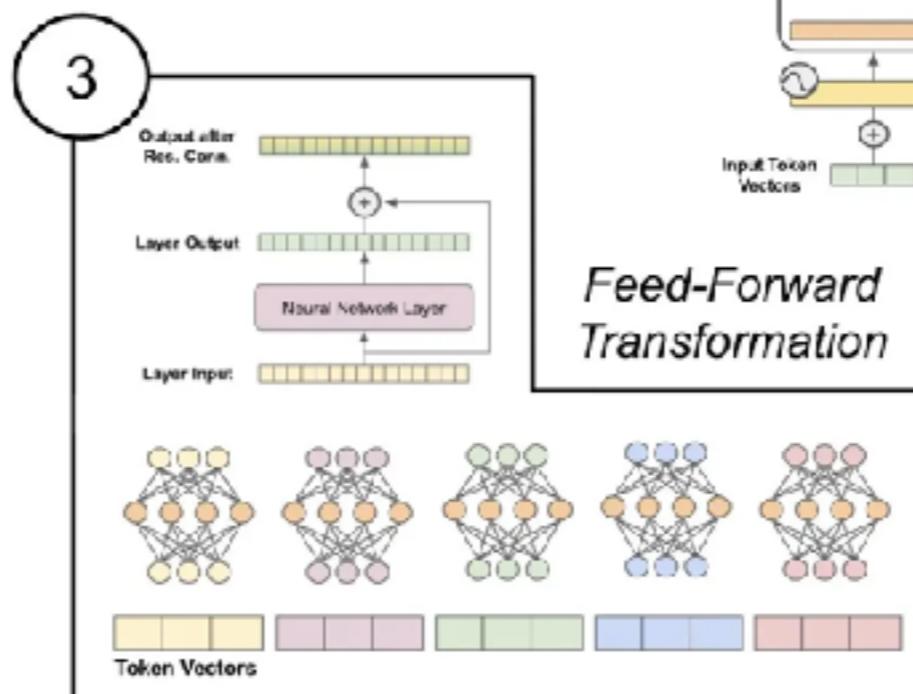
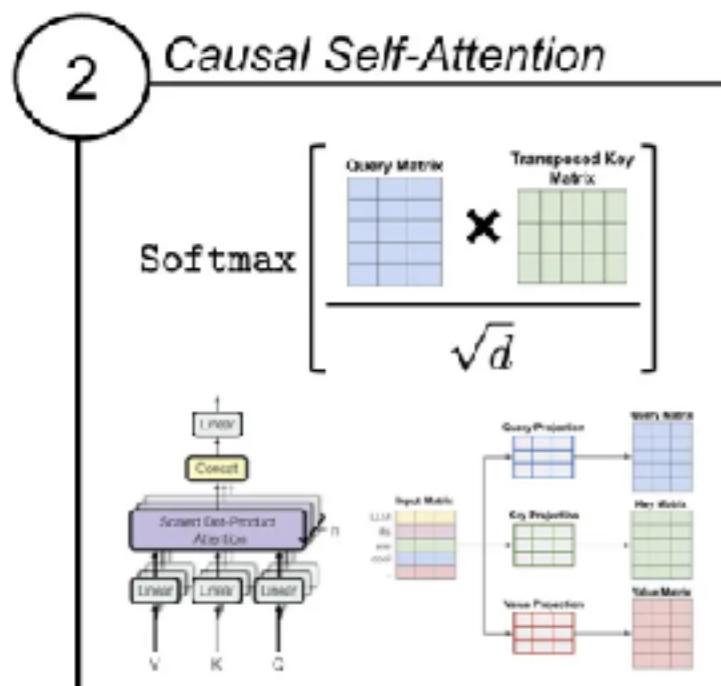
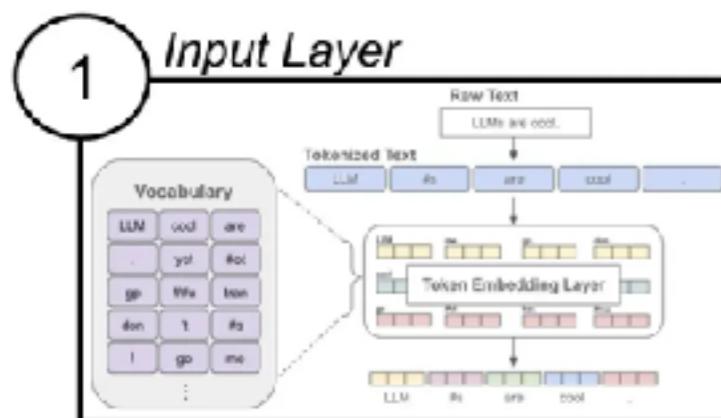
Neural network

Predicts the next word in a sequence



# LLM components !!

## Components of the Decoder-only Transformer



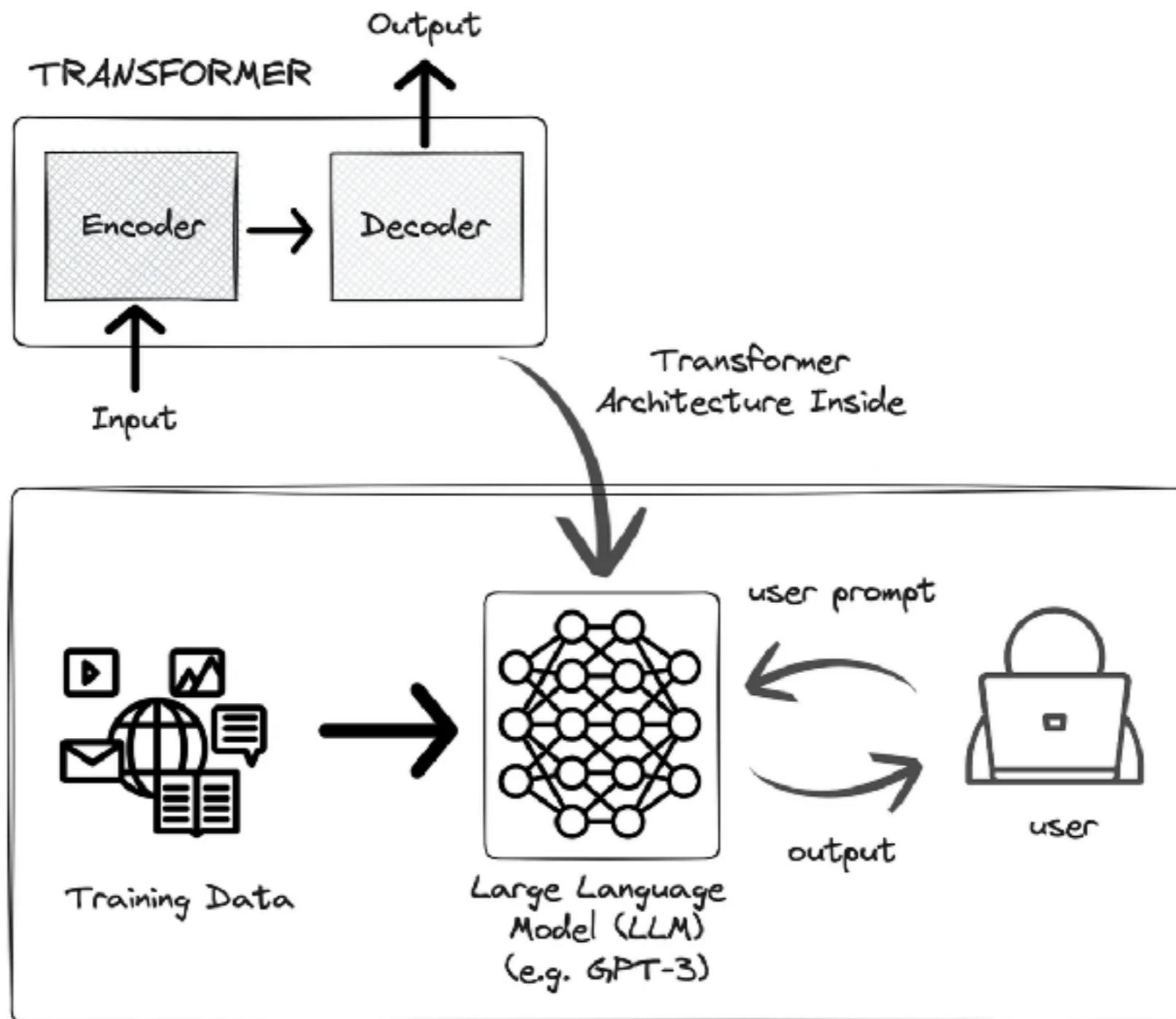
```
from torch import nn
class Block(nn.Module):
    def __init__(self, d, h, T, base, dropout=0.2, **kwargs):
        super().__init__()
        self.ln_1 = nn.LayerNorm(d)
        self.attn = CausalSelfAttention(d, h, T, base, dropout)
        self.ln_2 = nn.LayerNorm(d)
        self.ffnn = FFNN(d, base, dropout)

    def forward(self, x):
        x = x + self.attn(self.ln_1(x))
        x = x + self.ffnn(self.ln_2(x))
        return x
```

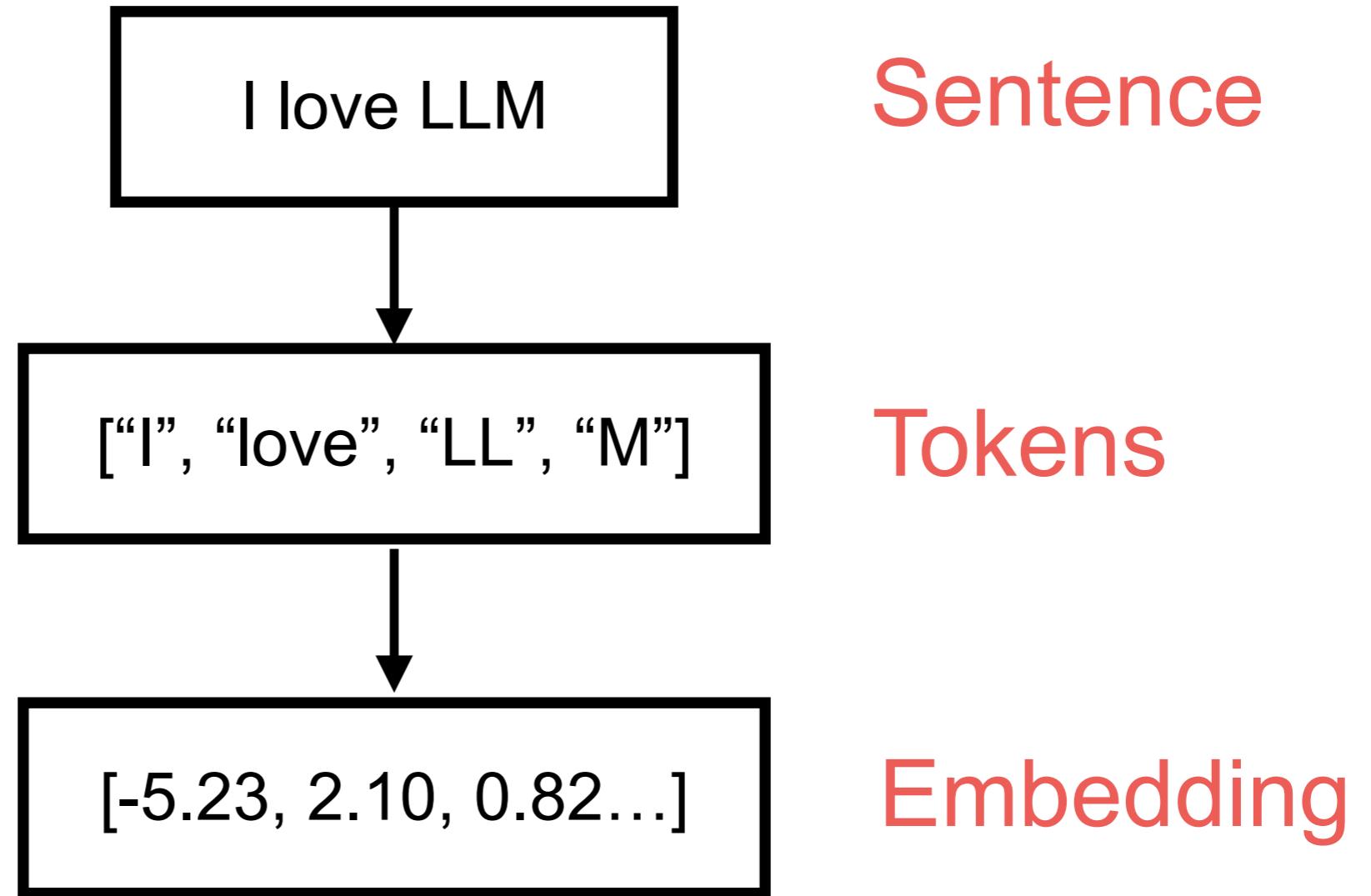
<https://stackoverflow.blog/2024/08/22/lms-evolve-quickly-their-underlying-architecture-not-so-much>



# Transformer inside

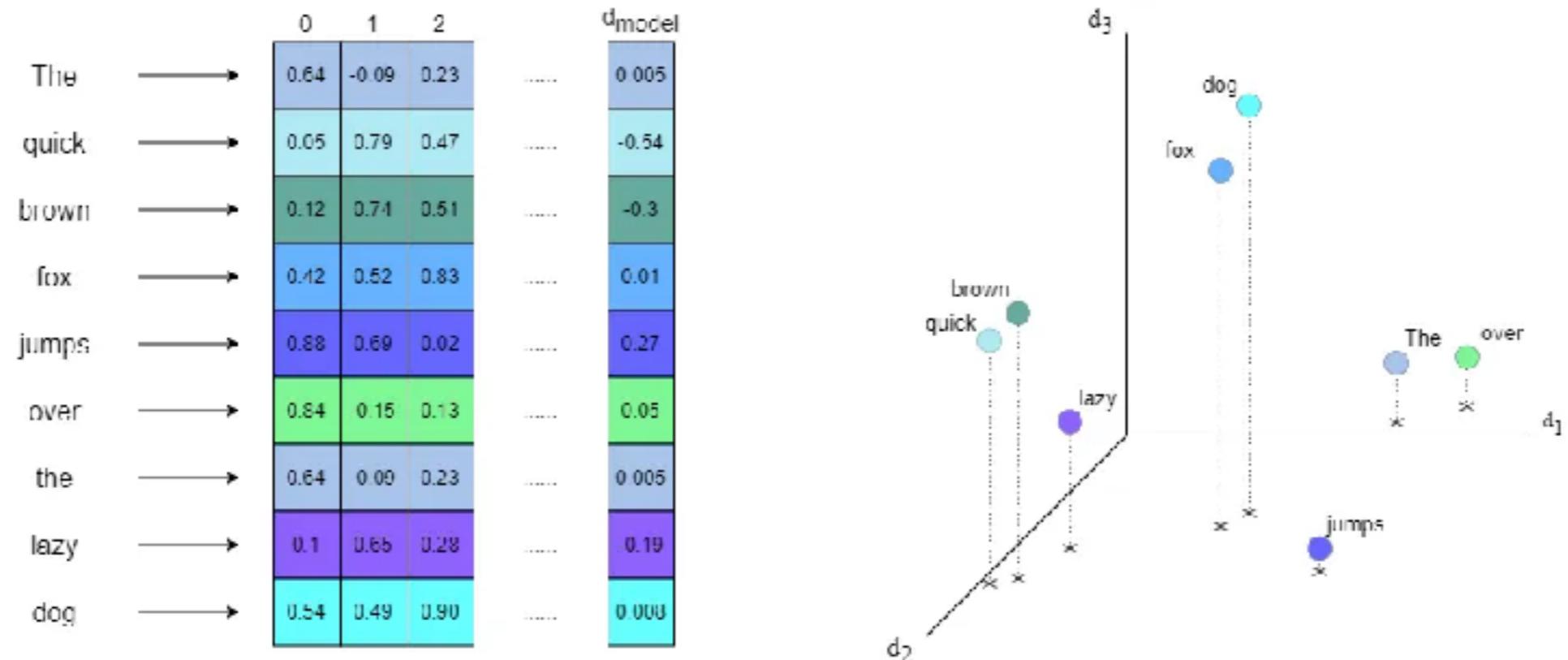


# Transformer process



# Embedding ?

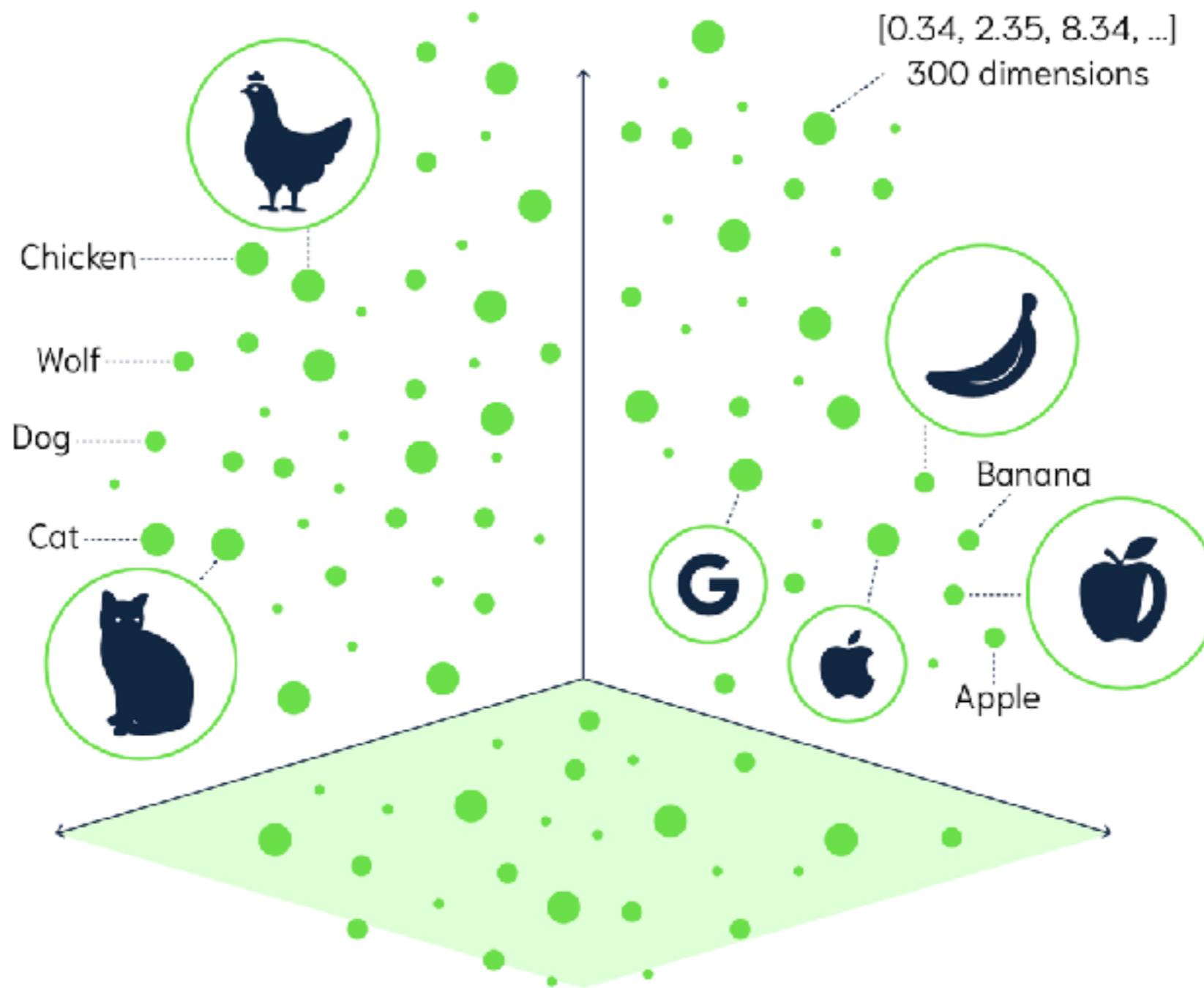
Map items of unstructured data to high-dimensional real vectors



<https://towardsdatascience.com/transfomers-in-depth-part-1-introduction-to-transformer-models-in-5-minutes-ad25da6d3cca>



# Visual of Vector space



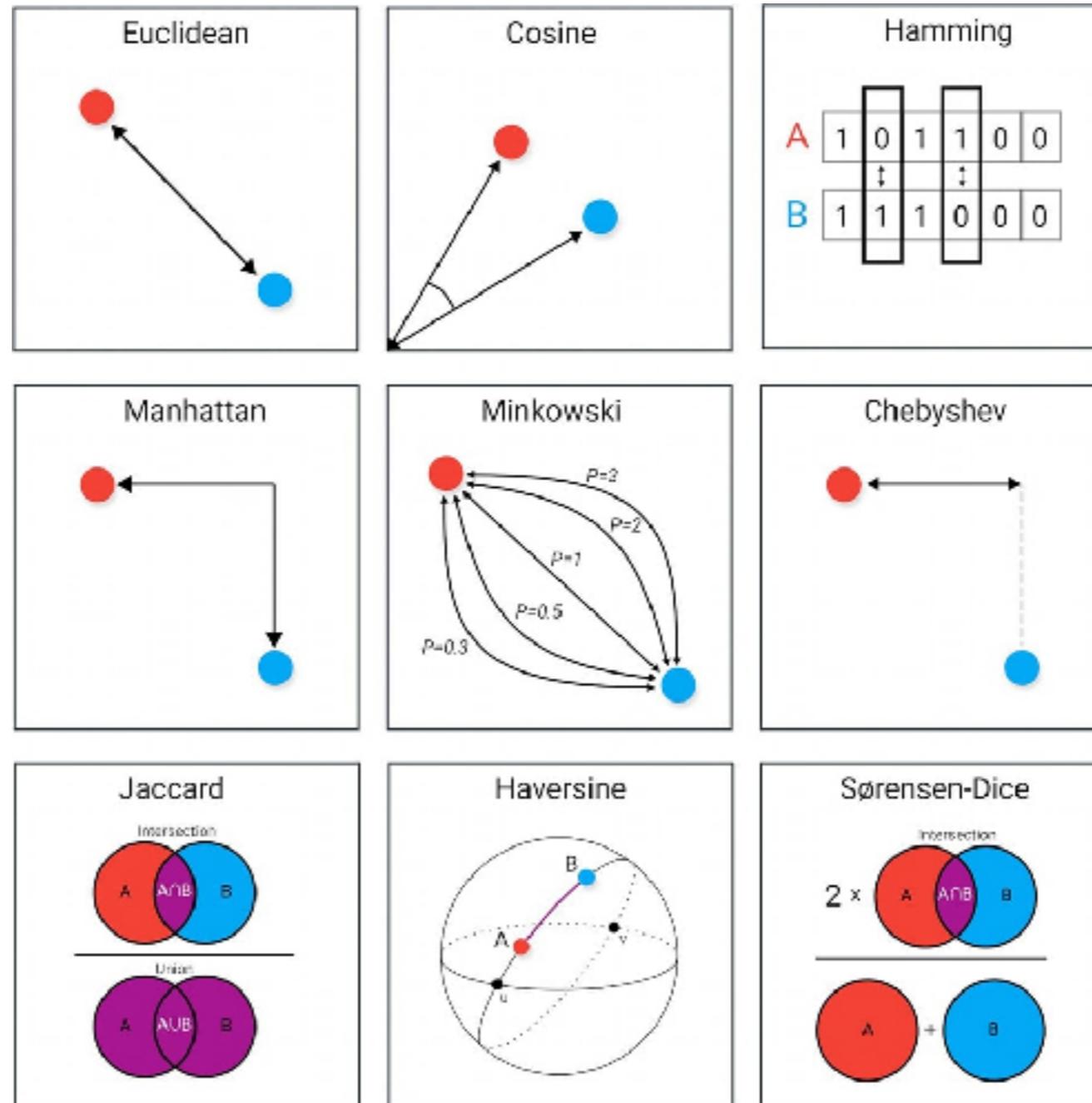
# Embedding Leaderboard

Rank (Box...)	Model	Zero-shot	Memory U...	Number of P...	Embedding D...	Max Tokens	Mean (T...	Mean (TaskT...	Bitext ...	Classific...
1	<a href="#">llama-embed-nemotron-Bb</a>	99%	28629	7B	4096	32768	69.45	61.09	<b>81.72</b>	73.21
2	<a href="#">Qwen3-Embedding-8B</a>	99%	28866	7B	4096	32768	<b>70.58</b>	<b>61.69</b>	80.89	<b>74.00</b>
3	<a href="#">gemini-embedding-001</a>	99%	Unknown	Unknown	3072	2048	68.37	59.59	79.28	71.82
4	<a href="#">Qwen3-Embedding-4B</a>	99%	15341	4B	2560	32768	69.45	60.86	79.36	72.33
5	<a href="#">Qwen3-Embedding-0.6B</a>	99%	2272	595M	1024	32768	64.34	56.01	72.23	66.83
6	<a href="#">gte-Qwen2-7B-instruct</a>	⚠ NA	29040	7B	3584	32768	62.51	55.93	73.92	61.55
7	<a href="#">Ling-Embed-Mistral</a>	99%	13563	7B	4096	32768	61.47	54.14	70.34	62.24
8	<a href="#">multilingual-e5-large-instruct</a>	99%	1068	560M	1024	514	63.22	55.08	80.13	64.94
9	<a href="#">embeddinggemma-300m</a>	99%	578	367M	768	2048	61.15	54.31	64.40	60.90
10	<a href="#">SFR-Embedding-Mistral</a>	96%	13563	7B	4096	32768	60.99	53.92	70.00	60.02
11	<a href="#">text-multilingual-embedding-002</a>	99%	Unknown	Unknown	768	2048	62.15	54.25	70.73	64.64

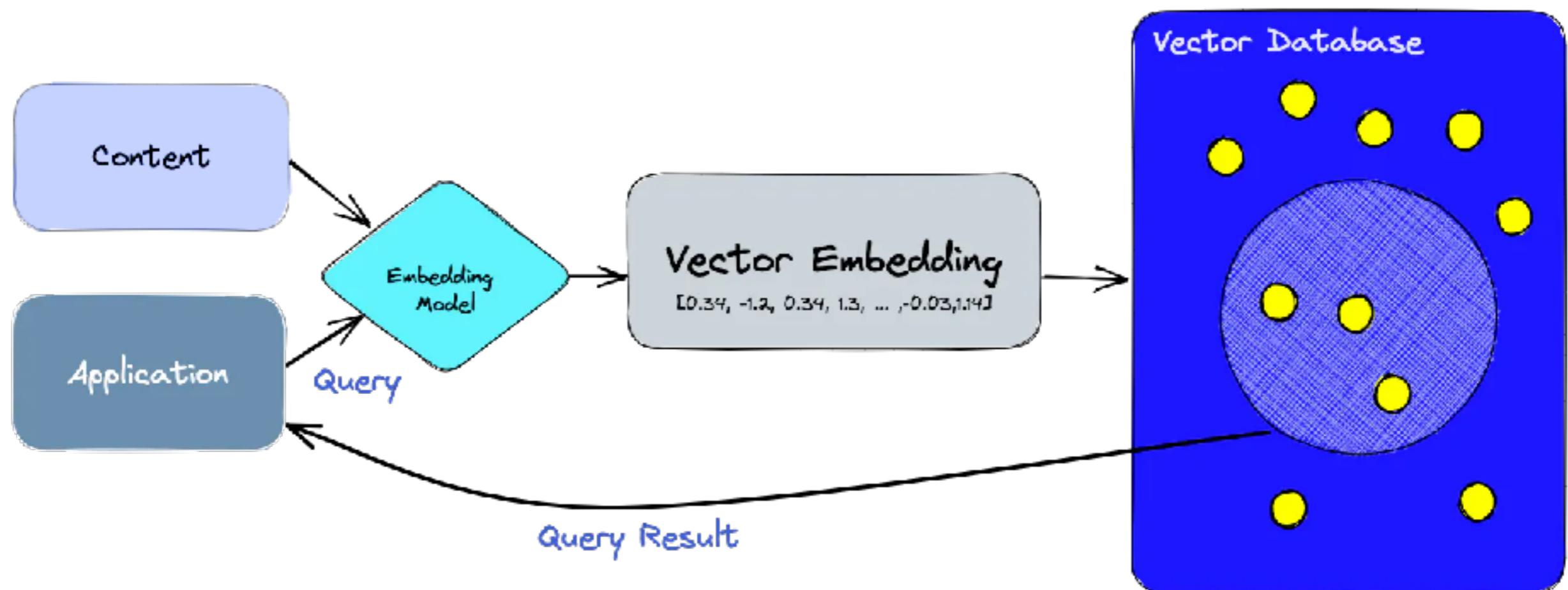
<https://huggingface.co/spaces/mteb/leaderboard>



# Distance measure in Data Science

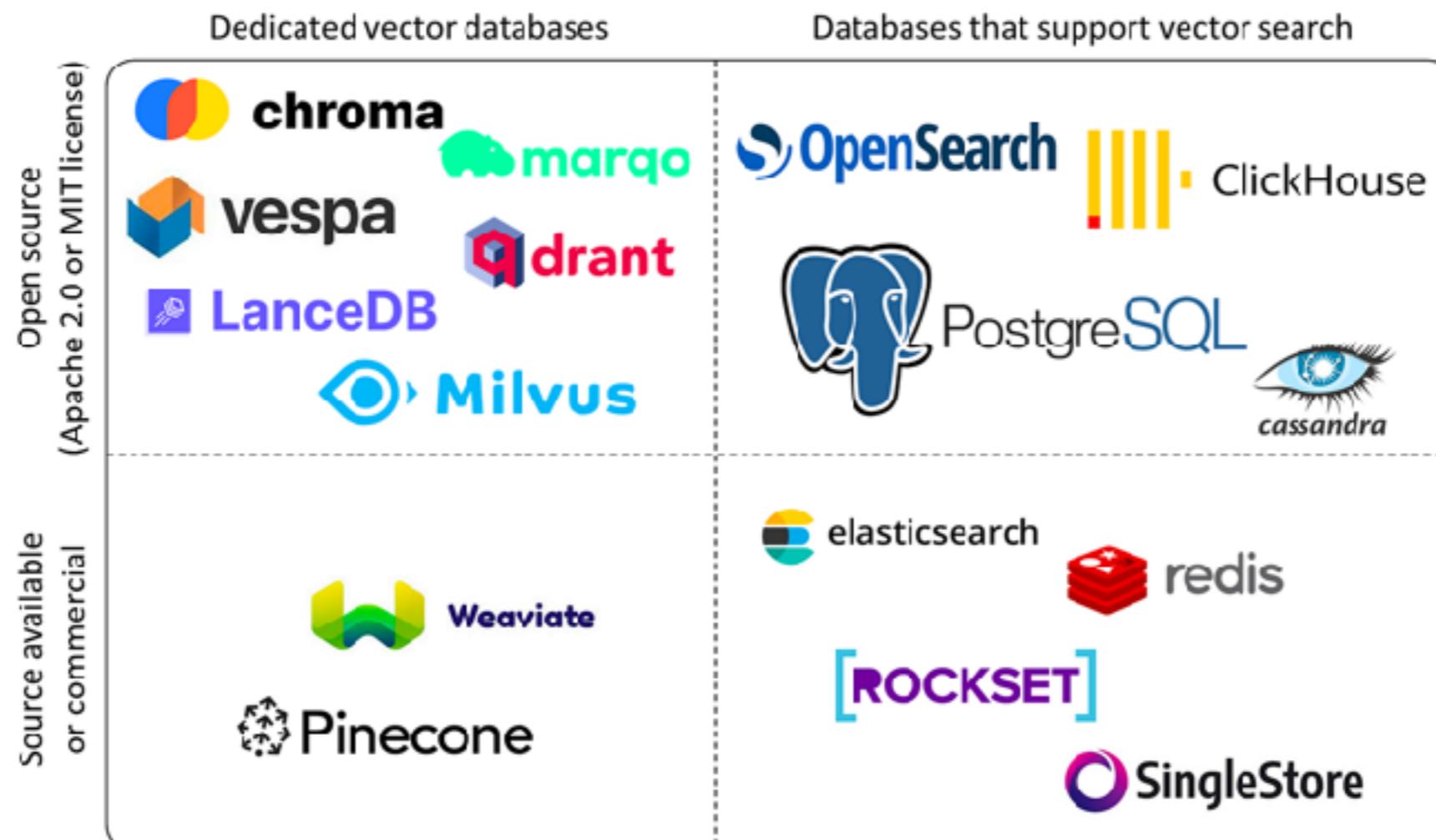


# Store data in Vector Database



# Vector Database ?

Map items of unstructured data to high-dimensional real vectors

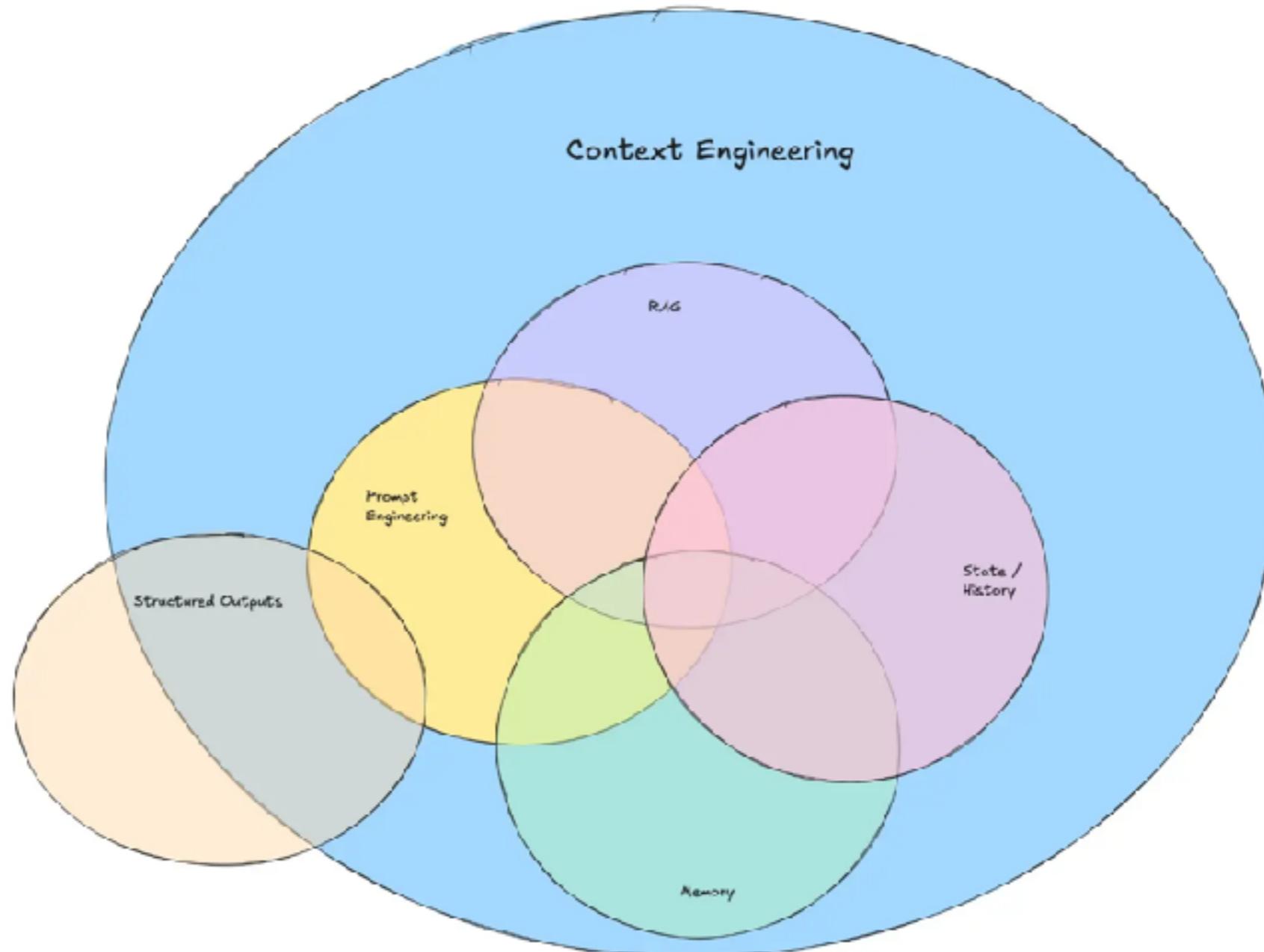


<https://towardsdatascience.com/transformers-in-depth-part-1-introduction-to-transformer-models-in-5-minutes-ad25da6d3cca>





# Context Engineering



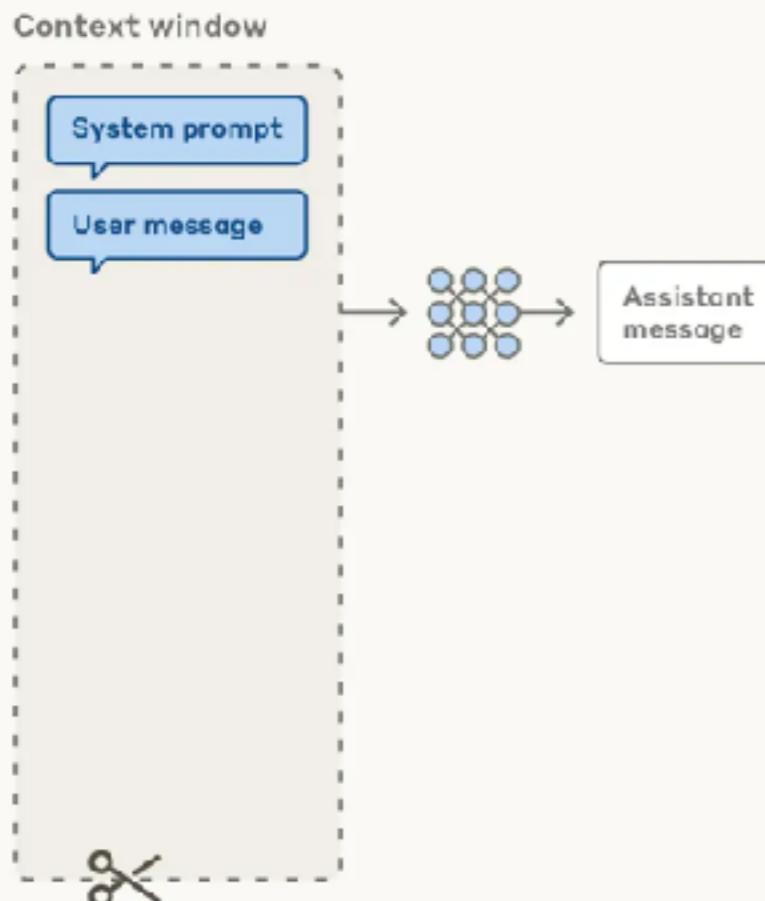
<https://www.promptingguide.ai/guides/context-engineering-guide>



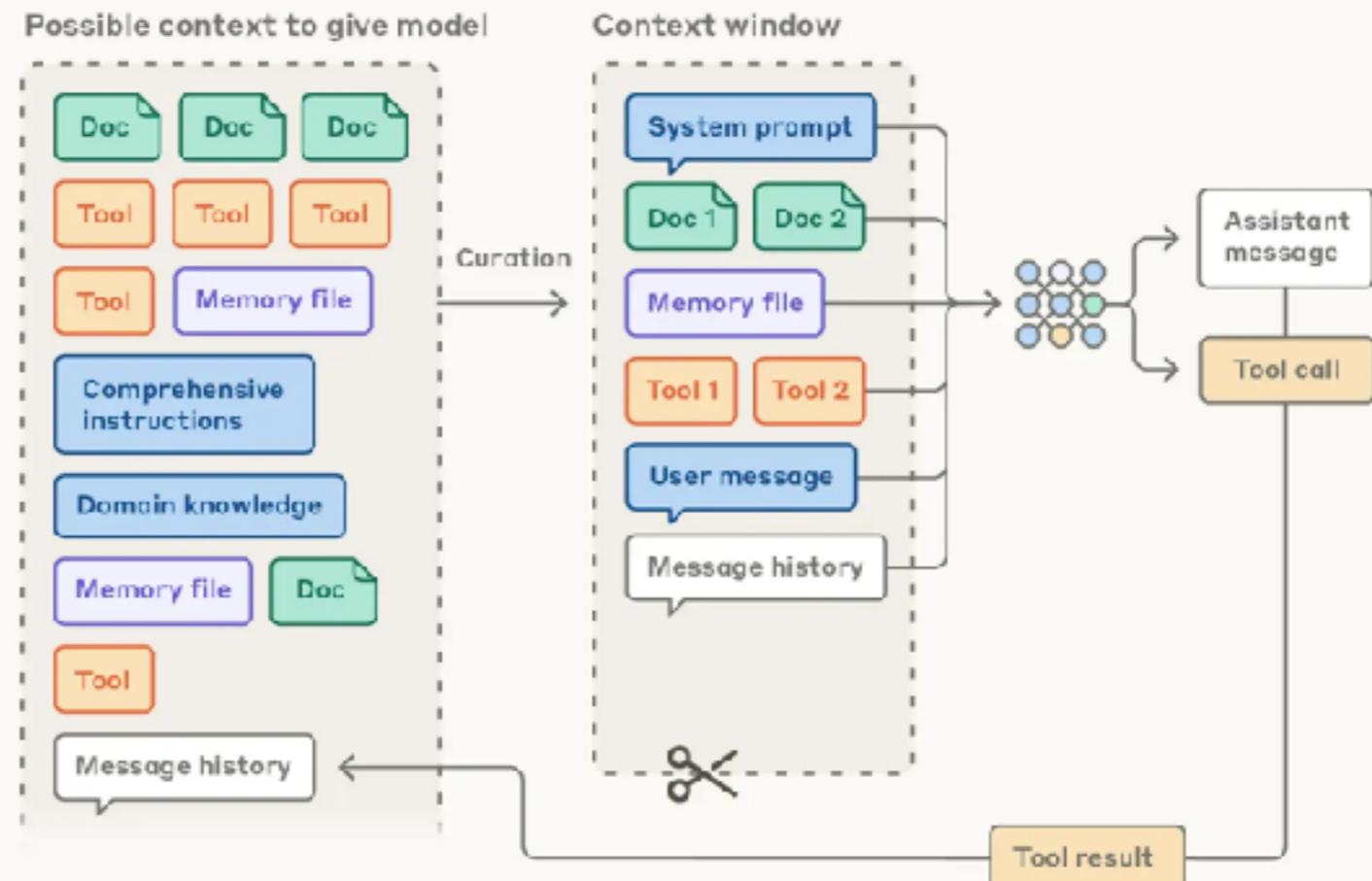
# Effective Context Engineering

## Prompt engineering vs. context engineering

Prompt engineering  
for single turn queries



Context engineering for agents



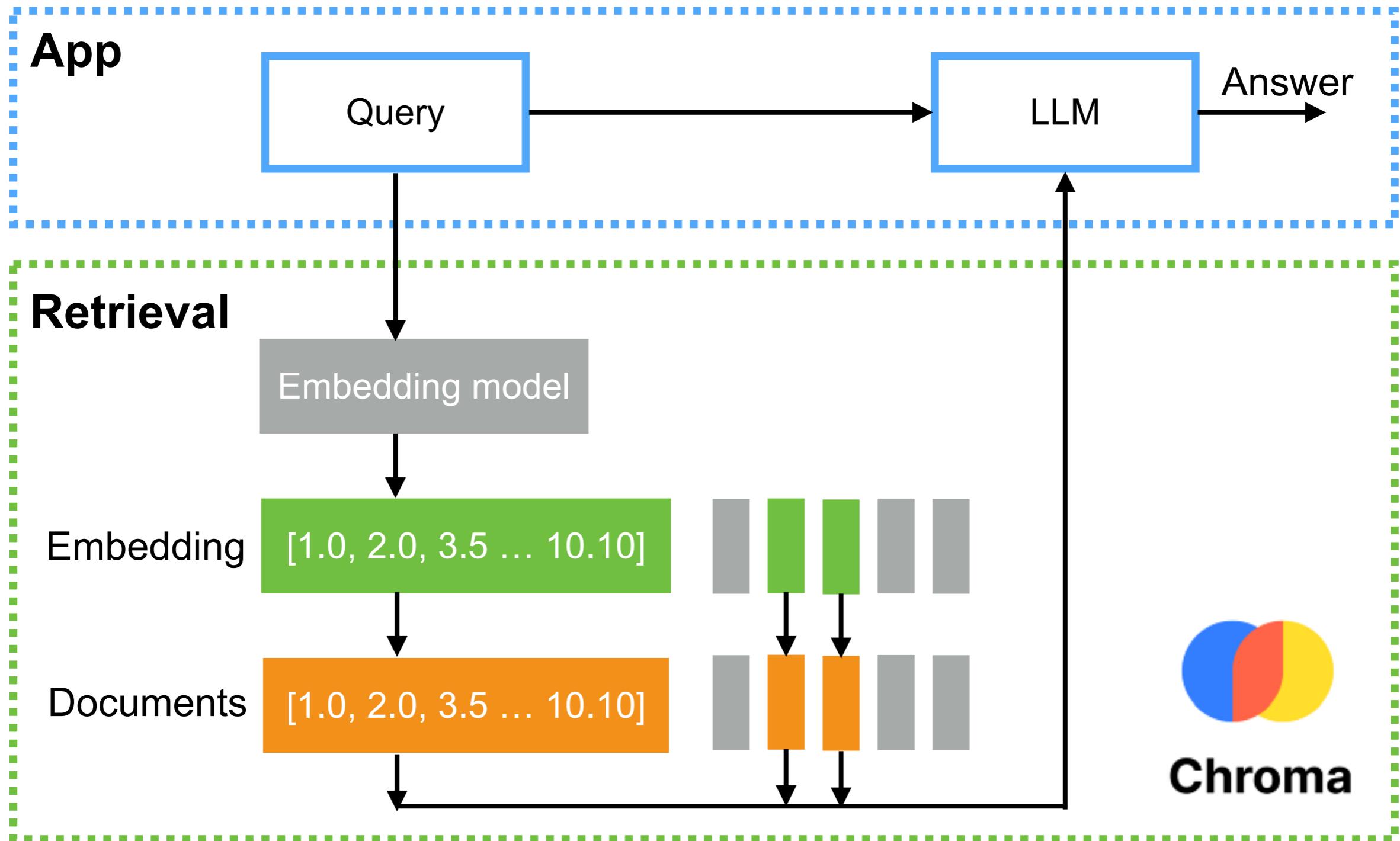
<https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents>



Workshop

© 2020 - 2026 Siam Chamnkit Company Limited. All rights reserved.

# Basic RAG



# Query Expansion

Query expansion is a widely used technique to improve the recall of search systems

Ambiguity

Vocabulary  
mismatch

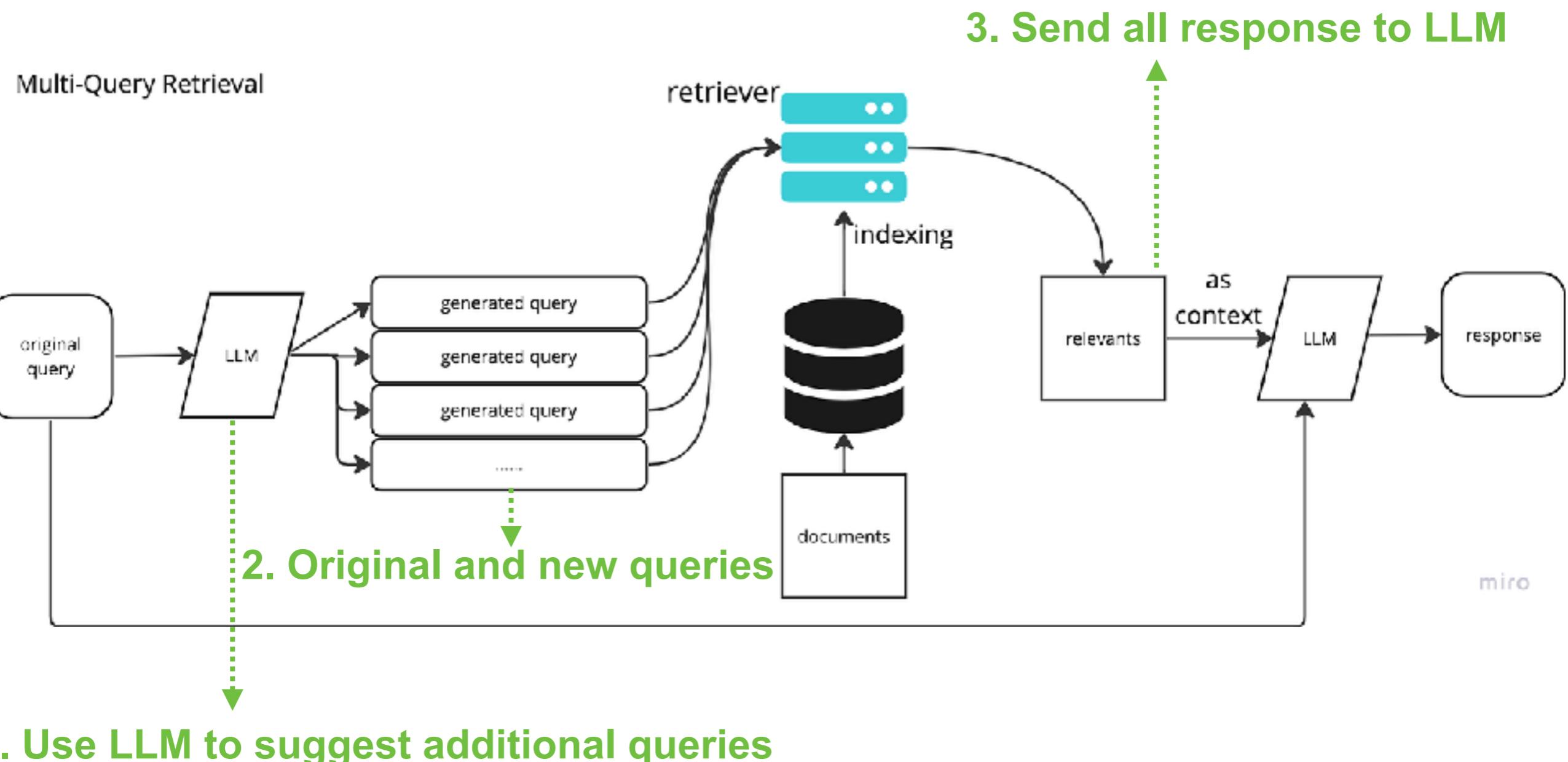
Lack of context

“Add synonyms, related context and contextual terms”

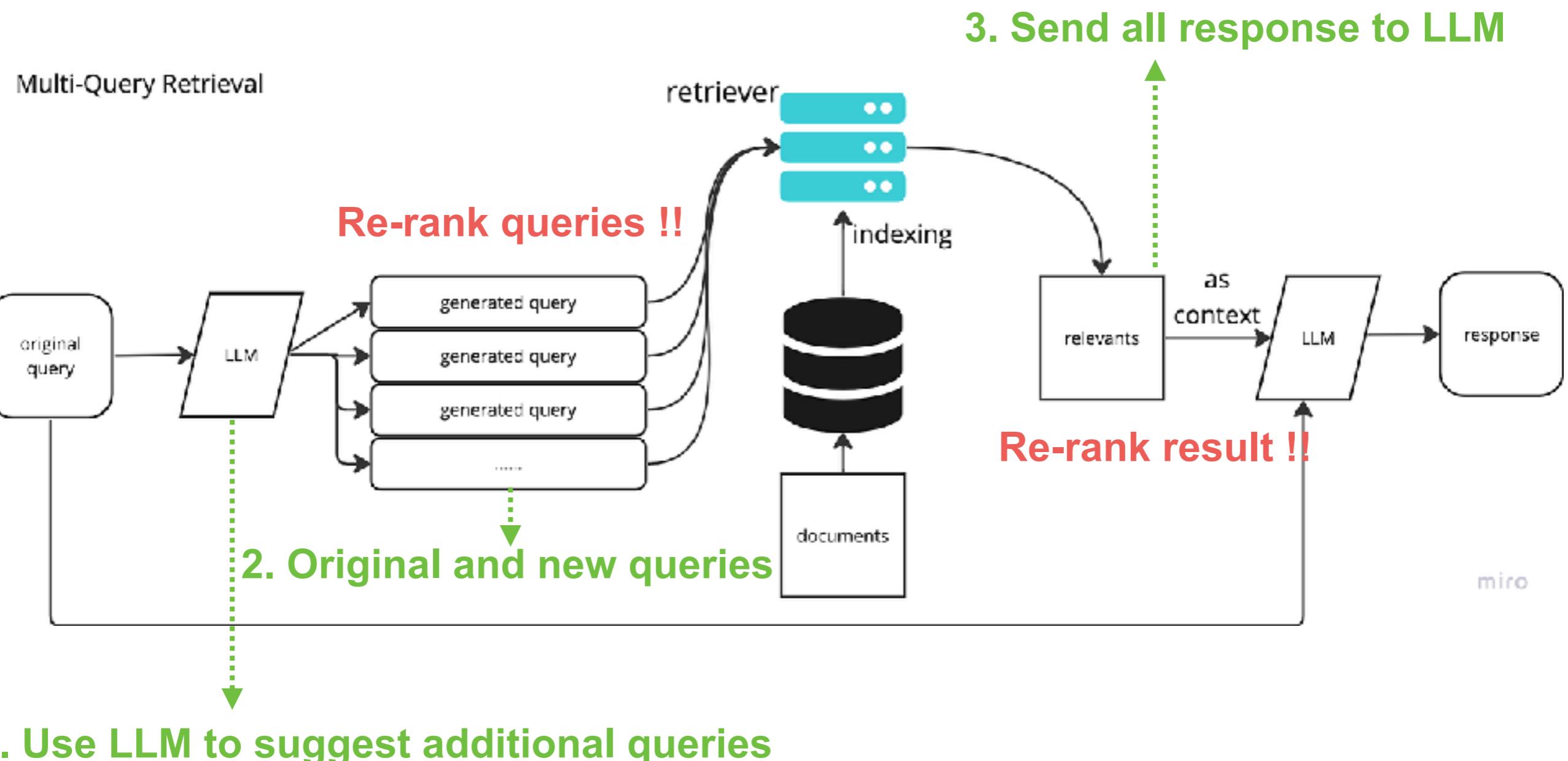
<https://arxiv.org/abs/2305.03653>



# Query Expansion

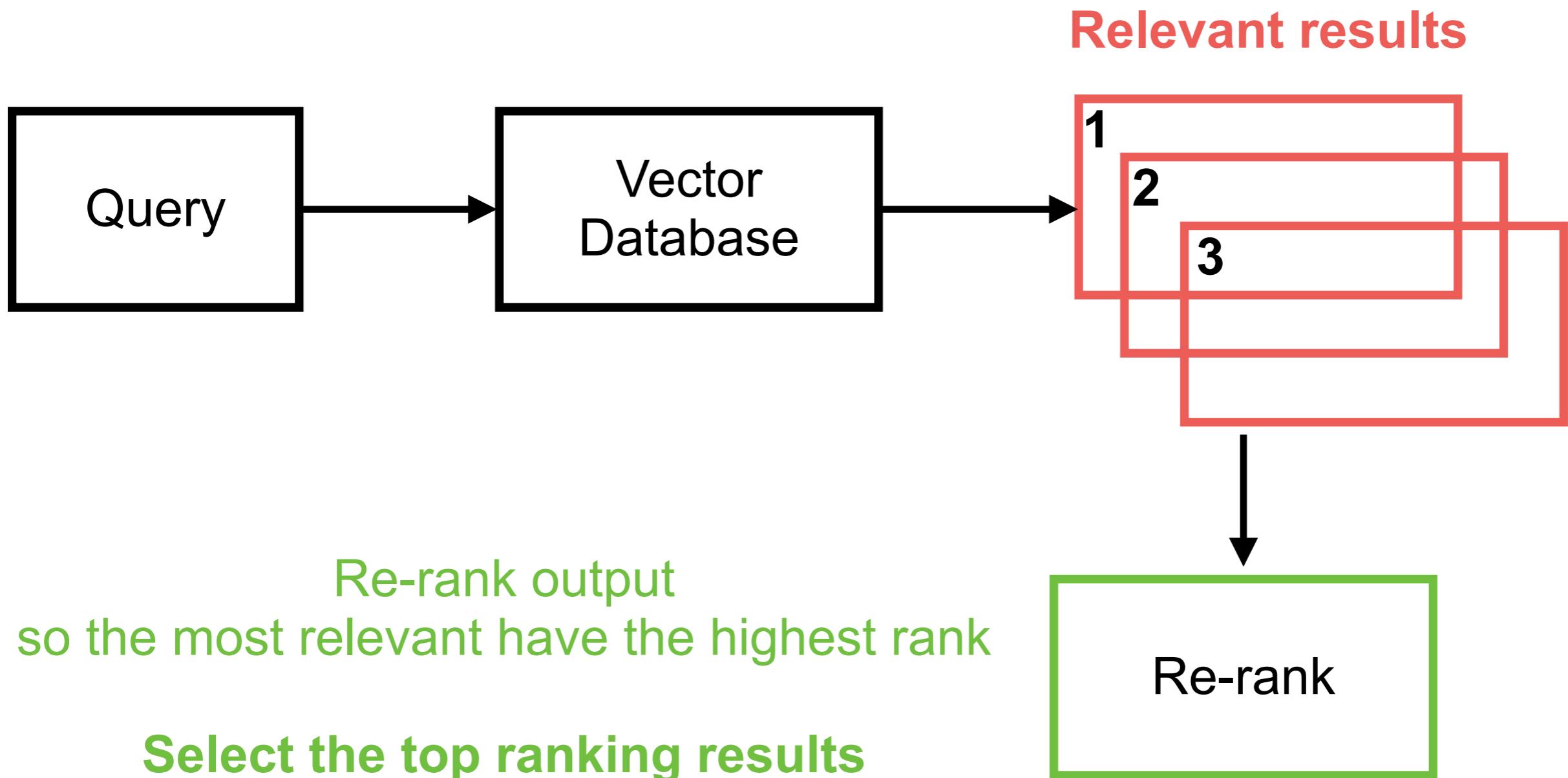


# Re-rank !!

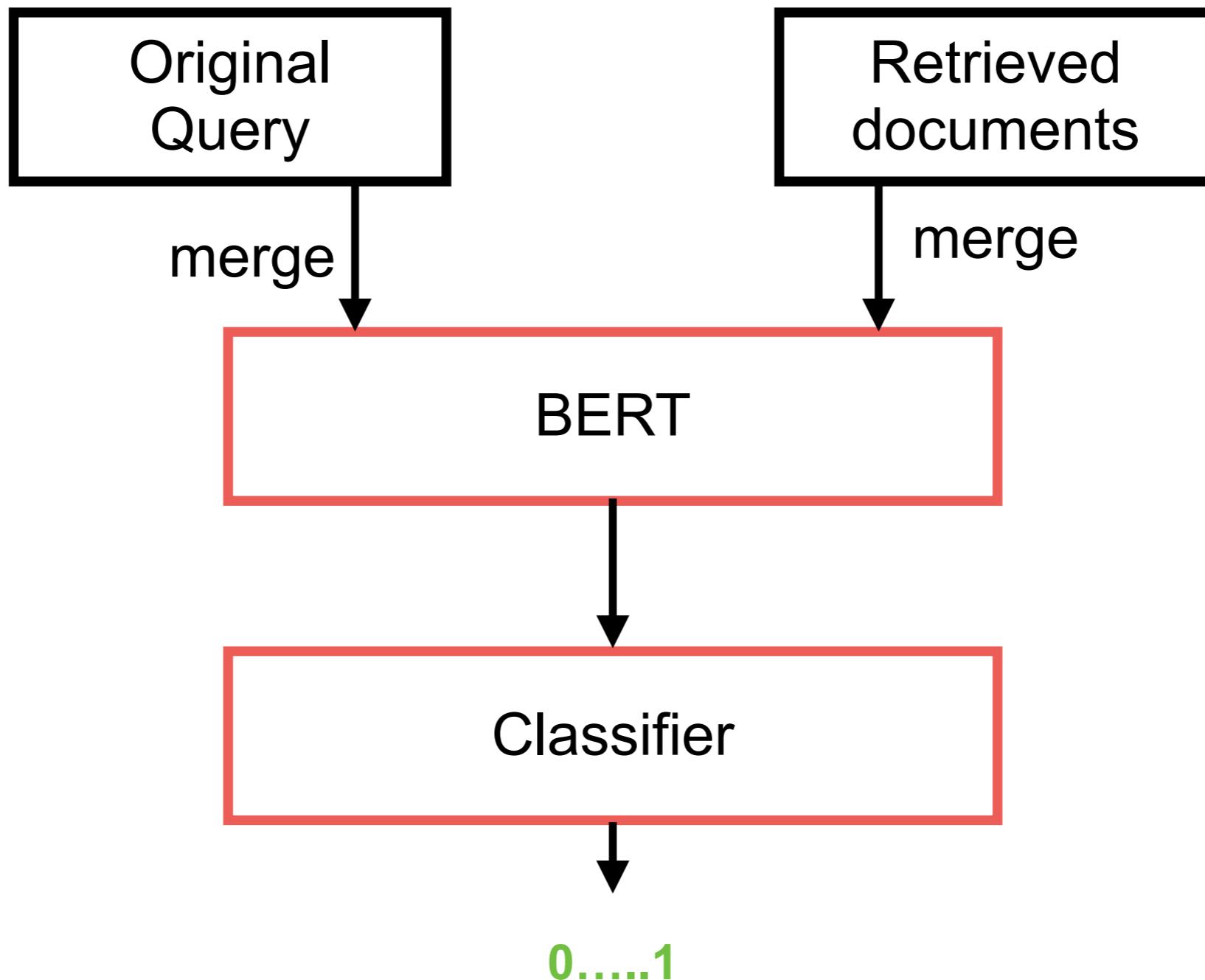


# Cross-encoder Re-ranking

How to ordering relevant results !!

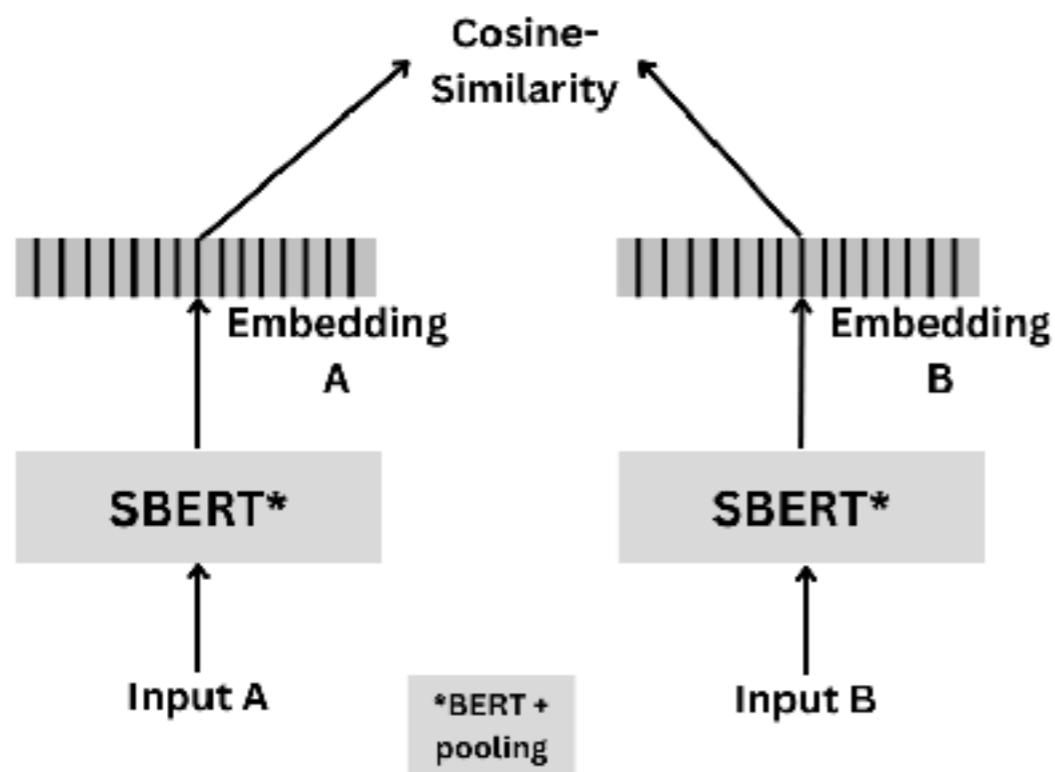


# Cross-Encoder

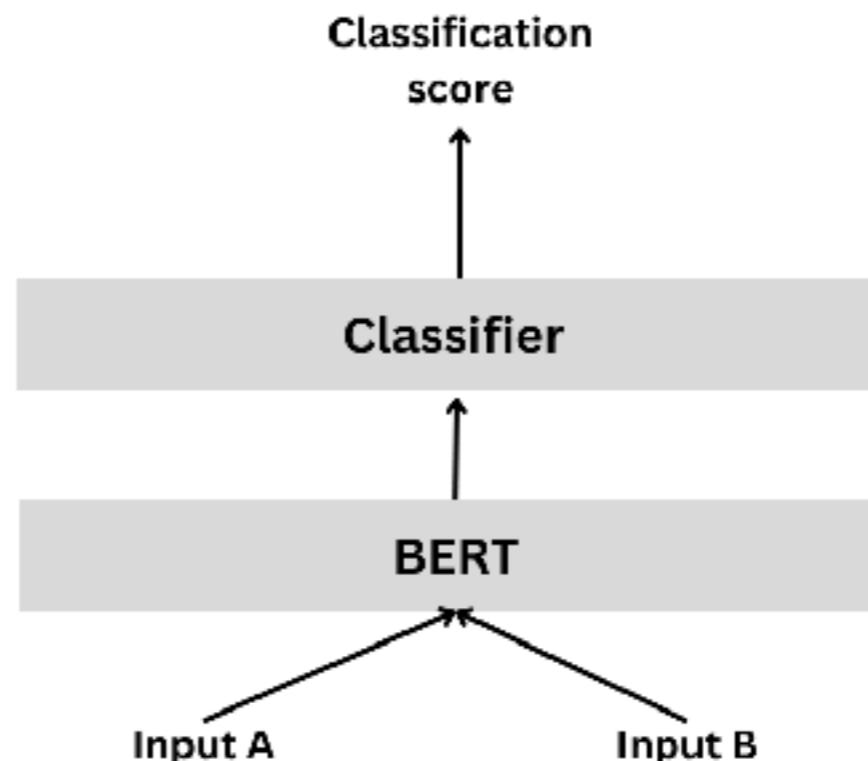


# Cross-Encoder !!

## Bi-encoder



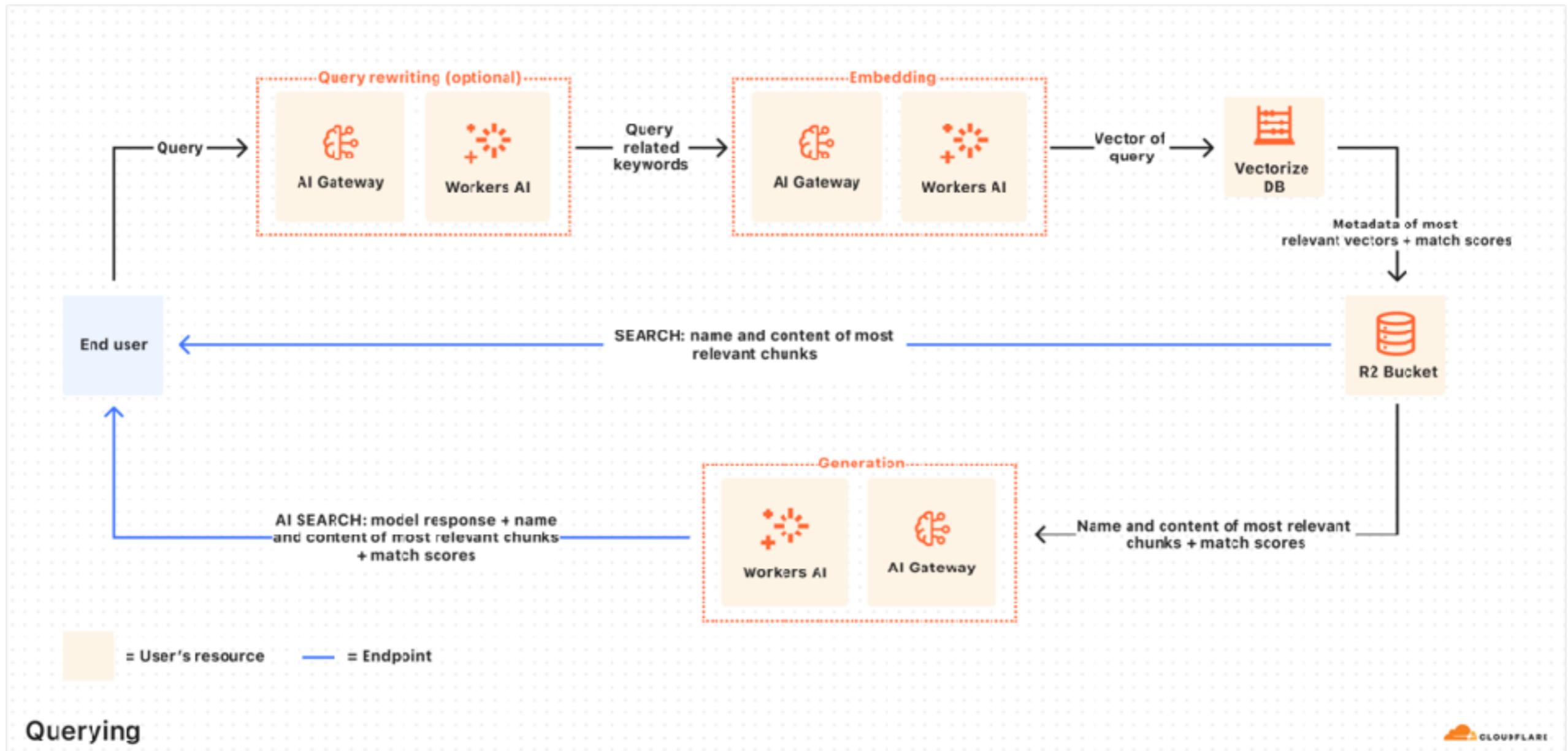
## Cross Encoder



[https://sbert.net/examples/cross\\_encoder/applications/README.html](https://sbert.net/examples/cross_encoder/applications/README.html)



# Cloudflare AutoRAG



<https://developers.cloudflare.com/autorag/>



Workshop

© 2020 - 2026 Siam Chamnkit Company Limited. All rights reserved.

# Local LLM



# Local LLM

Run LLM on local machine/device  
Try to customize with your requirement

Reduce cost

Data privacy

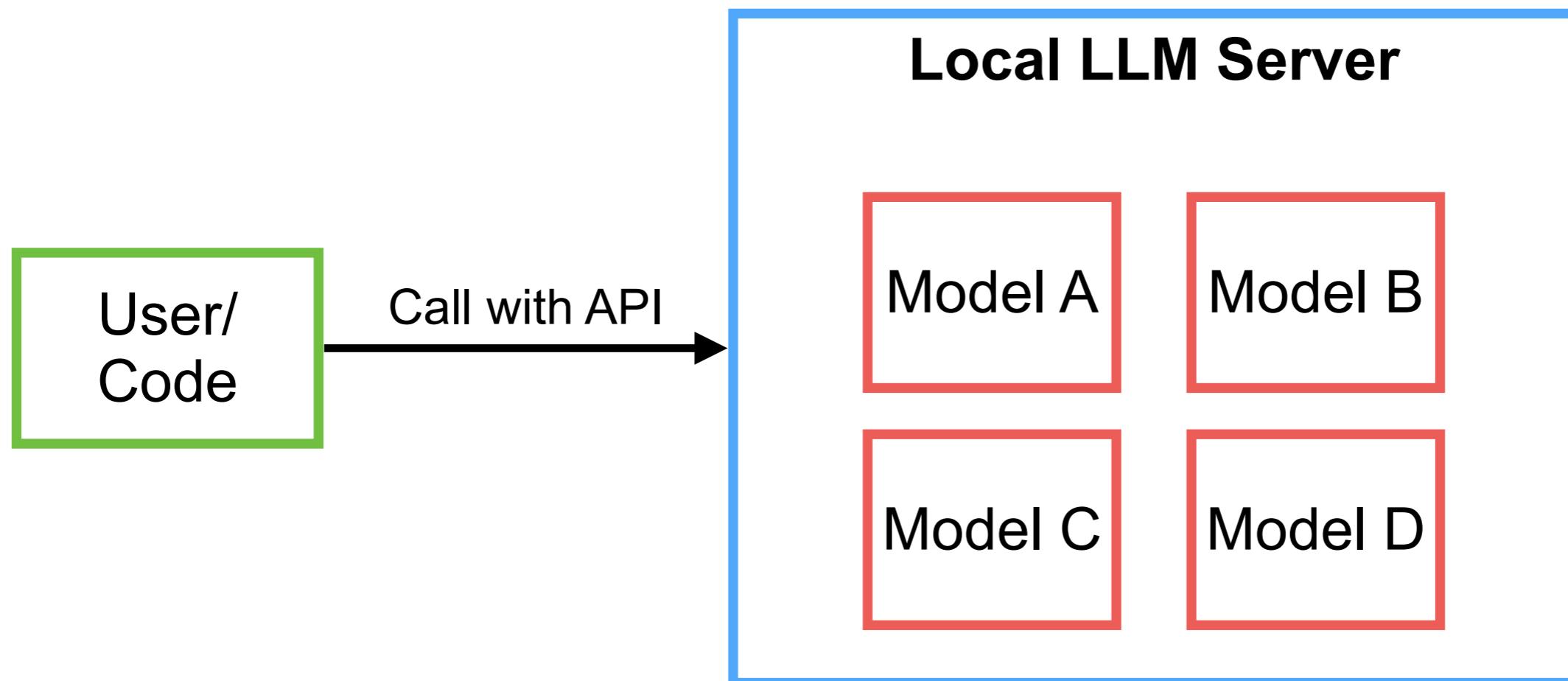
Responsive

Offline mode



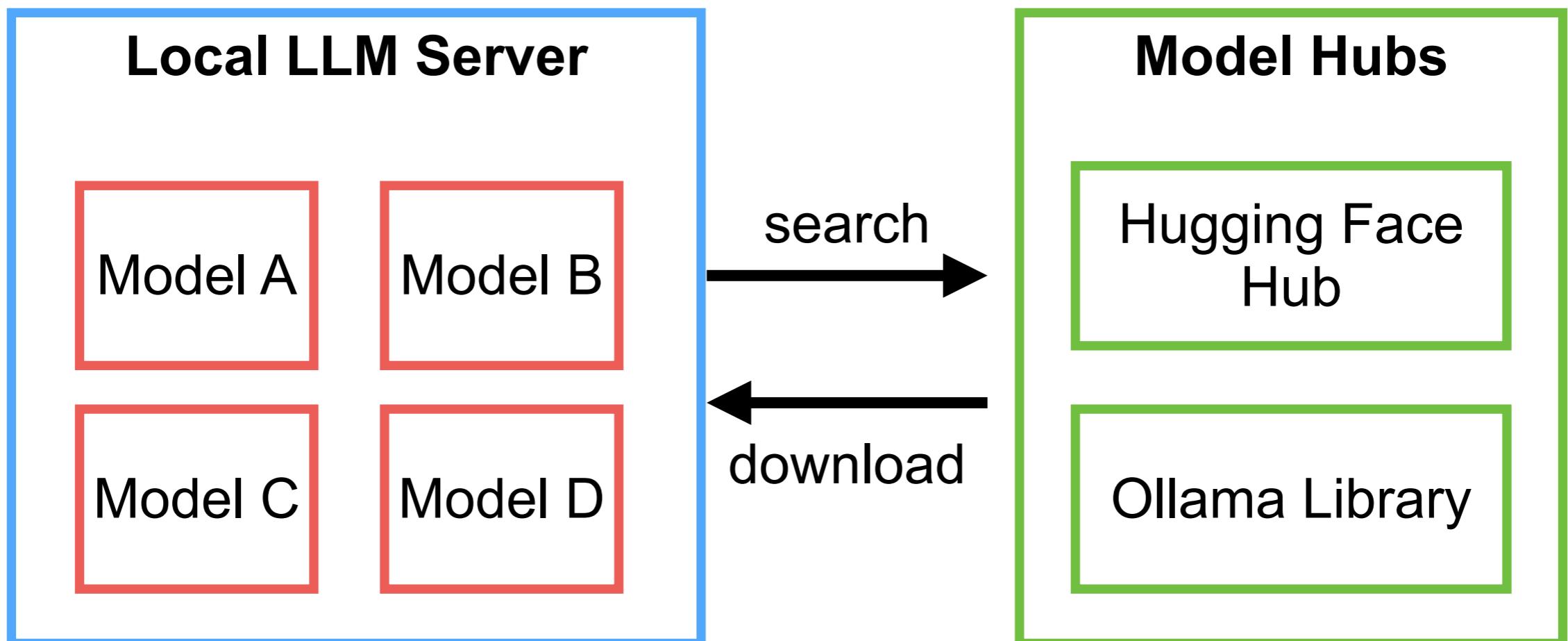
# Local LLM

Improve your LLM models, more accurate answer



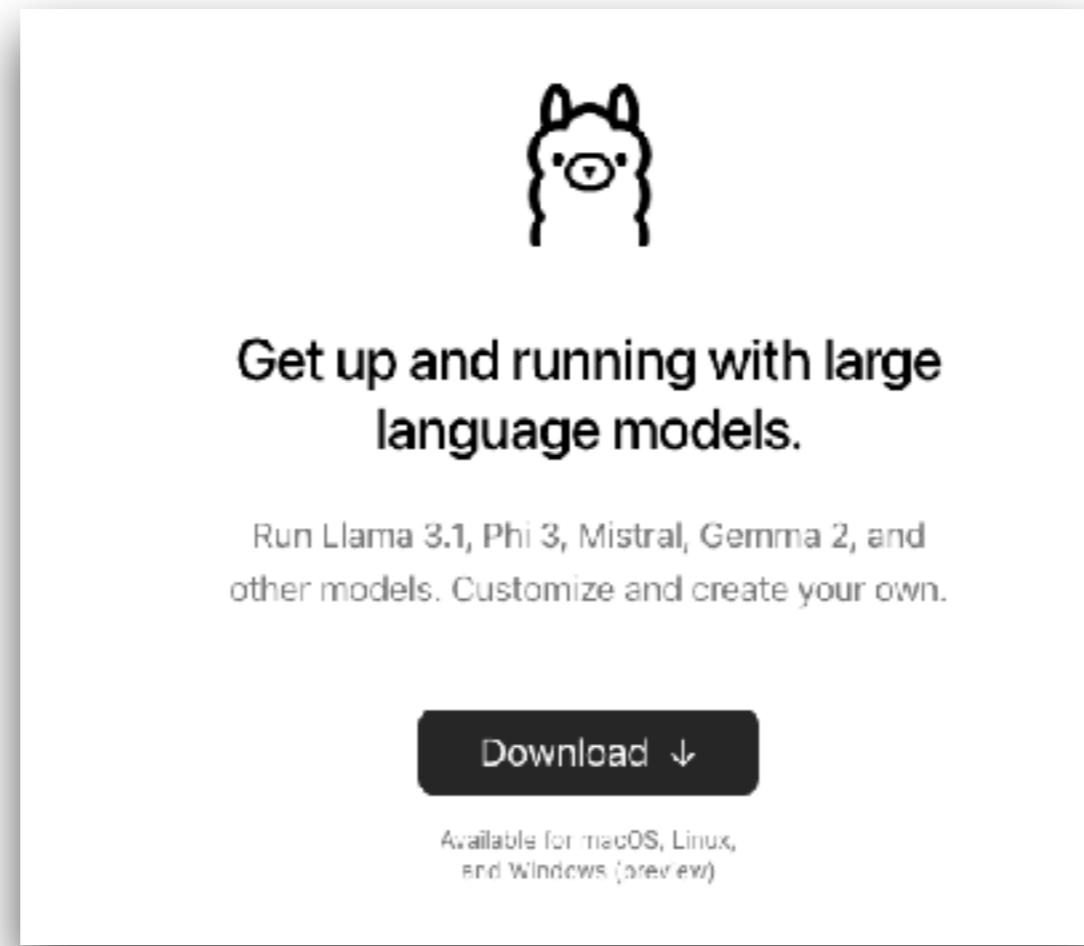
# Models ?

How to download models ?



# Local LLM with Ollama

\$ollama run **llama3.2**



<https://ollama.com/>



# Local LLM with LM Studio

The screenshot shows the LM Studio website on the left and the application interface on the right.

**Website (Left):**

- Header: LM Studio
- Top navigation: Docs, Blog, Download
- Main heading: LM Studio
- Text: Discover, download, and run local LLMs
- Announcement: LM Studio v0.3.0 is finally here! 🎉🎉🎉 Read the announcement
- Text: Run [LLaMa](#) [Phi](#) [Gemma](#) [DeepSeek](#) [Owen](#) [Mistral](#) on your computer ⓘ
- Text: Built with open source projects like [llama.cpp](#) and [lmstudio.js](#)
- Download buttons:
  - [Download LM Studio for Mac \(M1/M2/M3\) 0.3.2](#)
  - [Download LM Studio for Windows \(x64\) 0.3.2](#)
  - [Download LM Studio for Linux \(x86\) 0.3.2](#)
- Text: LM Studio is provided under the [terms of use](#).

**Application (Right):**

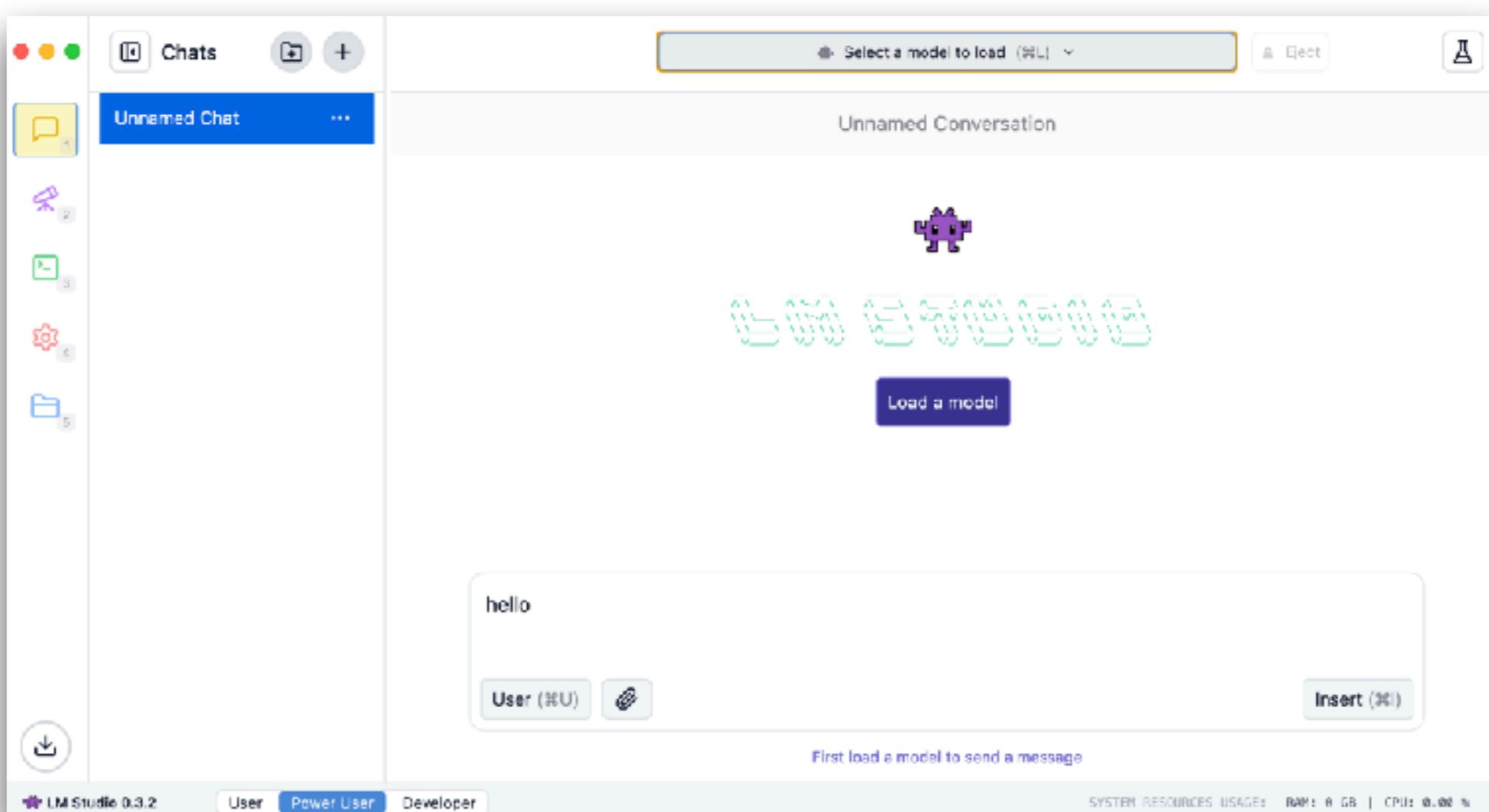
A screenshot of the LM Studio application window titled "LM Studio - Community Edition 0.3.0". The window displays a file tree on the left and configuration settings on the right. The configuration pane includes sections for "Advanced Configuration" (with fields for "System Prompt" and "Model"), "P Serial", and "C Sensors". A cursor is visible over the "C Sensors" section.

<https://lmstudio.ai/>



# Local LLM with LM Studio

Load model from Hugging Face



<https://lmstudio.ai/>



# Local LLM with LlamaEdge



LlamaEdge

Feature FAQ Models Docs | [View on GitHub](#)

The easiest, smallest and fastest local LLM runtime and API server.

[Quick Start with Gaia](#)

Powered by Rust & WasmEdge (A CNCF hosted project) ⓘ

<https://llamaedge.com/>



# Local LLM with LocalAI



<https://localai.io/>



# More

GPT4All

LlamaFile

Jan.ai

NextChat

Anything LLM

<https://github.com/Hannibal046/Awesome-LLM>



# LLM Models



# Hugging Face Model Hub

NEW AI Tools are now available in HuggingChat

The AI community building the future.

The platform where the machine learning community collaborates on models, datasets, and applications.

Models 450,541

Tasks

- Vision
- Computer Vision
- Natural Language Processing
- Audio
- Tabular
- Reinforcement Learning

Datasets

Spaces

Posts

Docs

Pricing

Log In

Sign Up

meta-llama/Llama-2-7b

Text Generation • Updated 4 days ago • 25.2k • 4.6k

stabilityai/stable-diffusion-v1-base

openai/openchat

lllyasviel/ControlNet-v1-2

ceresense/zeroscope\_v2\_XL

meta-llama/Llama-2-20b

tiiuae/falcon-40b-instruct

MirroredH/MixedCodes-v18-v1.0

CompVis/stable-diffusion-v2-4

StabilityAI/stable-diffusion-v2-2

Salesforce/qwen-2b-8w-inat

<https://huggingface.co/>



# Hugging Face :: Model

The screenshot shows the Hugging Face Model Hub interface. On the left, there's a sidebar with sections for Tasks (Libraries, Datasets, Languages, Licenses), Other, Filter Tasks by name, Multimodal (Image-Text-to-Text, Visual Question Answering, Document Question Answering, Video-Text-to-Text, Any-to-Any), and Computer Vision (Depth Estimation, Image Classification, Object Detection, Image Segmentation, Text-to-Image, Image-to-Text, Image-to-Image, Image-to-Video, Unconditional Image Generation, Video Classification, Text-to-Video, Zero-Shot Image Classification, Mask Generation, Zero-Shot Object Detection, Text-to-3D). The main area is titled 'Models 233,861' and has a search bar with 'llama'. It includes 'Full-text search' and 'Sort: Trending' buttons. The search results list several Llama models:

- black-forest-labs/FLUX.1-dev** (Text-to-Image, Updated Aug 16, 919k, 4.73k)
- meta-llama/Meta-Llama-3.1-8B-Instruct** (Text Generation, Updated Aug 21, 3.09M, 2.61k)
- jinaai/reader-lm-1.5b** (Text Generation, Updated 5 days ago, 8.28k, 382)
- black-forest-labs/FLUX.1-schnell** (Text-to-Image, Updated Aug 16, 1.06M, 2.35k)
- nvidia/Llama-3\_1-Nemotron-51B-Instruct** (Text Generation, Updated about 14 hours ago, 61, 79)
- dleemiller/word-llama-12-supercat** (Updated Aug 12, 81)
- ICTNLP/Llama-3.1-8B-Omni** (Updated 12 days ago, 1.39k, 324)

<https://huggingface.co/>



# Big Code model leader board

★ Big Code Models Leaderboard

Inspired from the [Open LLM Leaderboard](#) and [Open LLM-Perf Leaderboard](#), we compare performance of base multilingual code generation models on [HumanEval](#) benchmark and [MultiPL-E](#). We also measure throughput and provide information about the models. We only compare open pre-trained multilingual code models, that people can start from as base models for their trainings.

Evaluation table    Performance Plot    About    Submit results

See All Columns

Search for your model and press ENTER...

Filter model types

all     base     instruction-tuned     EXT external-evaluation

T	Model	Win Rate	humaneval-python	java	javascript	c++
♦ EXT	<a href="#">OpenCodeInterpreter-DS-33B</a>	55.83	75.23	54.8	69.06	64.47
♦ EXT	<a href="#">Nxcode-CQ-7B-croc</a>	55.42	87.23	60.91	71.69	68.04
♦	<a href="#">CodeQwen1.5-7B-Chat</a>	55.08	87.2	61.04	70.31	67.85
♦ EXT	<a href="#">CodeFuse-DeepSeek-33b</a>	54.33	76.83	60.76	66.46	65.22
♦ EXT	<a href="#">DeepSeek-Coder-33b-instruct</a>	52	80.02	52.03	65.13	62.36
♦ EXT	<a href="#">Artigenz-Coder-DS-6.7B</a>	51.5	70.89	56.84	66.16	59.75
♦ EXT	<a href="#">DeepSeek-Coder-7b-instruct</a>	50.33	86.22	53.34	65.8	59.66
♦ EXT	<a href="#">OpenCodeInterpreter-DS-6.7B</a>	49.67	73.2	51.41	63.85	60.81

<https://huggingface.co/spaces/bigcode/bigcode-models-leaderboard>



# Model in Ollama

The screenshot shows the Ollama library interface. At the top left is a circular icon of a cartoon llama head. To its right, the word "Models" is displayed. Below this is a search bar containing the text "deepseek". To the right of the search bar is a dropdown menu set to "Featured".

The first model listed is "deepseek-coder-v2". Its description reads: "An open-source Mixture-of-Experts code language model that achieves performance comparable to GPT4-Turbo in code-specific tasks." Below the description are three blue buttons labeled "Code", "16B", and "236B". Underneath these buttons are three small icons with the numbers "307K", "65", and "3 months ago" respectively.

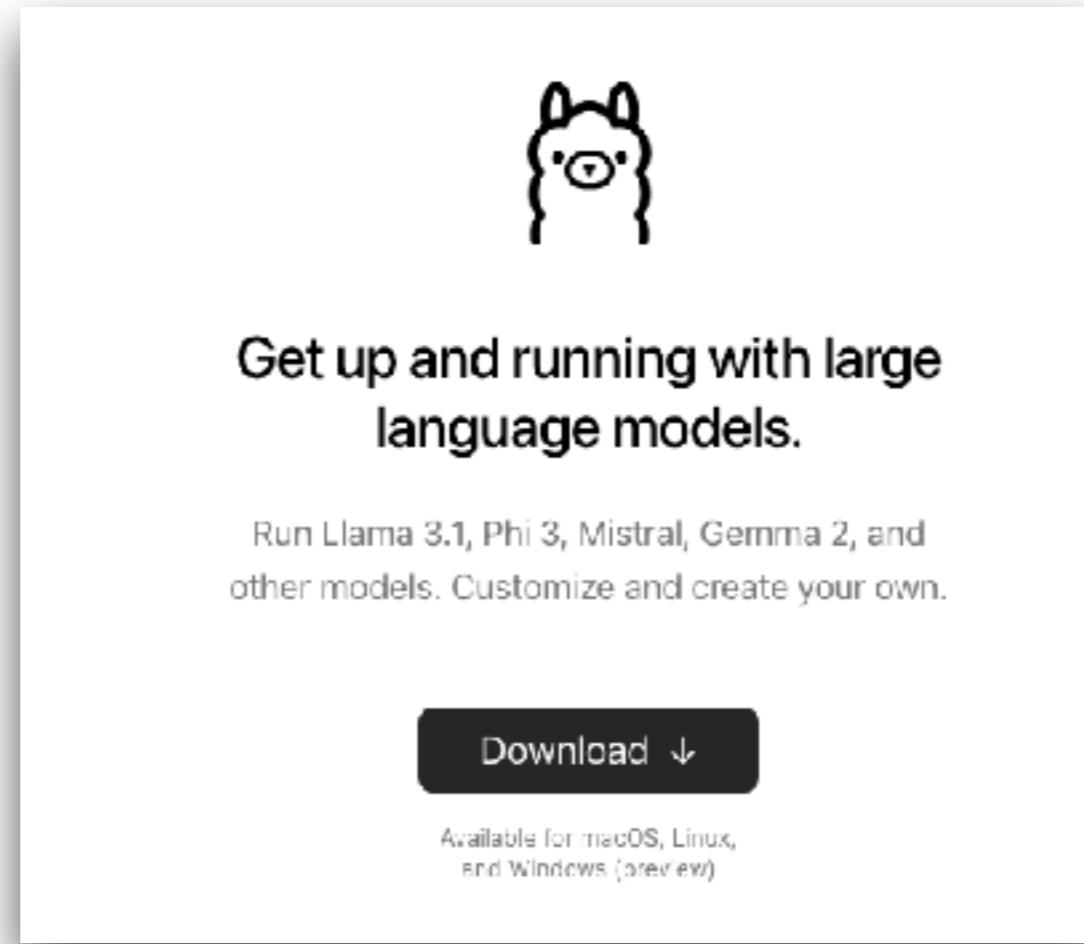
The second model listed is "deepseek-coder". Its description reads: "DeepSeek Coder is a capable coding model trained on two trillion code and natural language tokens." Below the description are four blue buttons labeled "Code", "1B", "7B", and "33B". Underneath these buttons are three small icons with the numbers "303.9K", "102", and "9 months ago" respectively.

<https://ollama.com/library>



# Workshop with Ollama

\$ollama run **llama3.1**



<https://github.com/up1/workshop-ai-with-technical-team/wiki/Local-LLM-with-Ollama>



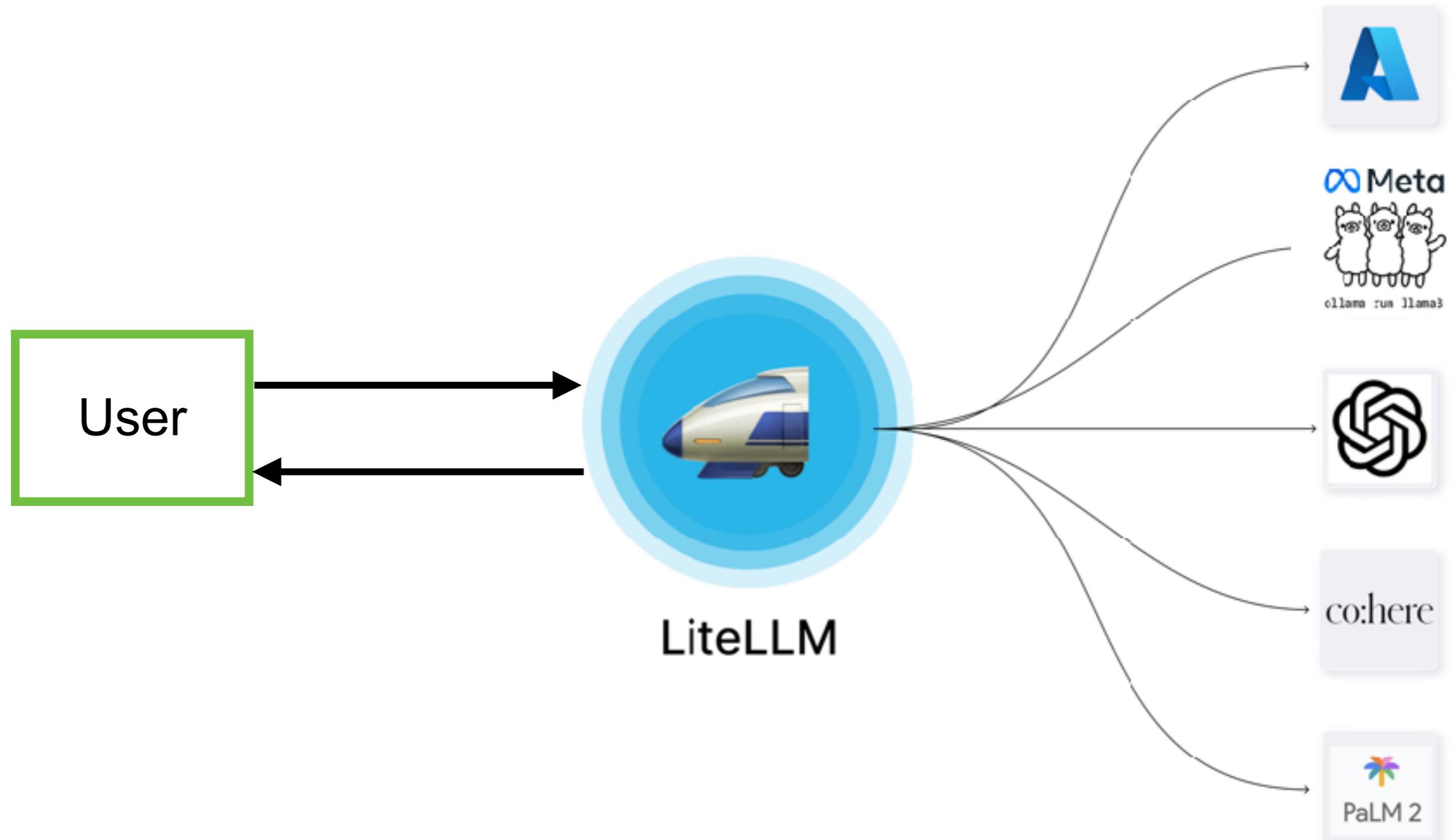
Workshop

© 2020 - 2026 Siam Chamnankit Company Limited. All rights reserved.

# **LiteLLM as a Proxy**



# LiteLLM as a Proxy

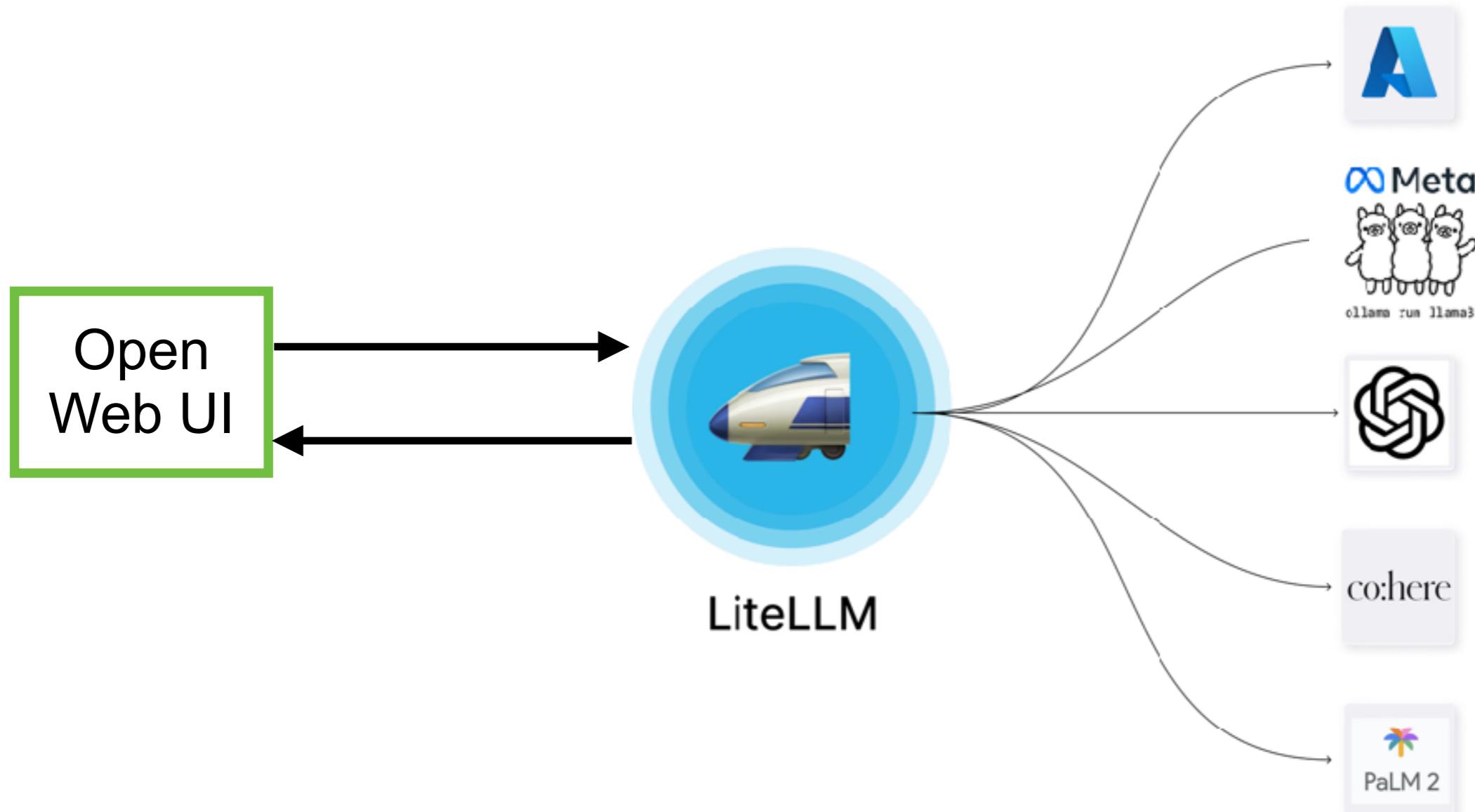


<https://www.litellm.ai/>



# Workshop

Use docker compose to build and run



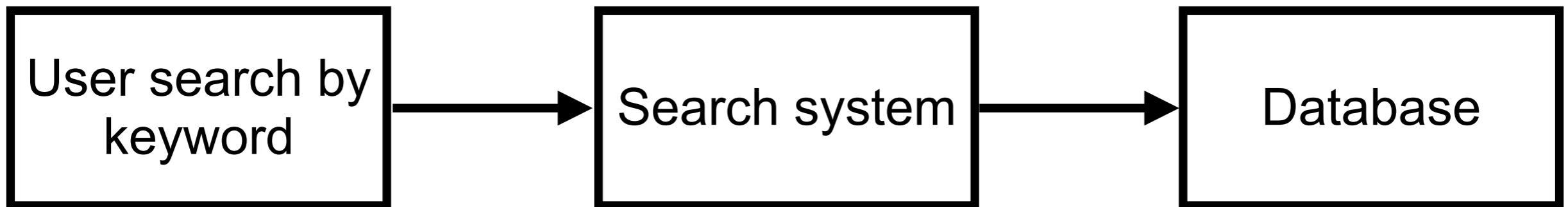
<https://github.com/up1/workshop-ai-with-technical-team/wiki/LiteLLM-and-WebUI>



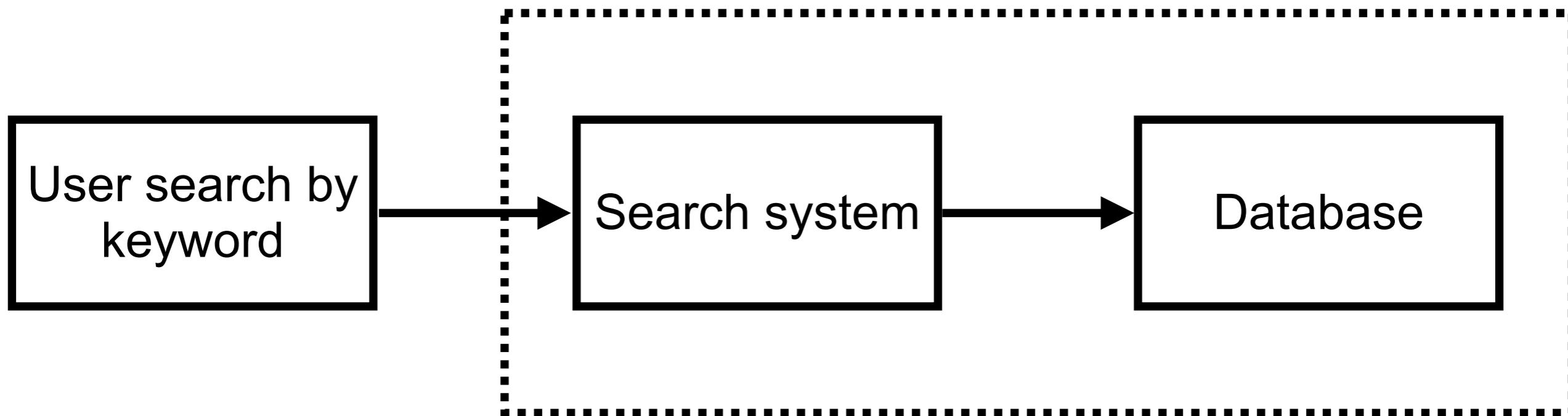
# Flow of search system



# Search from user



# Search from user



Improve search ?  
Keyword, semantic, hybrid ?



# Search Techniques ?

NAME LIKE %ຂໍ້ມູນ%  
Full-text search (FTS)

Need more accuracy !!



# Start your journey

