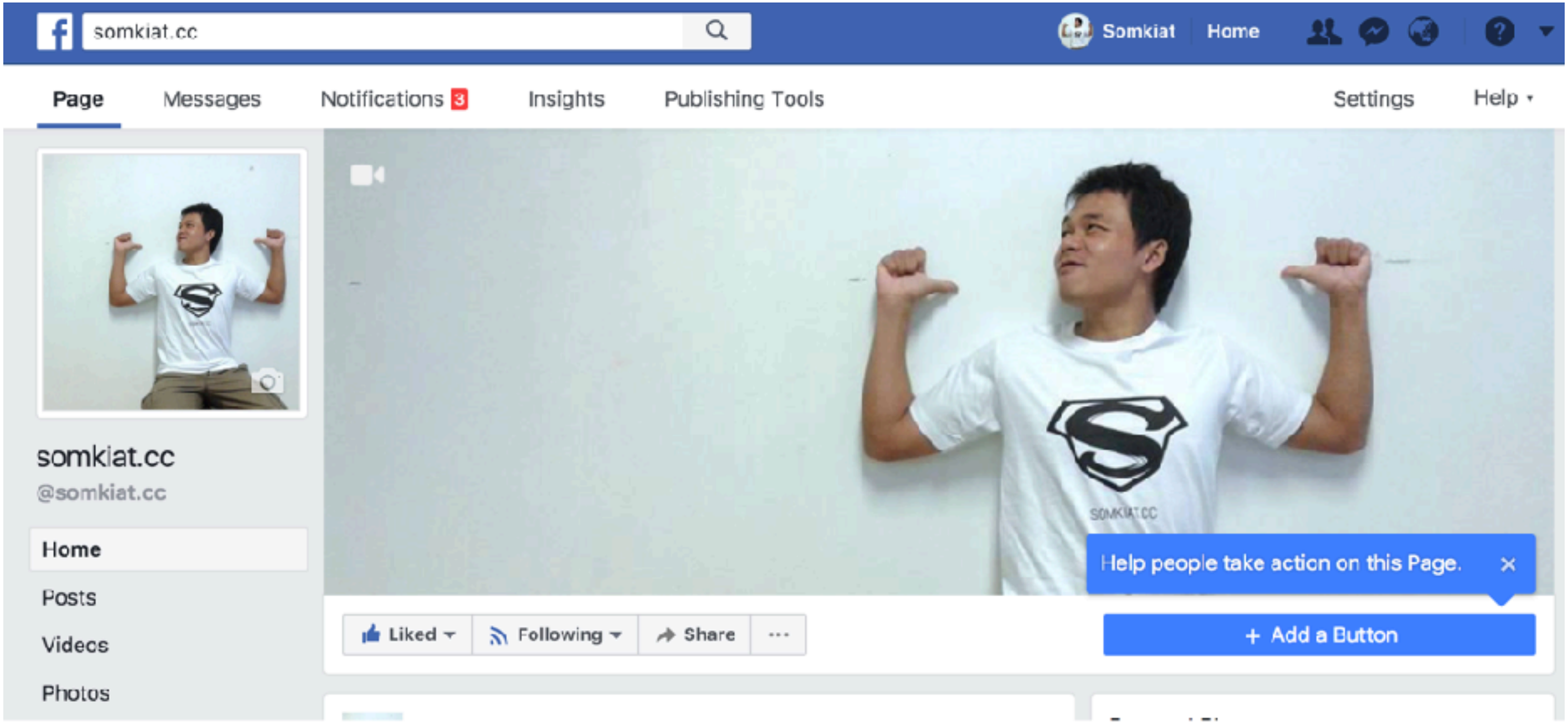
สยามชำนาญกิจ
SIAM CHAMNAN KIT

Improve search system
Keyword search
Semantic search
Hybrid 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

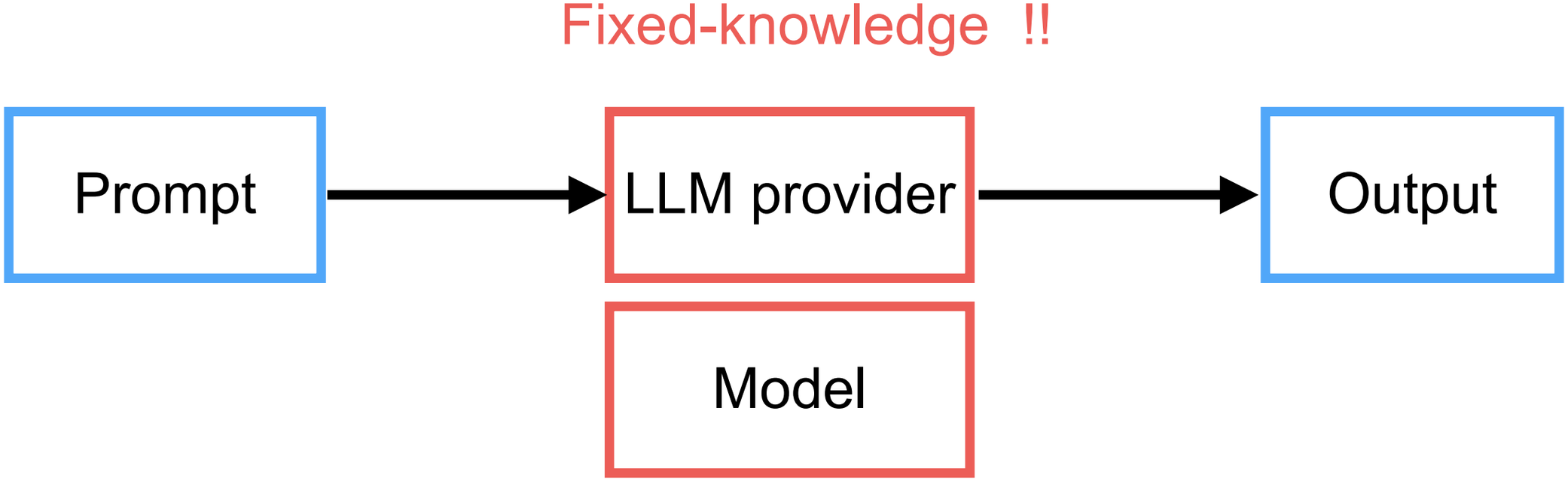
Generator

Uses the retrieved information to generate a response

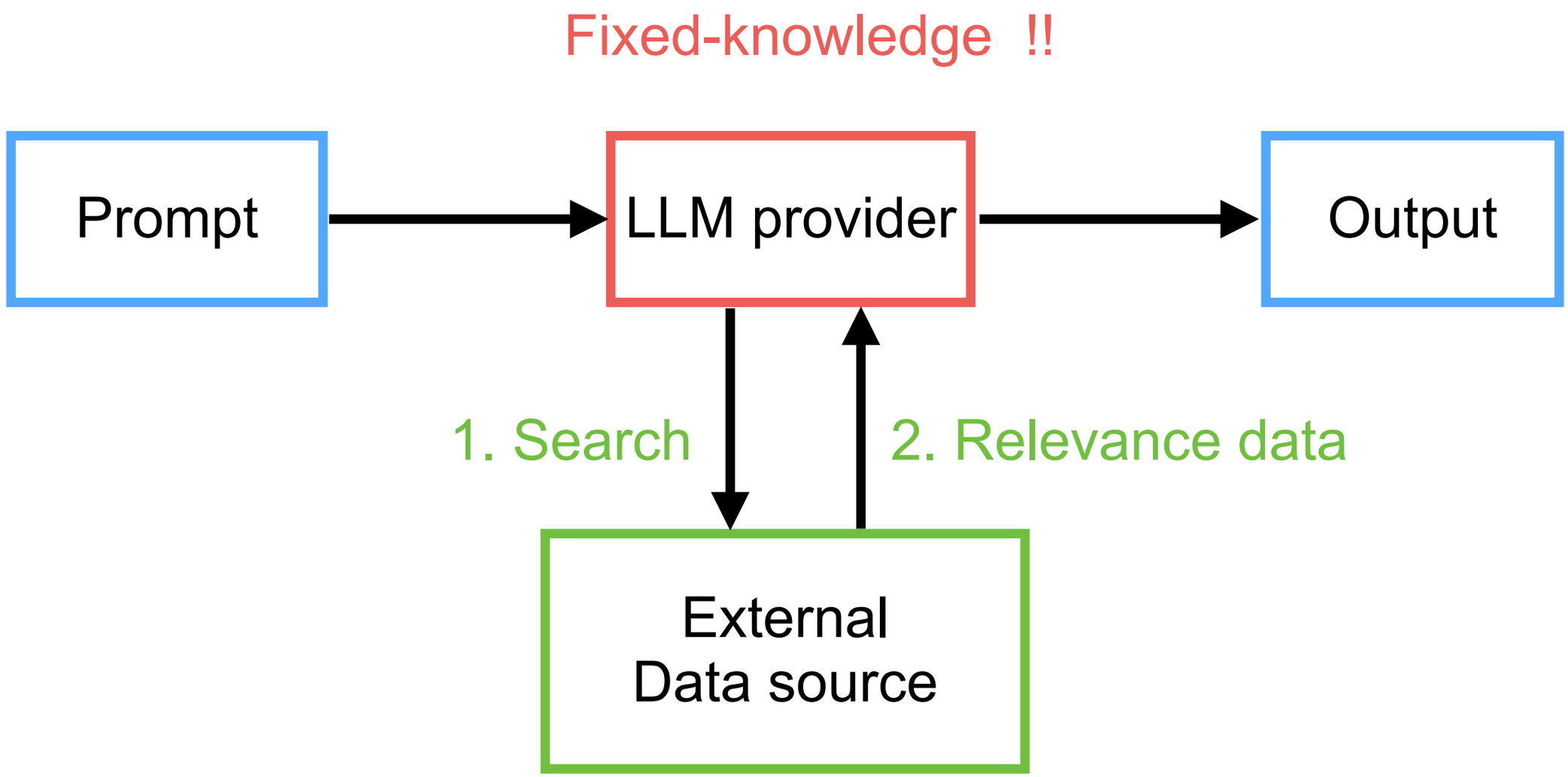
Enhancements

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

Limitation of LLM



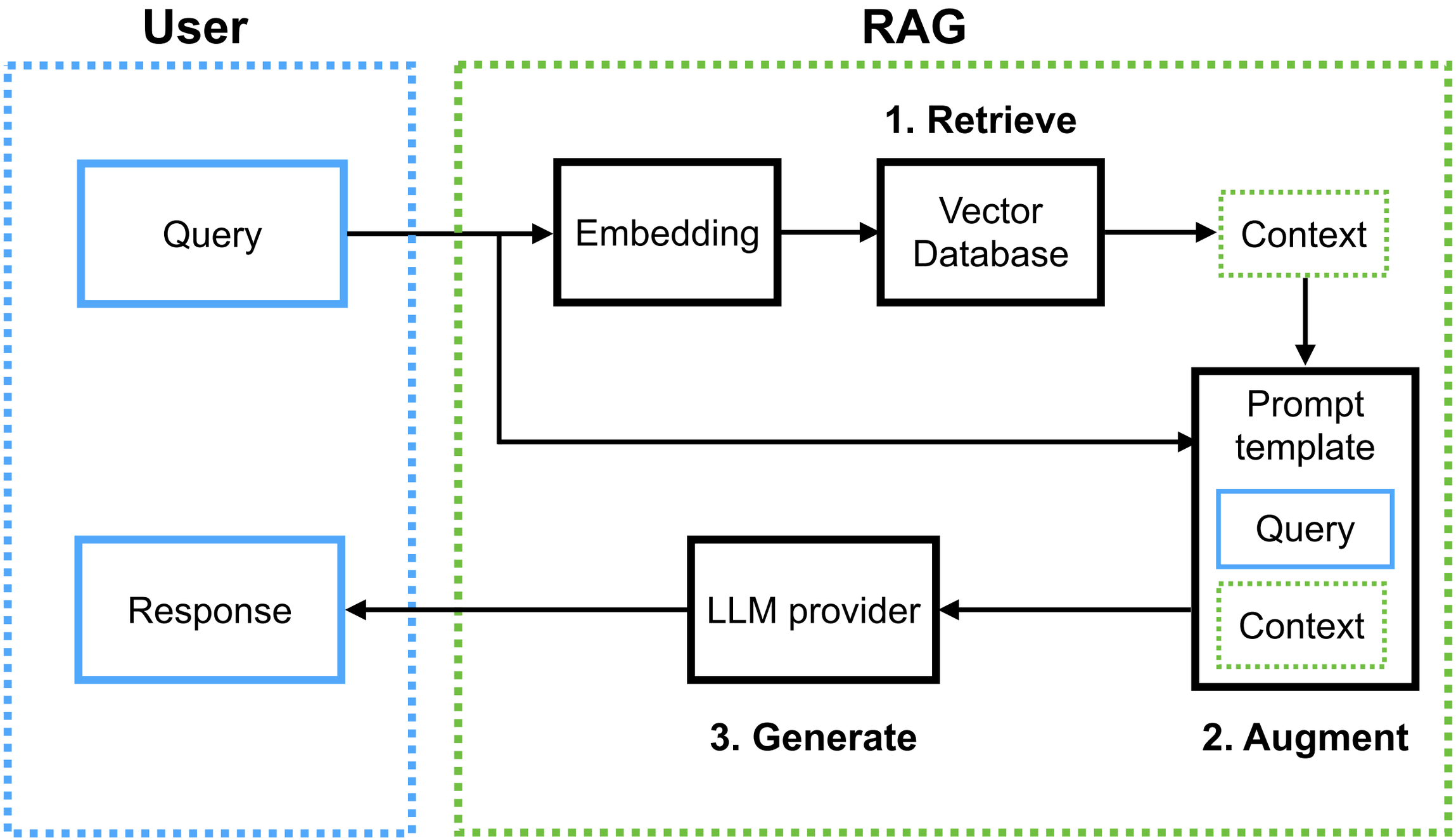
RAG ?



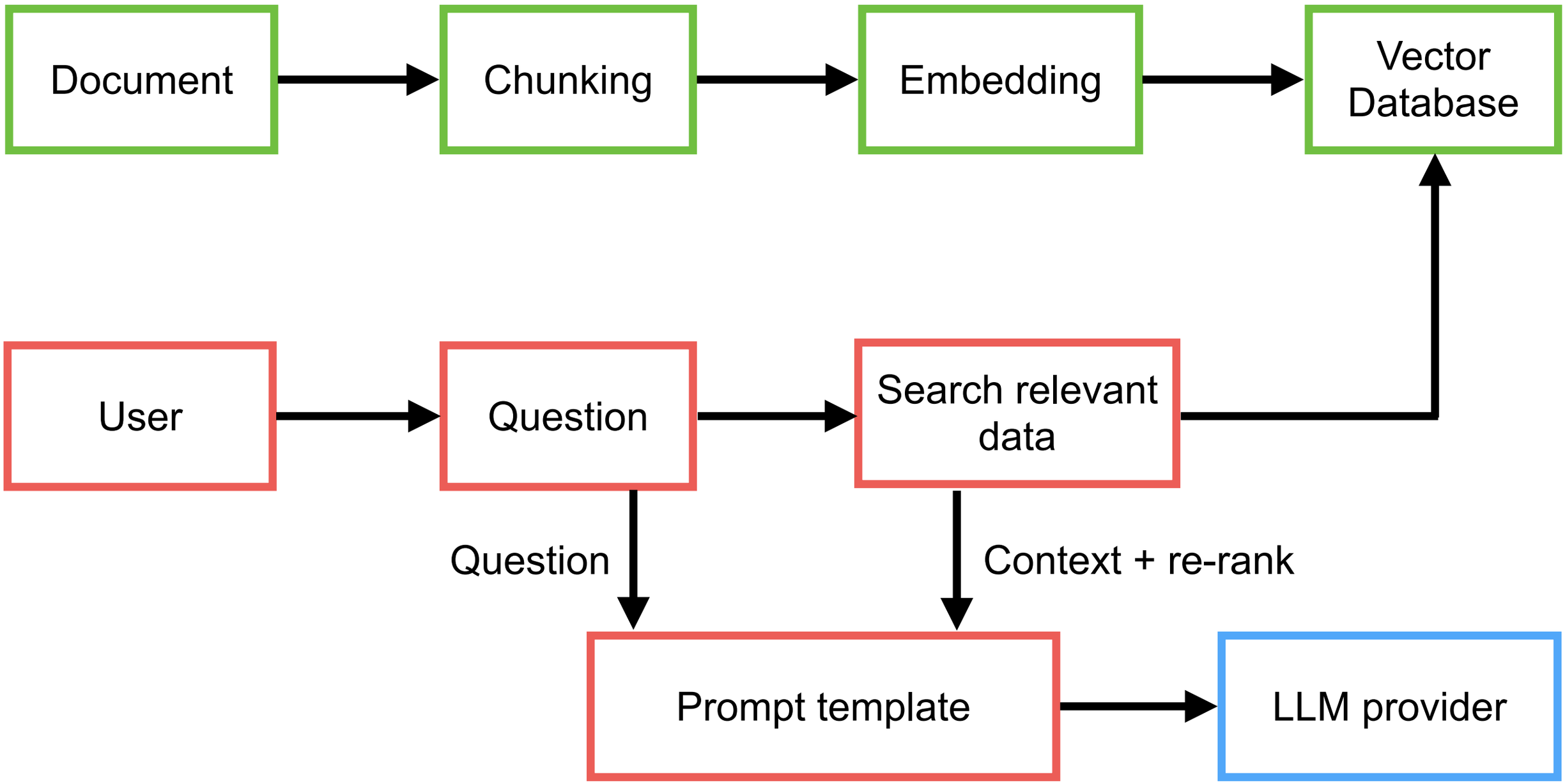
How to search/retrieve data ?

Data source	Data preparation
Search	Search result

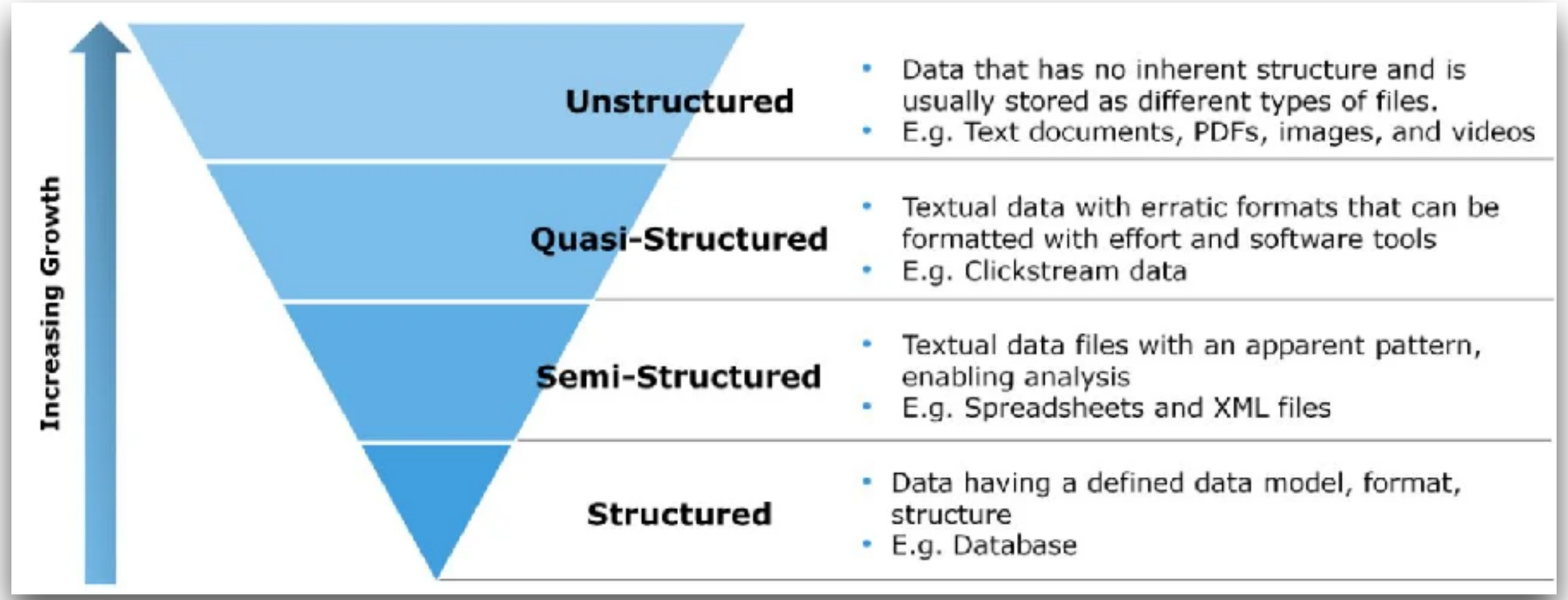
Basic RAG Architecture



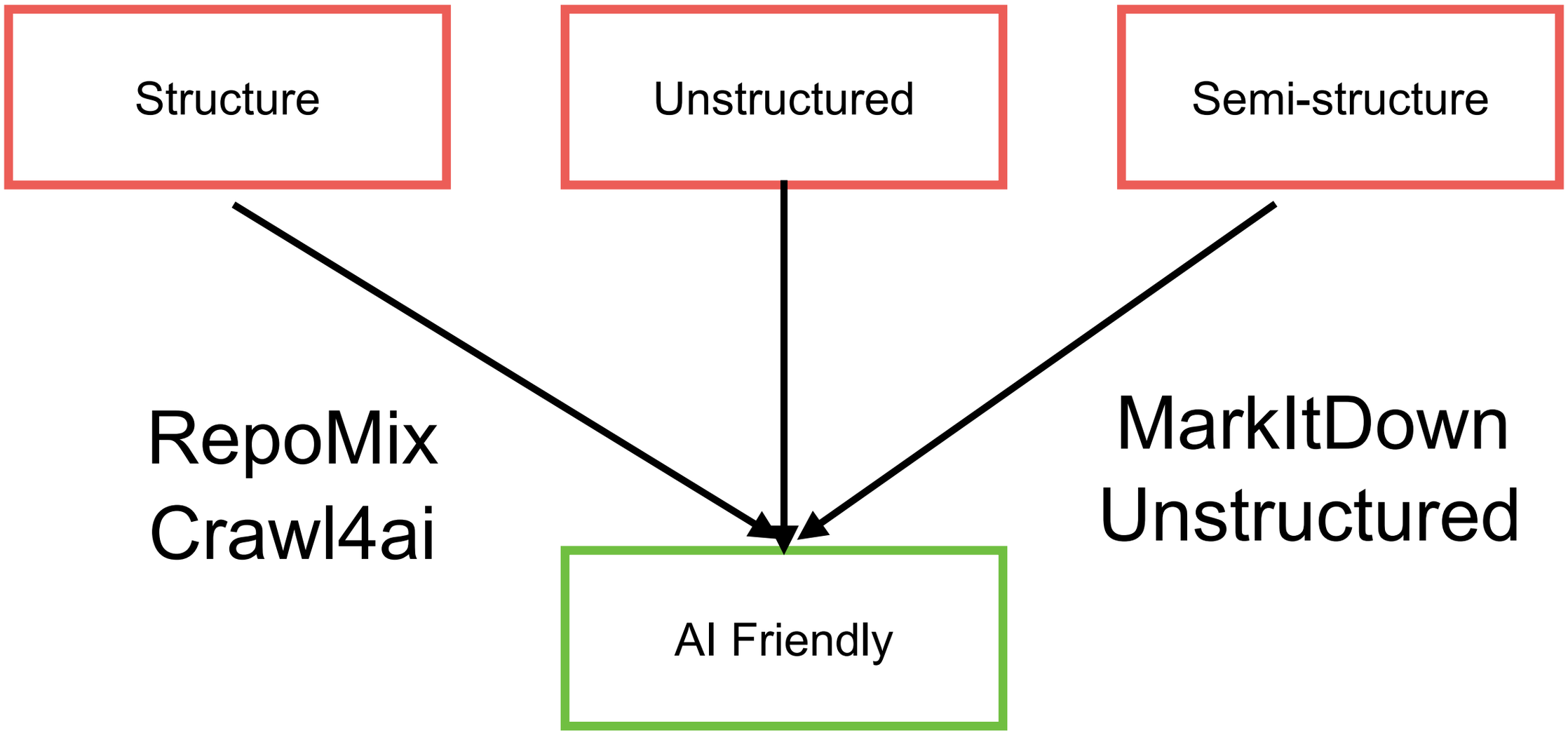
RAG process



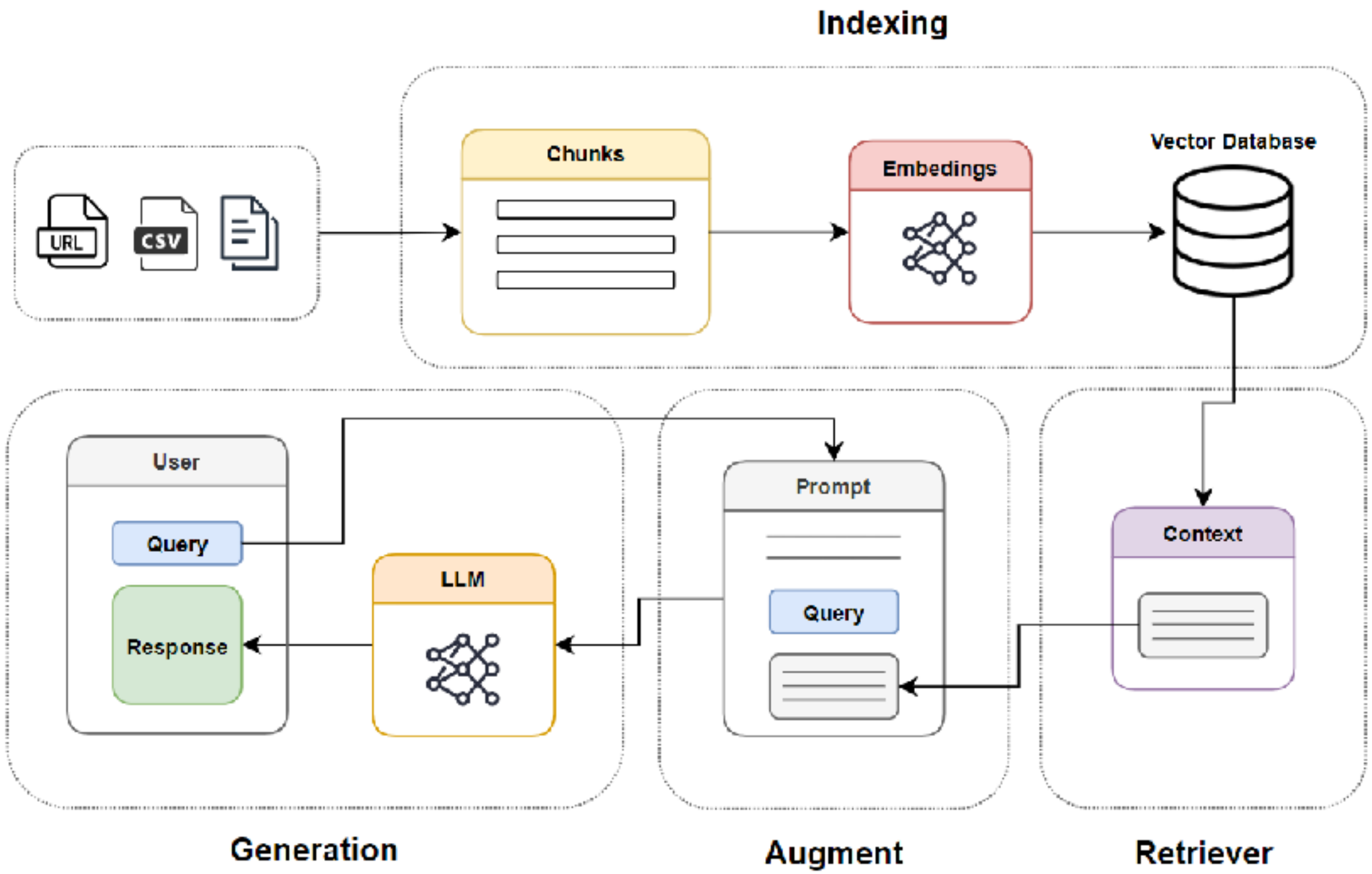
Structures of Data ?



Friendly Data for LLM/AI

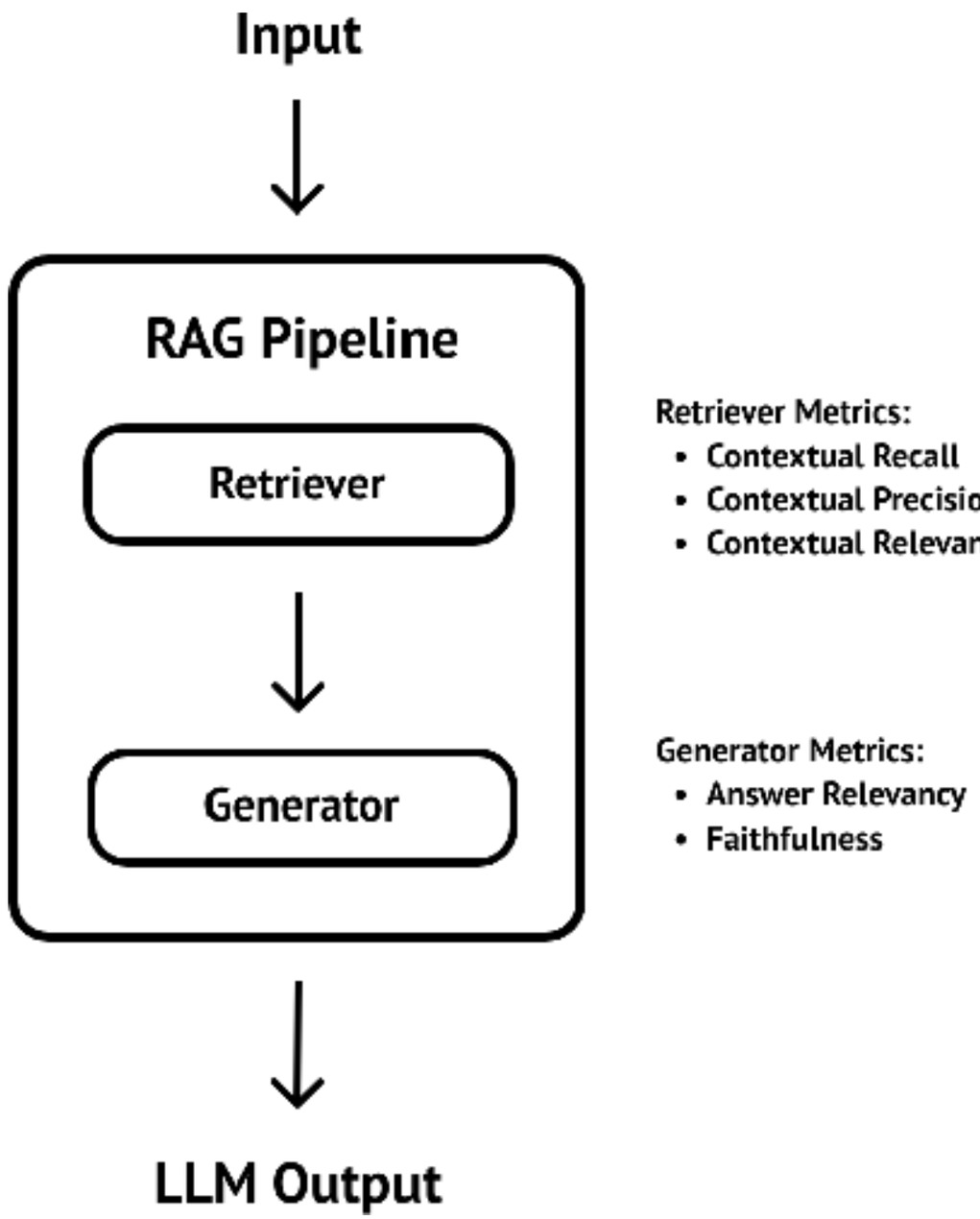


RAG Cookbook (techniques)

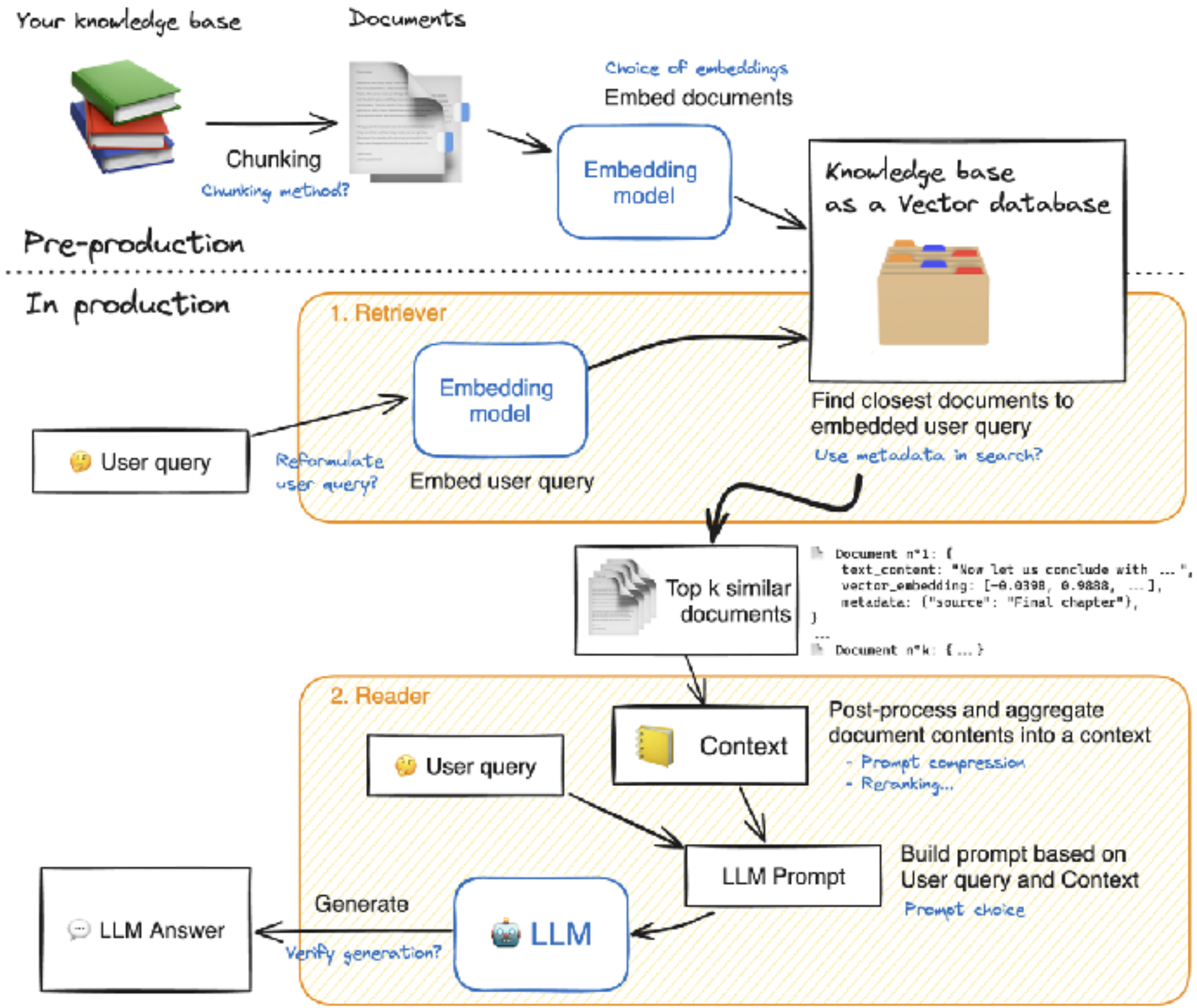


https://github.com/athina-ai/rag-cookbooks

RAG Evaluation !!

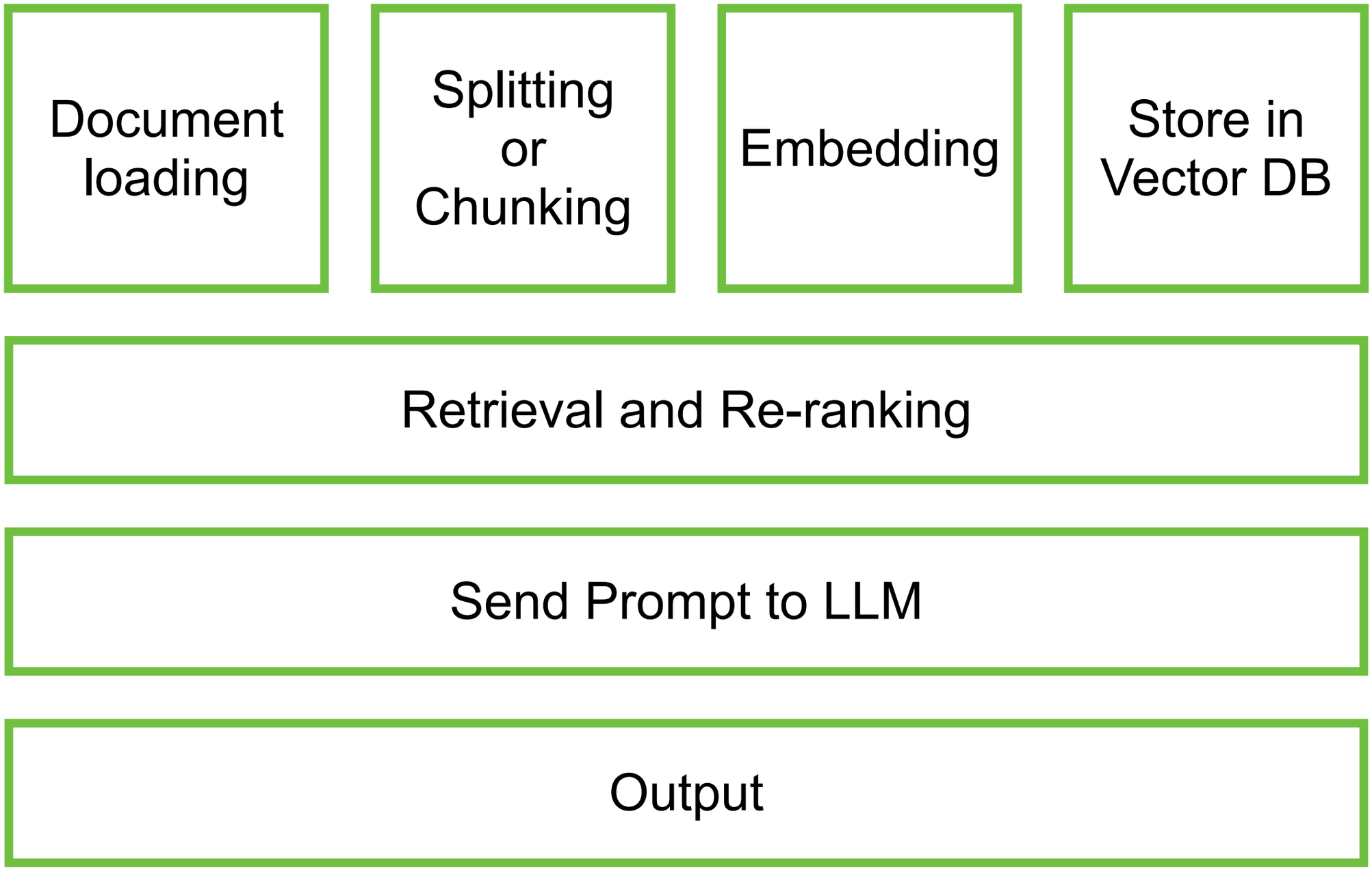


https://www.deepeval.com/guides/guides-rag-evaluation



https://huggingface.co/learn/cookbook/en/rag_evaluation

RAG Implementation



Failure Points of RAG

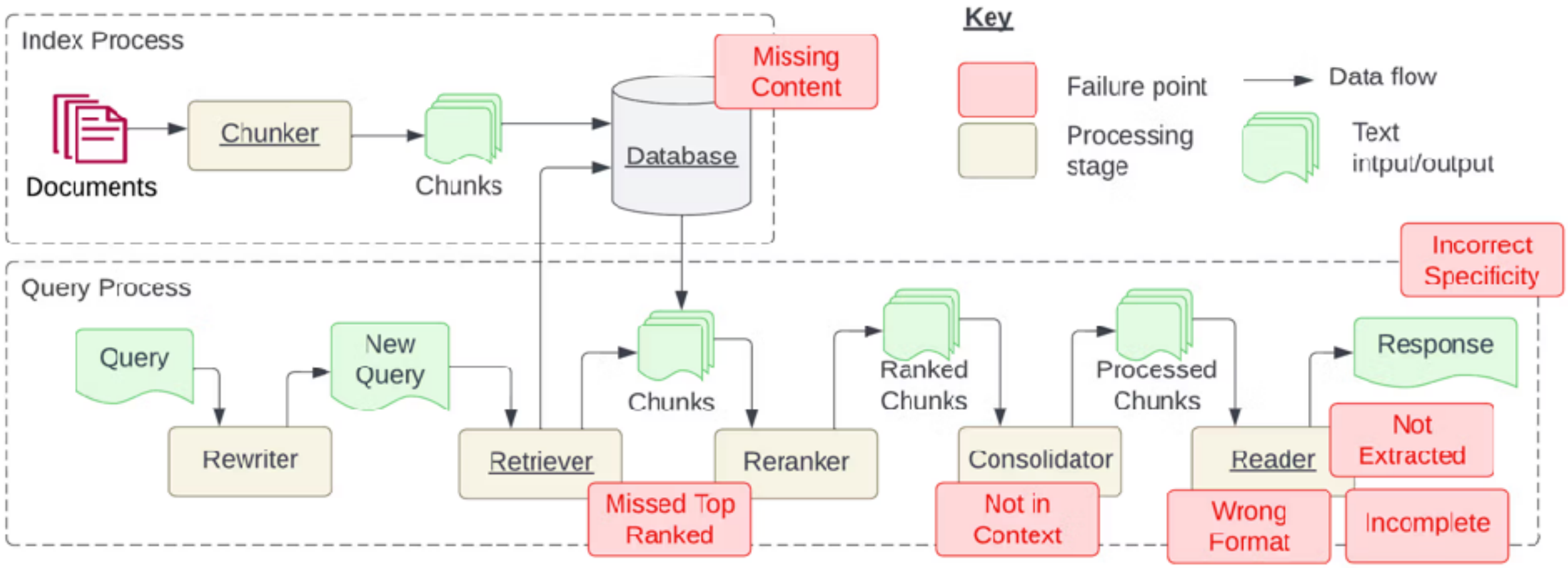


Figure 1: Indexing and Query processes required for creating a Retrieval Augmented Generation (RAG) system. The indexing process is typically done at development time and queries at runtime. Failure points identified in this study are shown in red boxes. All required stages are underlined. Figure expanded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

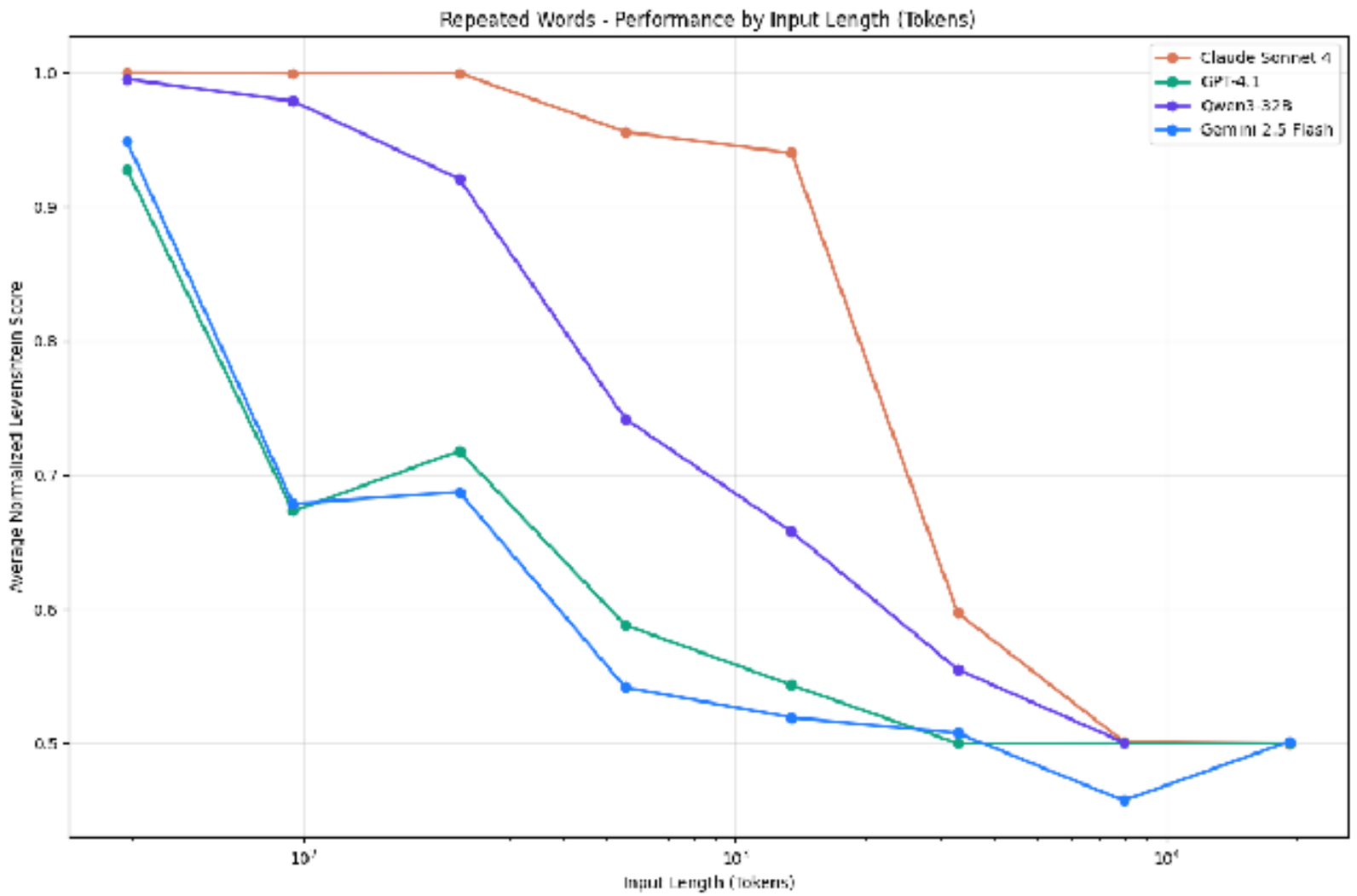
Chunk 1

Chunk 2

Chunk 3

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

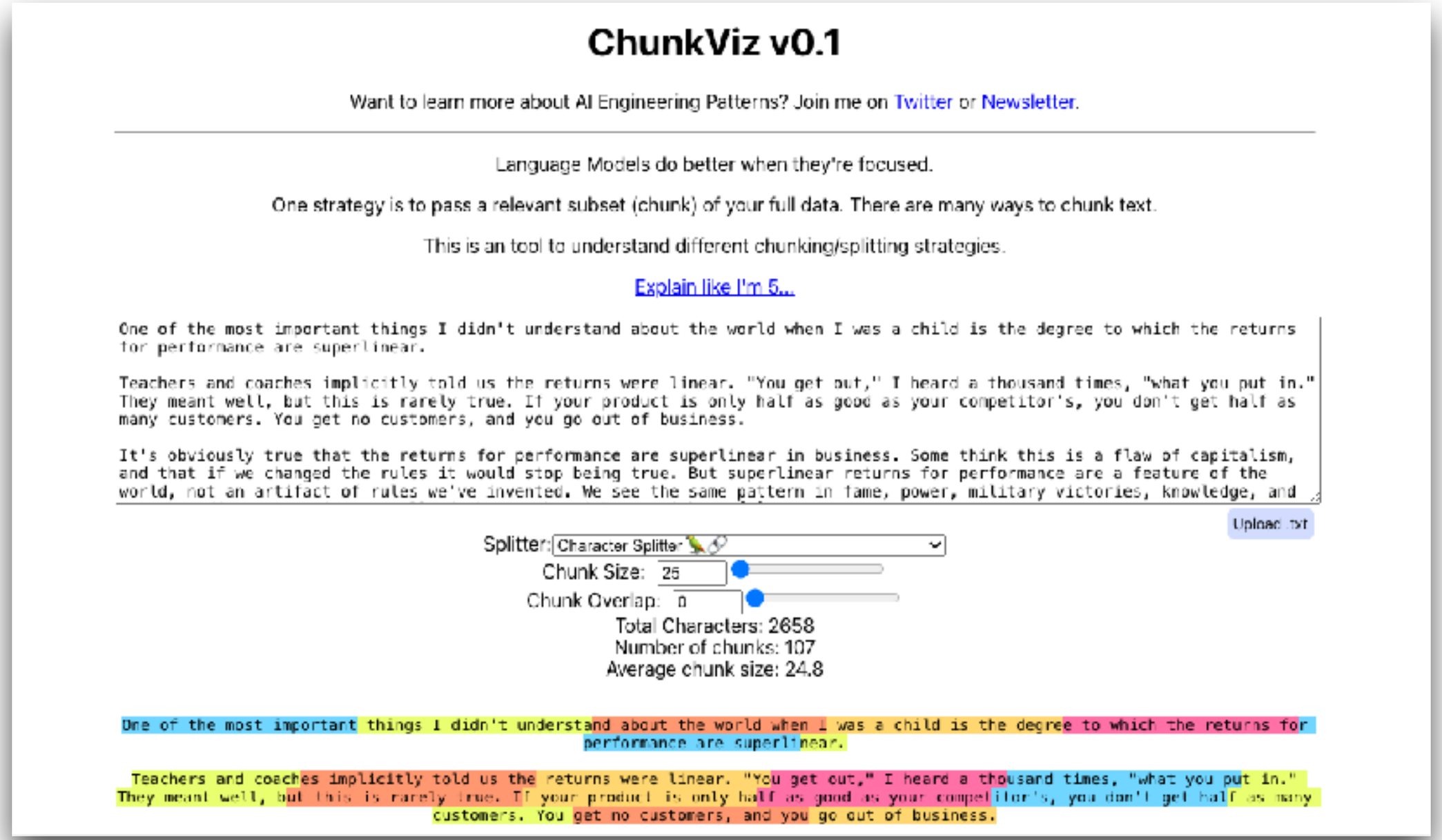
Bad Chunking

Loss context

Redundancy

Inconsistency

Chunking Visualization



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

Retrieval optimization ?

Re-ranking

Hybrid search

Query transformation

Multi-vector embedding

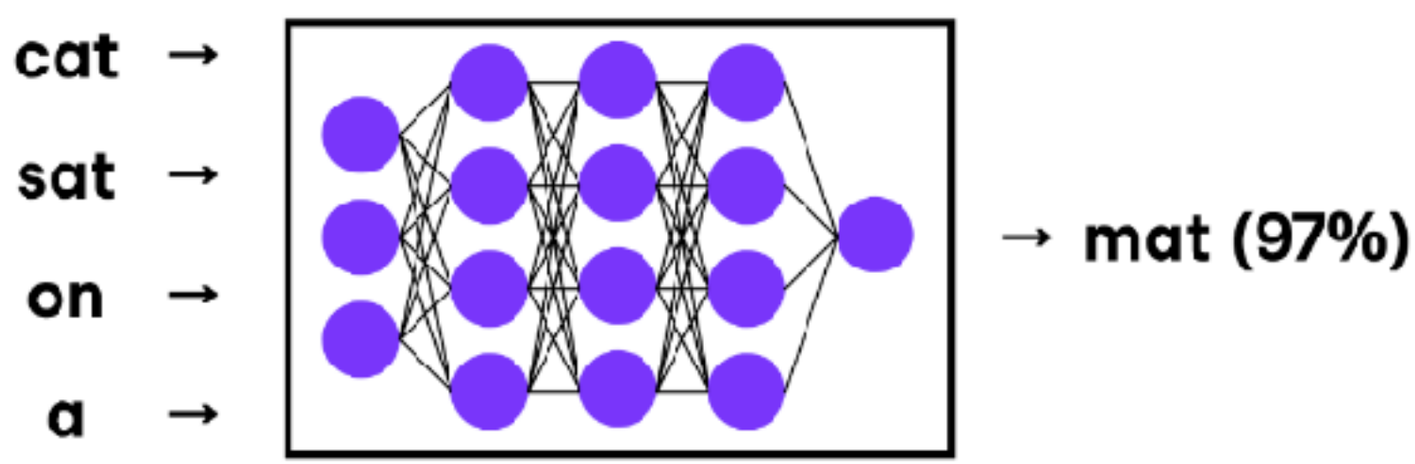
Contextual retrieval

Vector database Selection

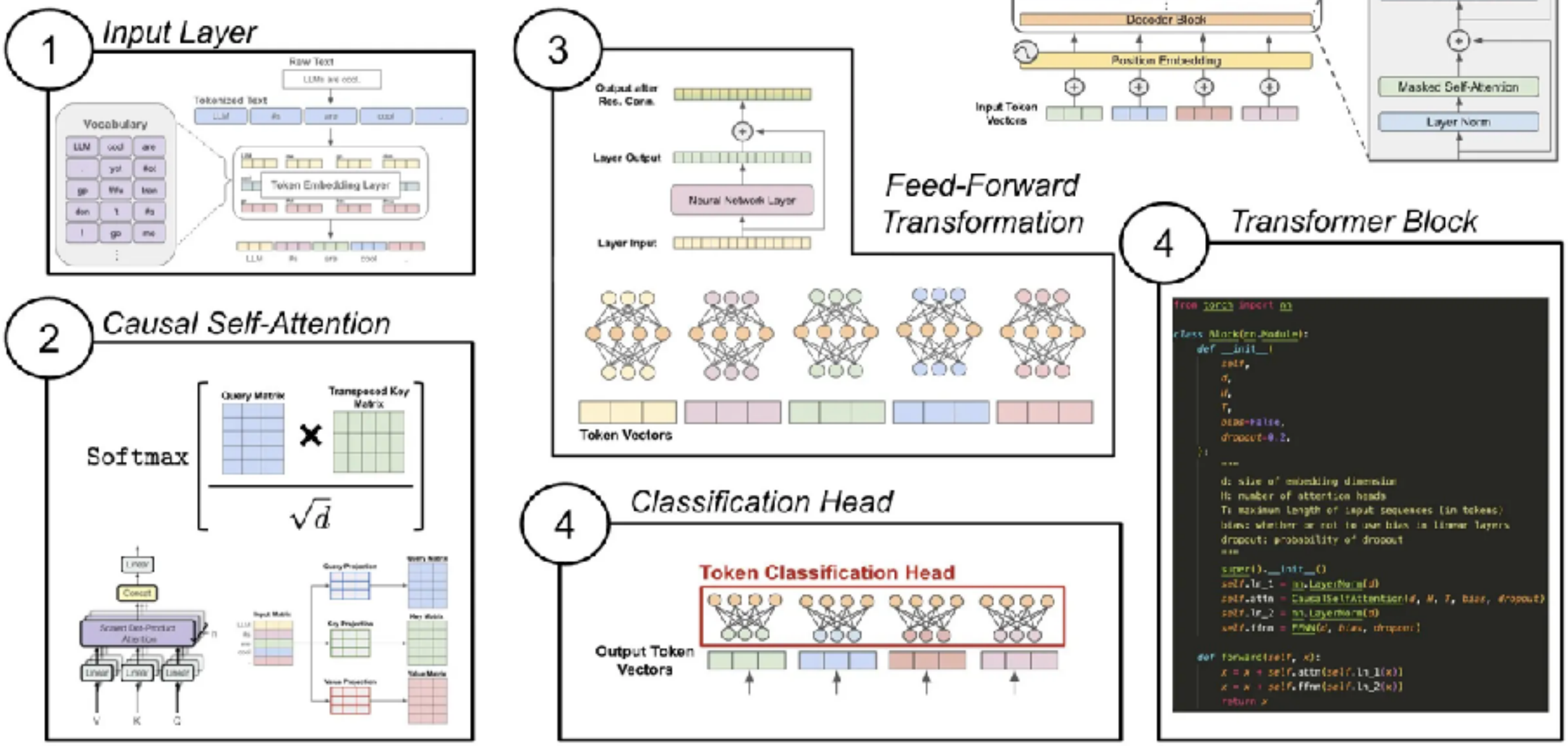
Large Language Model (LLM)

Neural network

Predicts the next word in a sequence

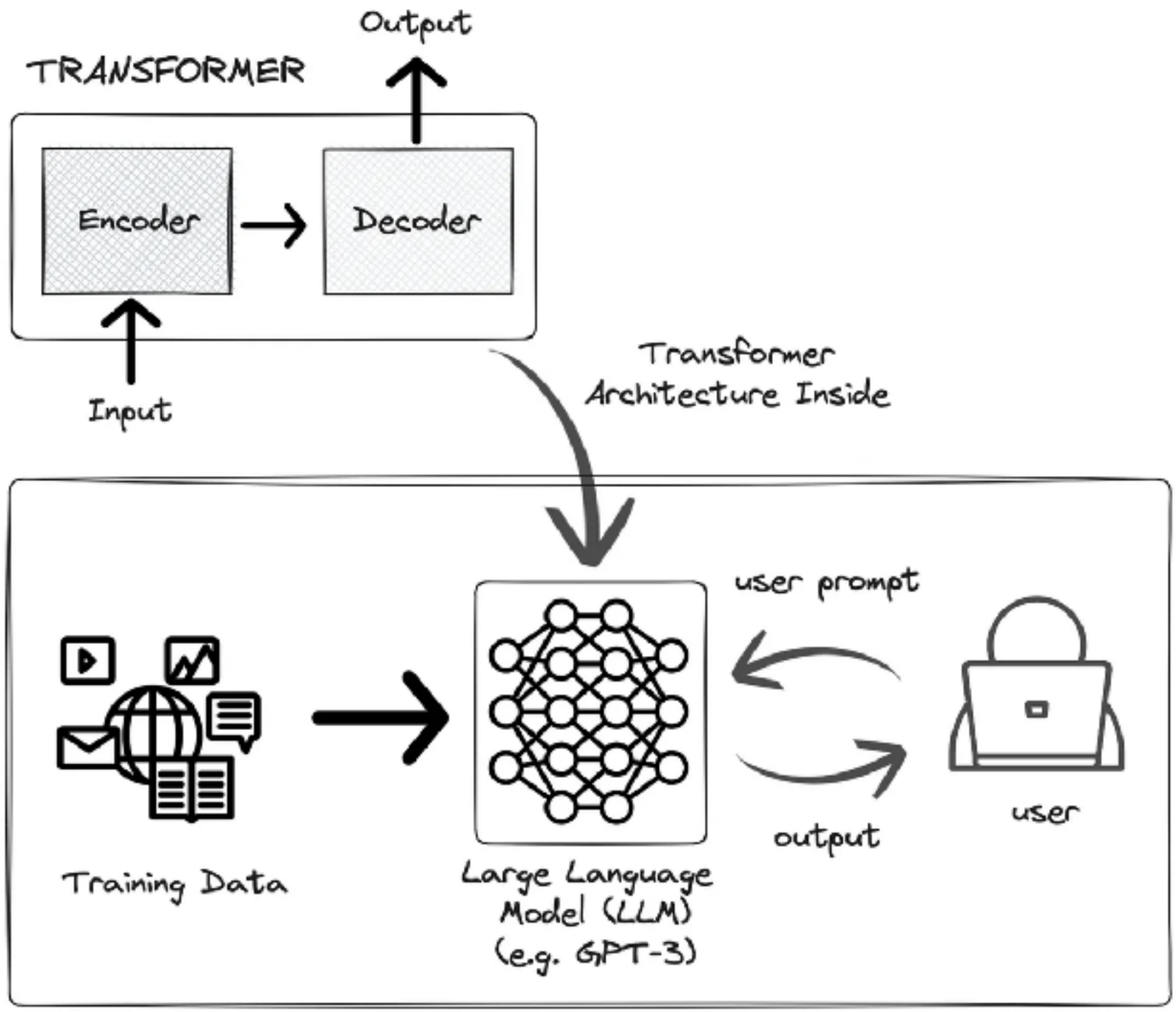


LLM components !!

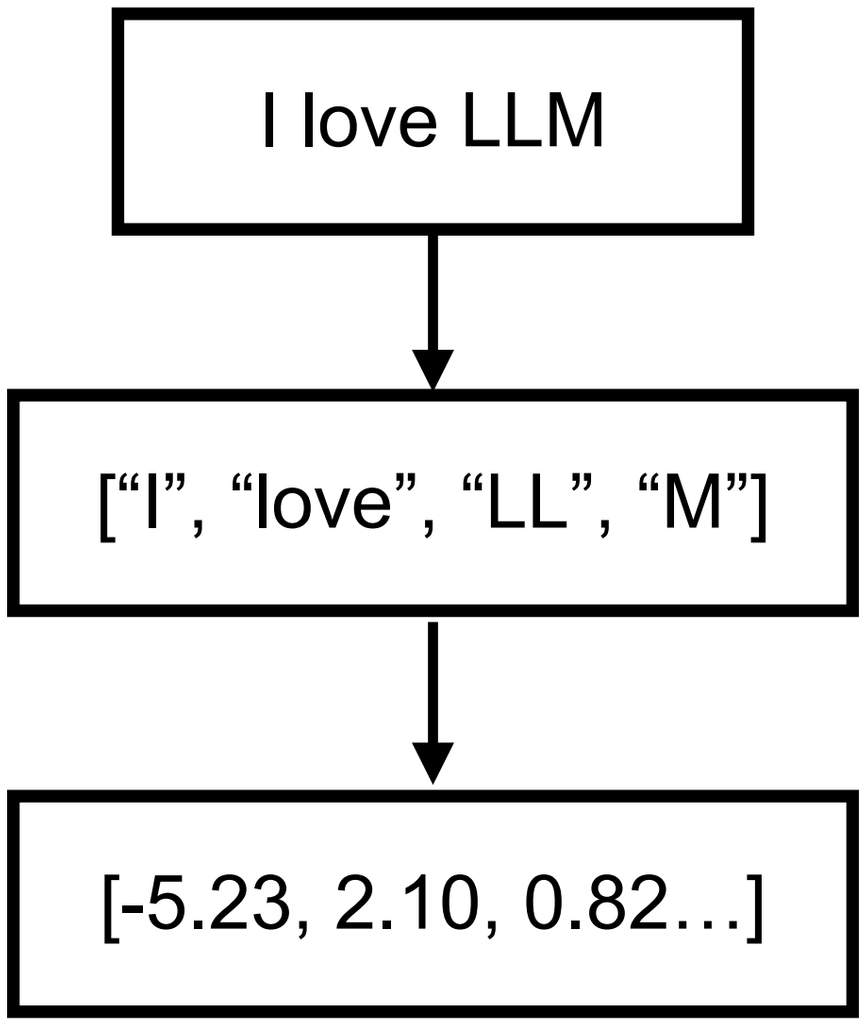


https://stackoverflow.blog/2024/08/22/llms-evolve-quickly-their-underlying-architecture-not-so-much

Transformer inside



Transformer process



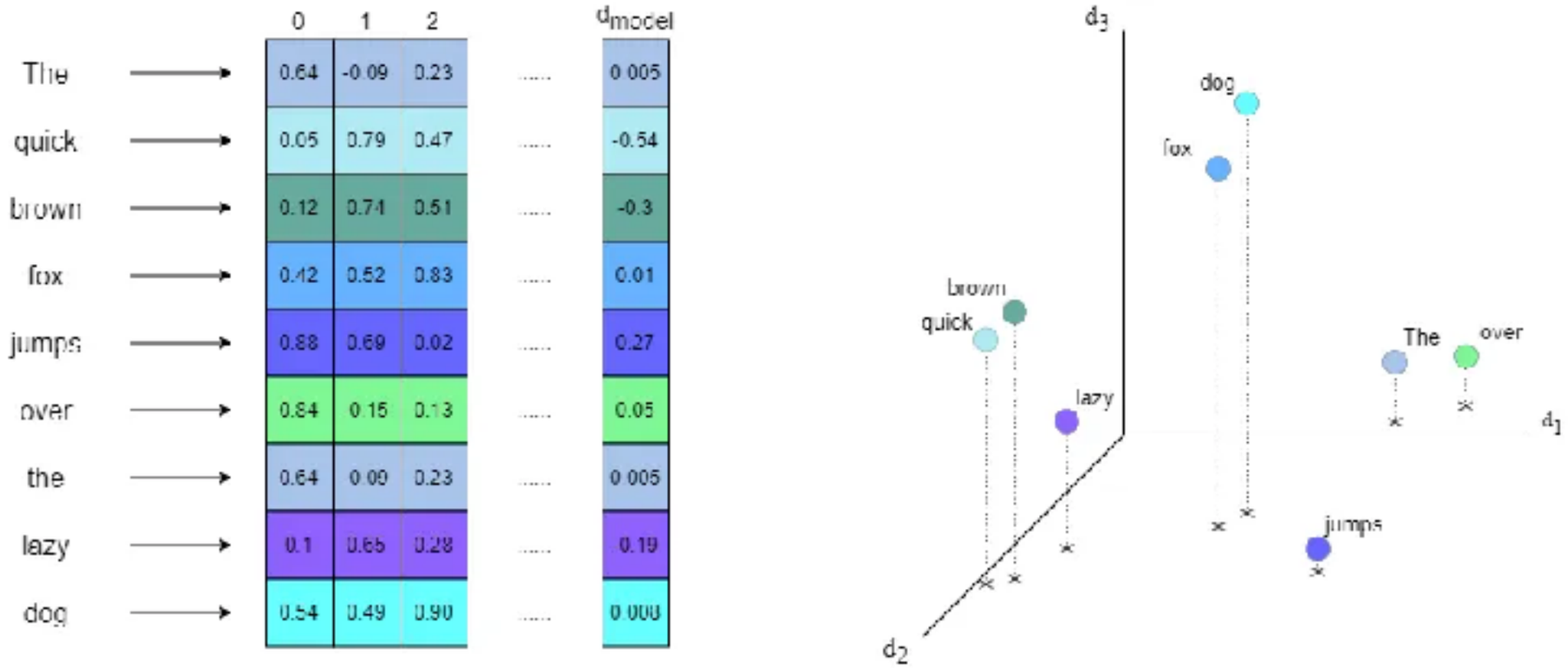
Sentence

Tokens

Embedding

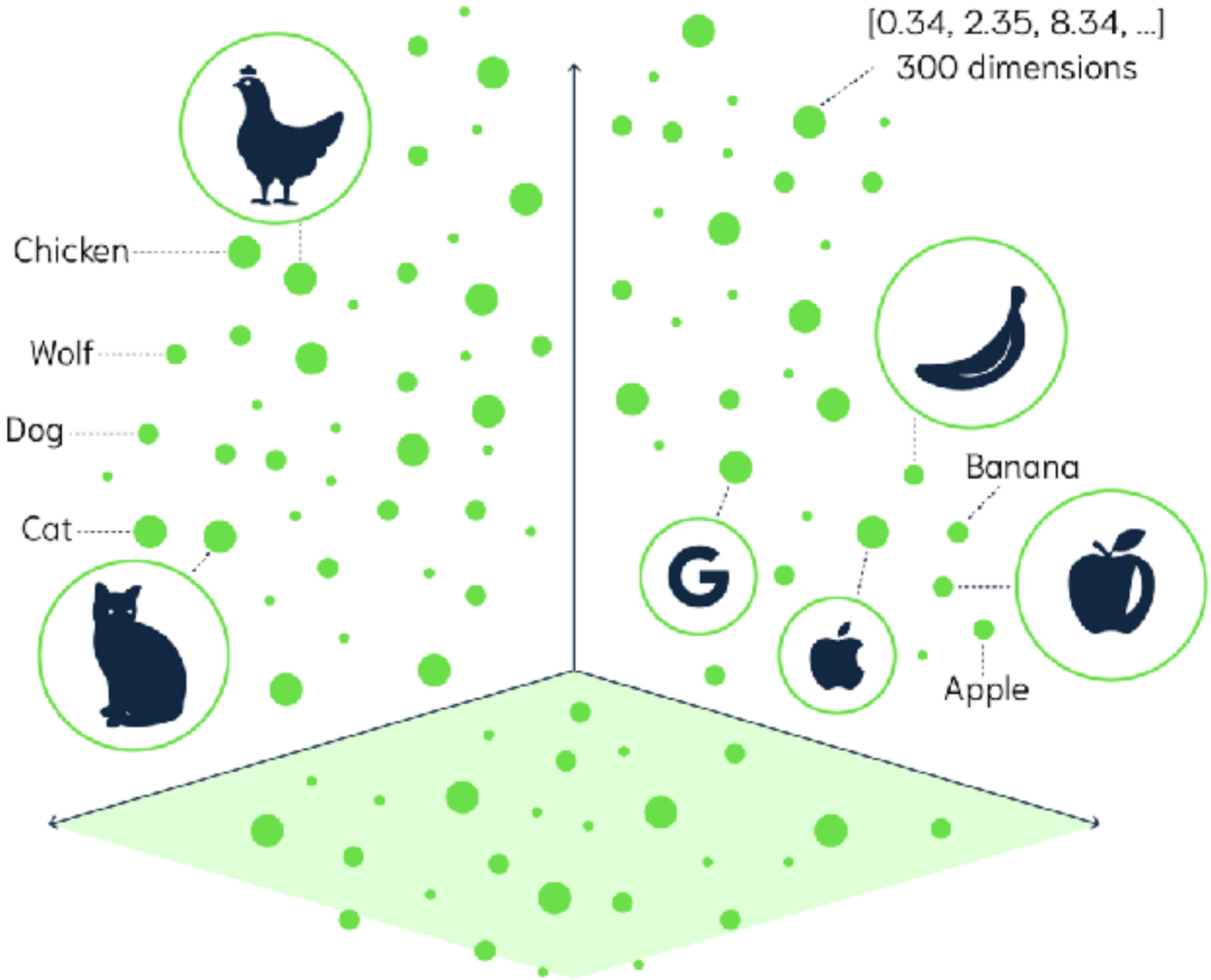
Embedding ?

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

Visual of Vector space

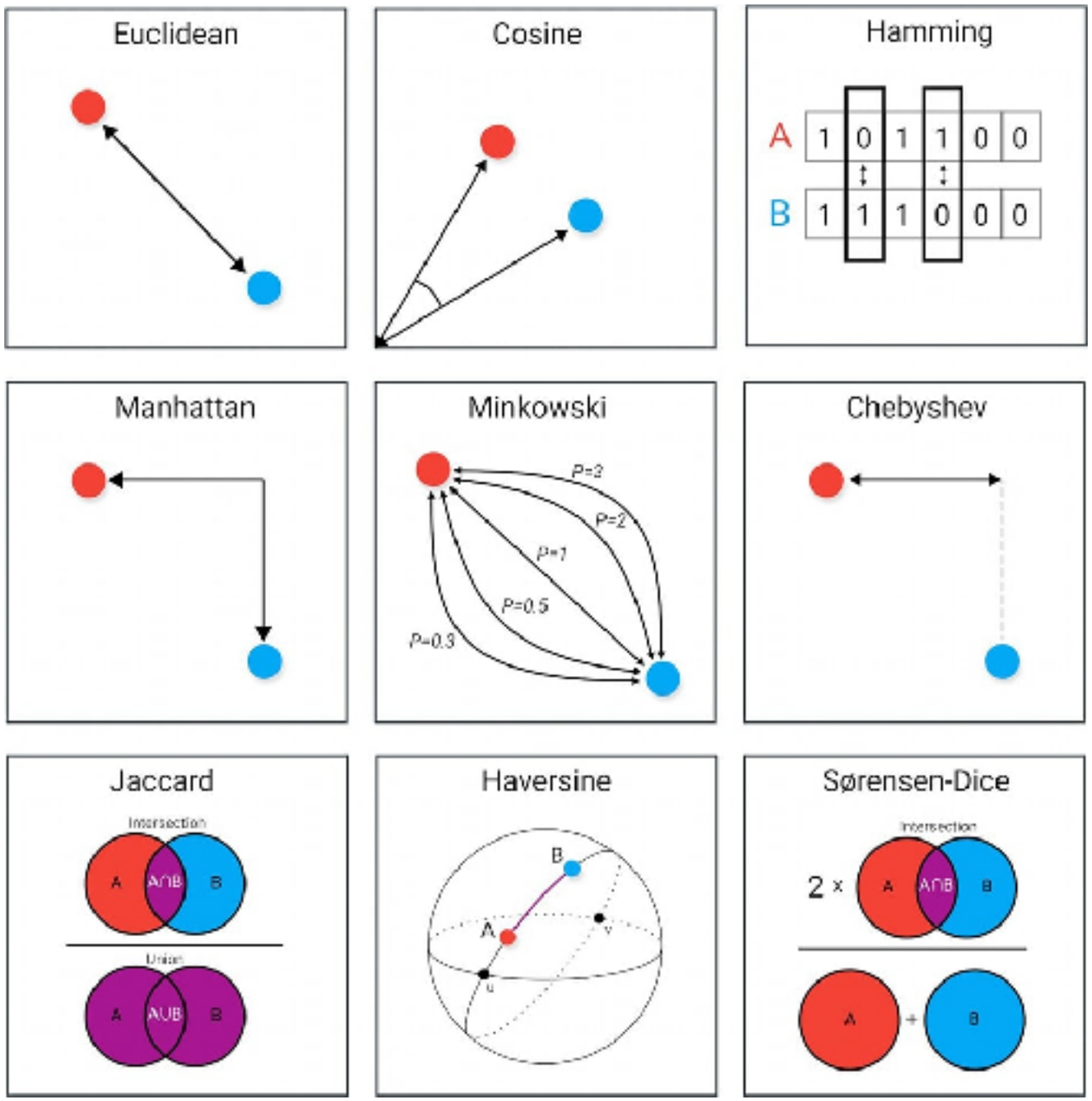


Embedding Leaderboard

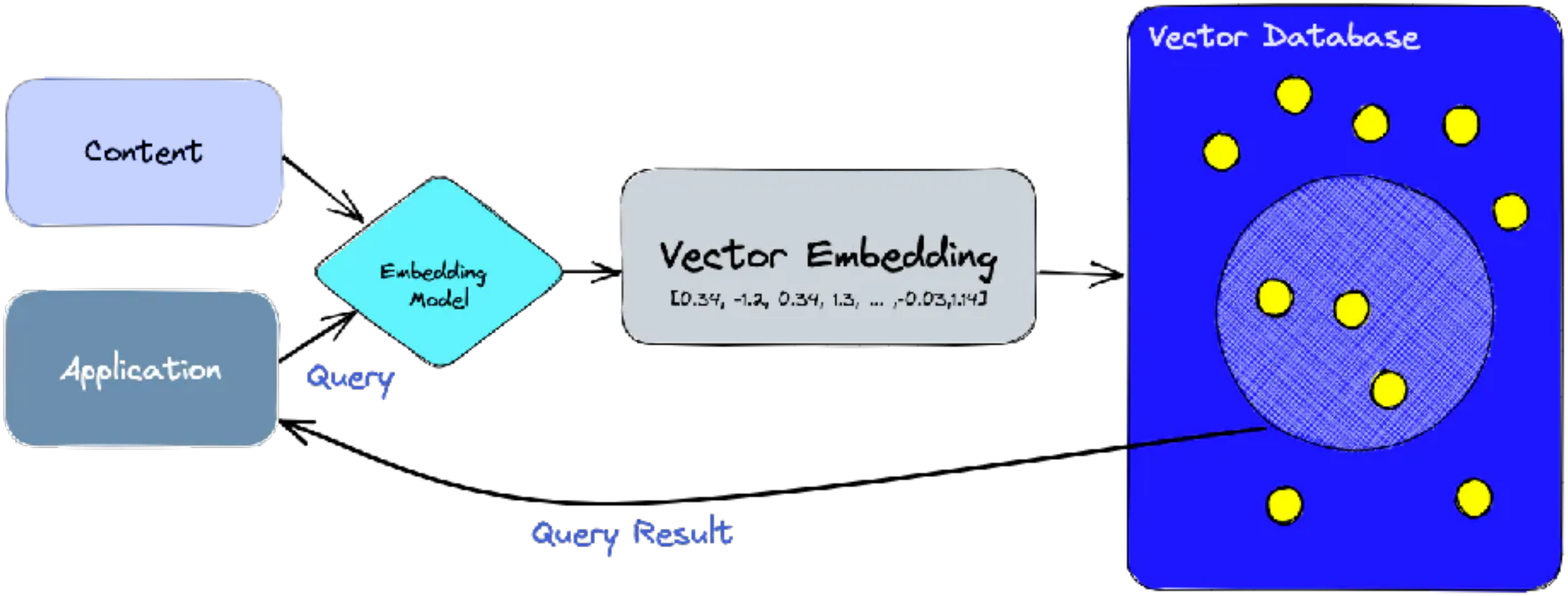
Rank (Bor…	Model	Zero-shot	Memory U…	Number of P…	Embedding D…	Max Tokens	Mean (T…	Mean (TaskT…	Bitext …	Classific
1	llama-embed-nemotron-8b	99%	28629	7B	4096	32768	69.46	61.09	**81.72**	73.21
2	Qwen3-Embedding-8B	99%	28866	7B	4096	32768	**70.58**	**61.69**	80.89	**74.00**
3	gemini-embedding-001	99%	Unknown	Unknown	3072	2048	68.37	59.59	79.28	71.82
4	Qwen3-Embedding-4B	99%	15341	4B	2560	32768	69.45	60.86	79.36	72.33
5	Qwen3-Embedding-0.6B	99%	2272	595M	1024	32768	64.34	56.01	72.23	66.83
6	gte-Qwen2-7B-instruct	⚠ NA	29040	7B	3584	32768	62.51	55.93	73.92	61.55
7	Linq-Embed-Mistral	99%	13563	7B	4096	32768	61.47	54.14	70.34	62.24
8	multilingual-e5-large-instruct	99%	1068	560M	1024	514	63.22	55.08	80.13	64.94
9	embeddinggemma-300m	99%	578	307M	768	2048	61.15	54.31	64.40	60.90
10	SFR-Embedding-Mistral	96%	13563	7B	4096	32768	60.90	53.92	70.00	60.02
11	text-multilingual-embedding-002	99%	Unknown	Unknown	768	2048	62.16	54.25	70.73	64.64

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

Distance measure in Data Science

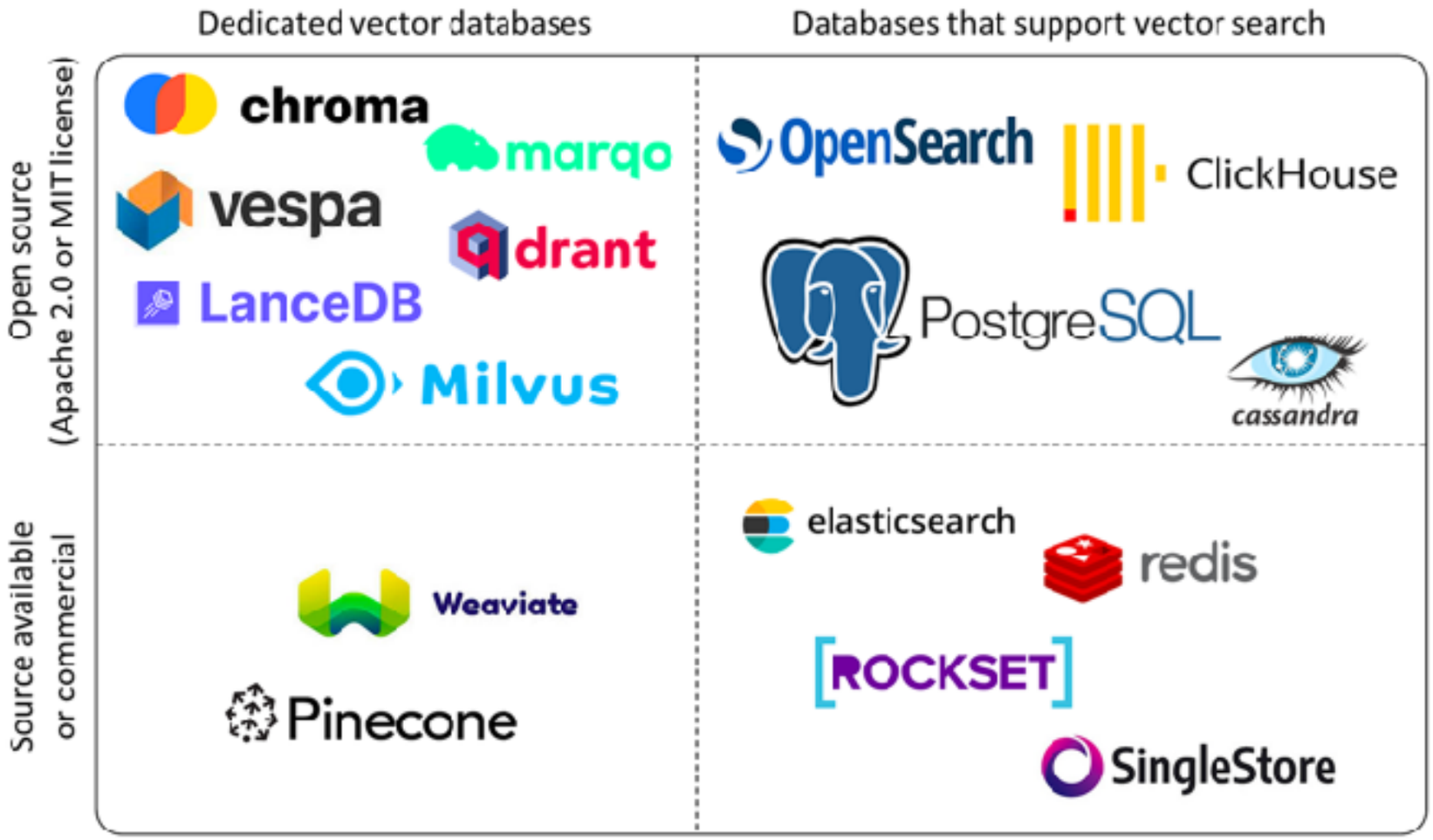


Store data in Vector Database



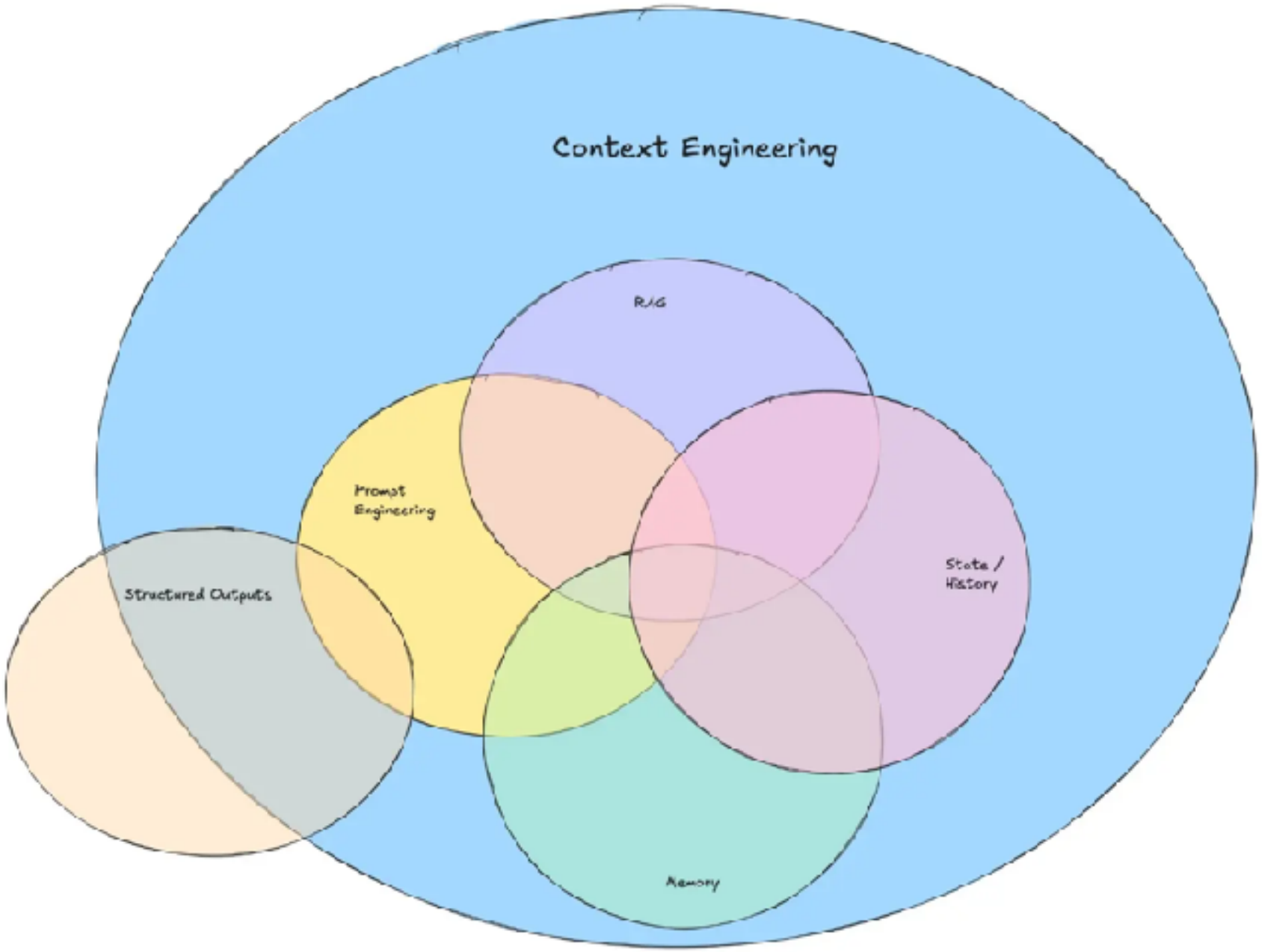
Vector Database ?

Map items of unstructured data to high-dimensional real **vectors**



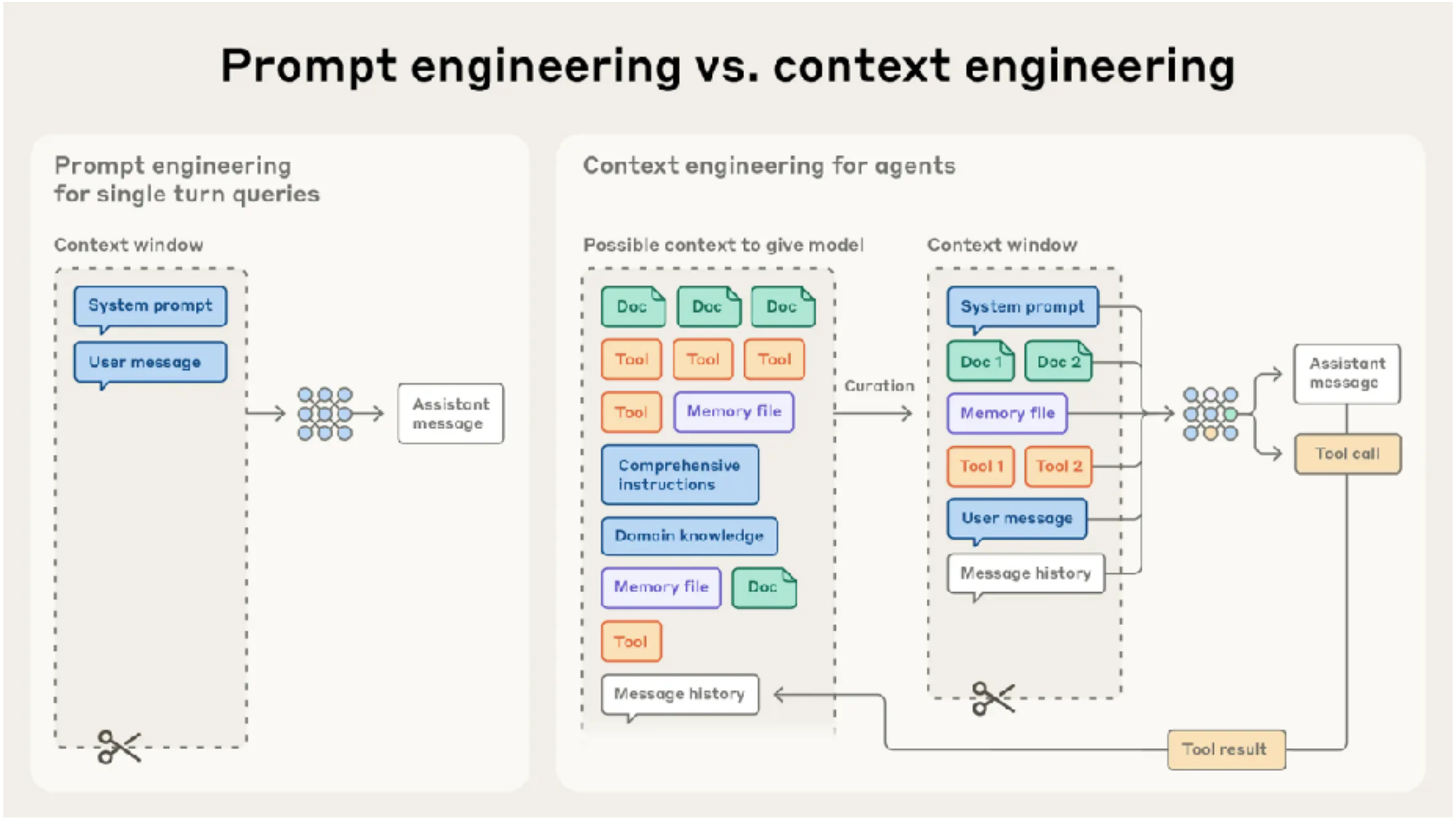
https://towardsdatascience.com/transformers-in-depth-part-1-introduction-to-transformer-models-in-5-minutes-ad25da6d3cca

Context Engineering



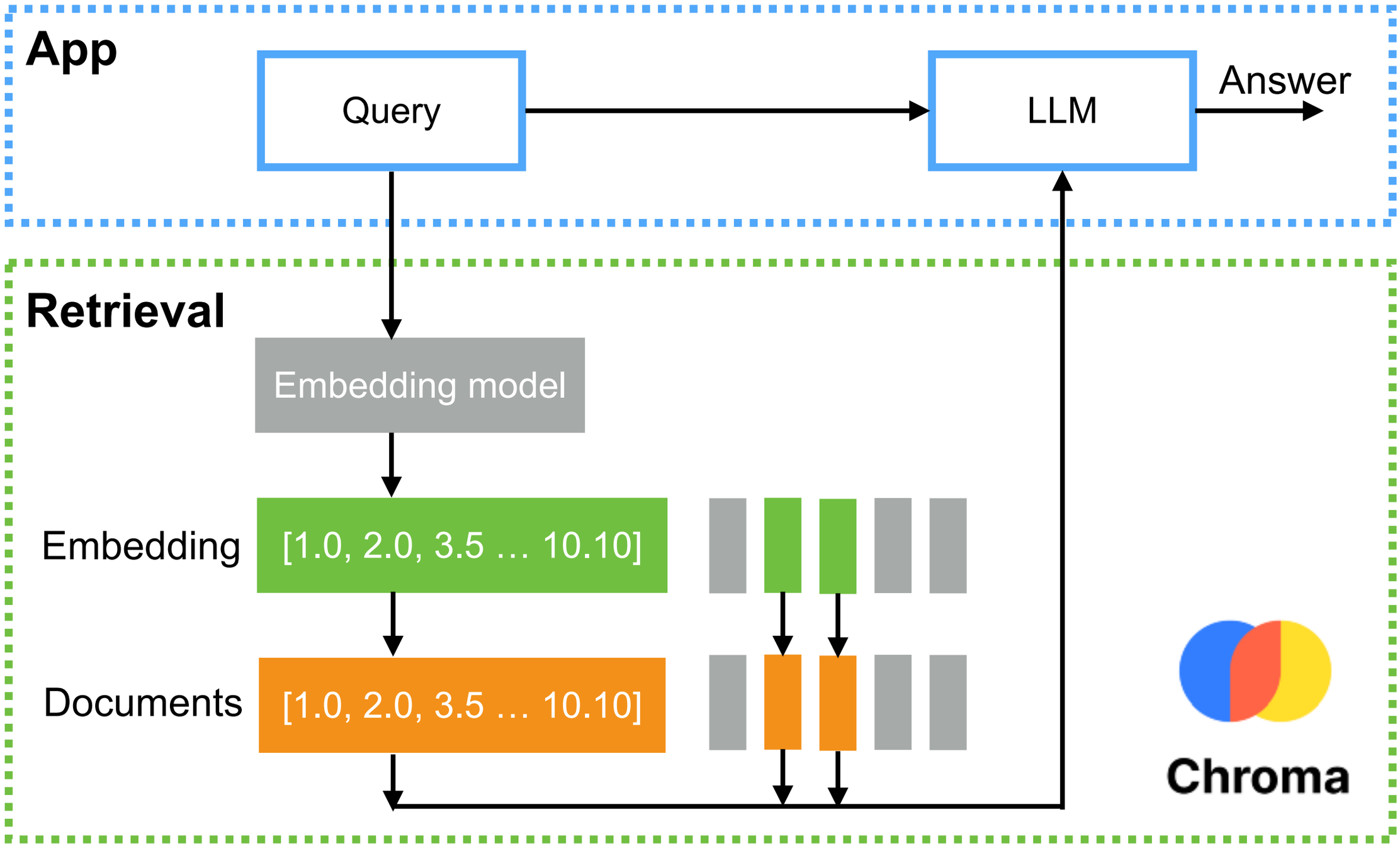
https://www.promptingguide.ai/guides/context-engineering-guide

Effective Context Engineering



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

Basic RAG



Query Expansion

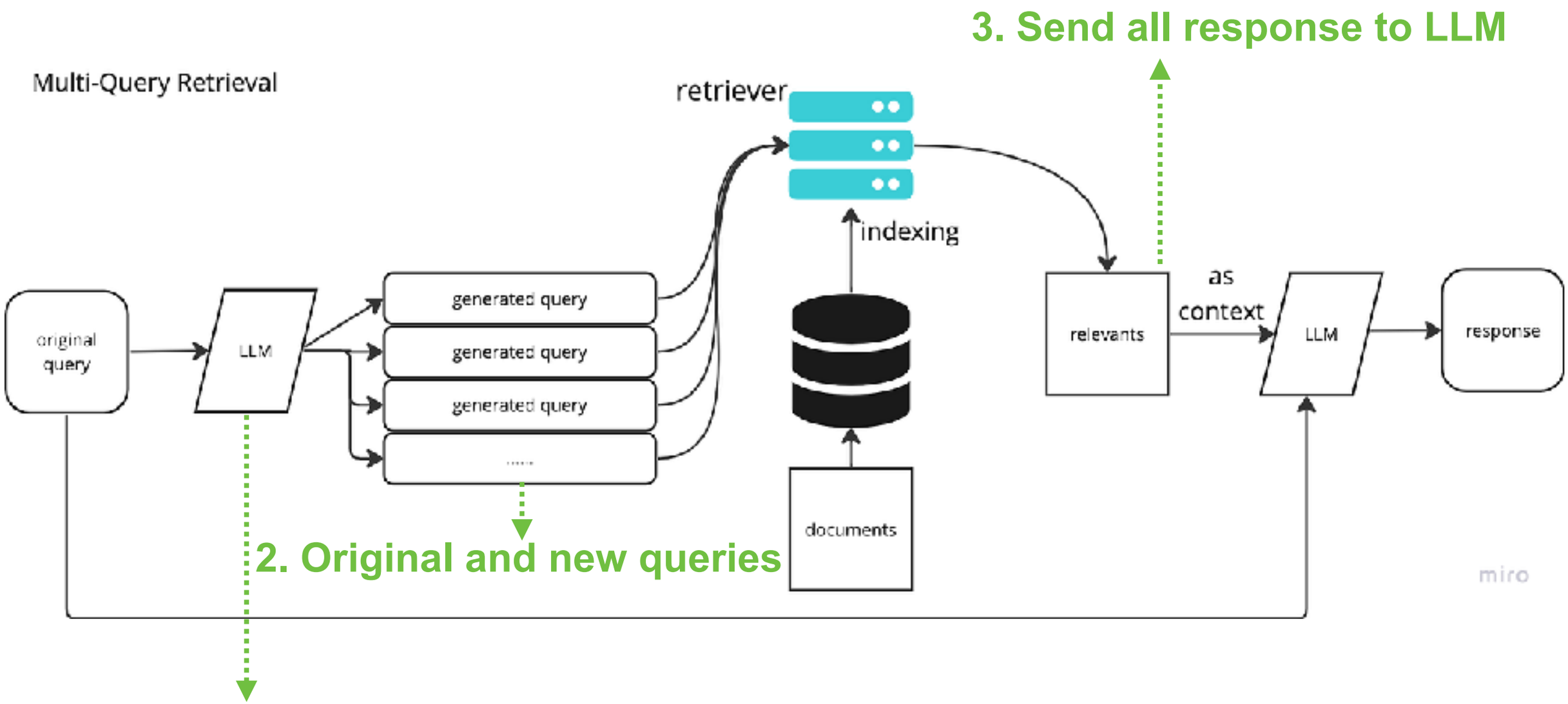
Query expansion is a widely used technique to improve the recall of search systems

Ambiguity	Vocabulary mismatch	Lack of context

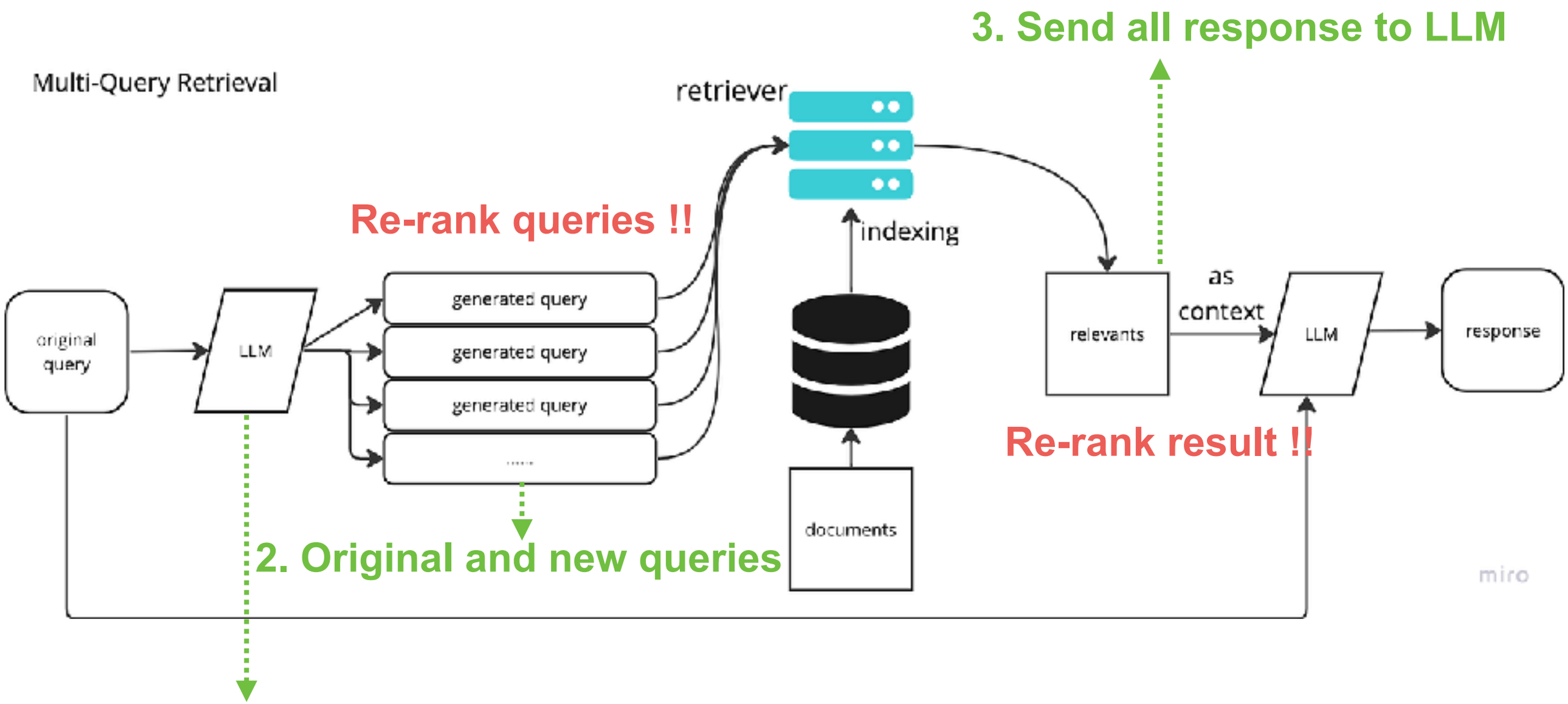
"Add synonyms, related context and contextual terms"

https://arxiv.org/abs/2305.03653

Query Expansion

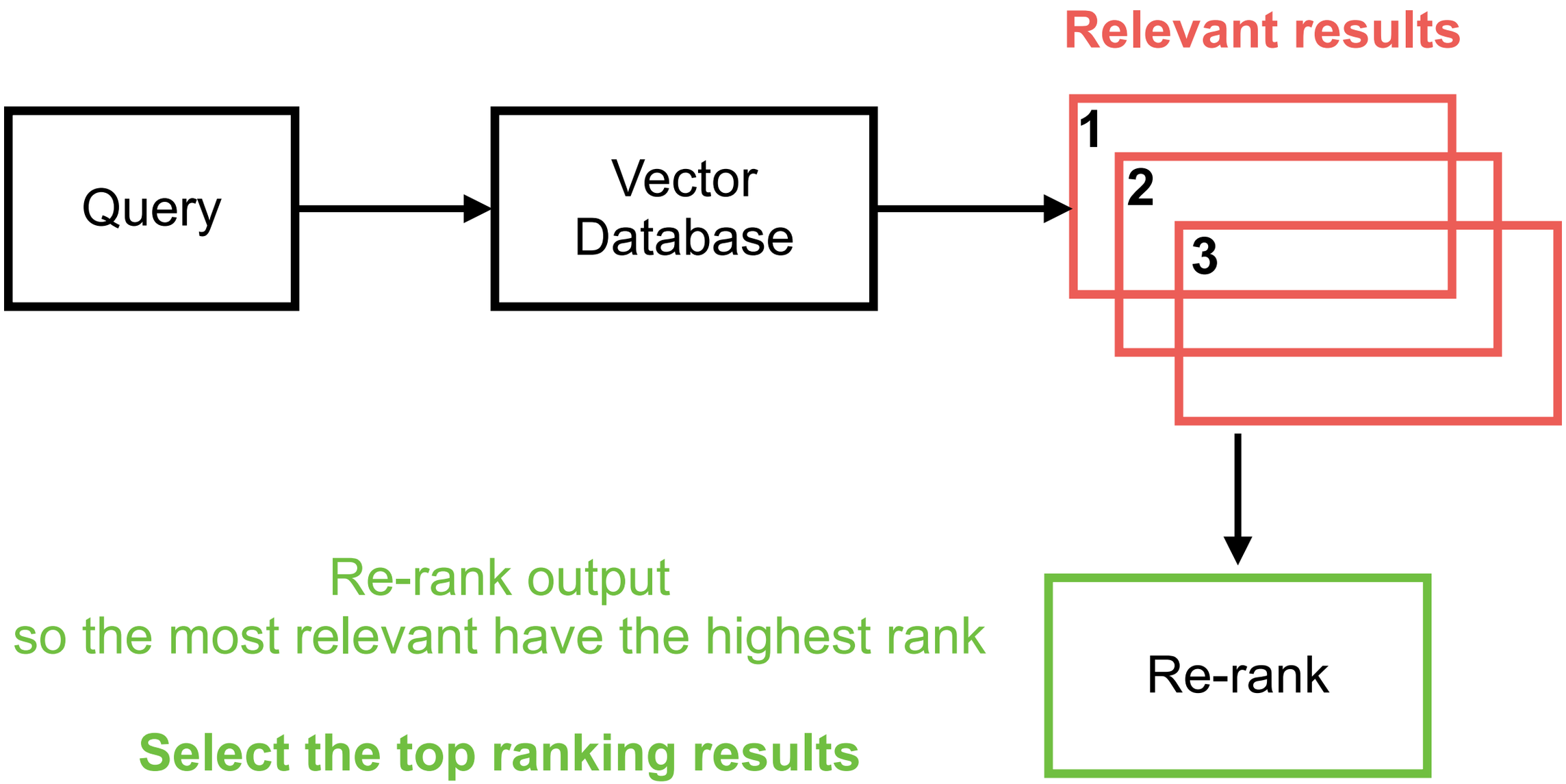


Re-rank !!



Cross-encoder Re-ranking

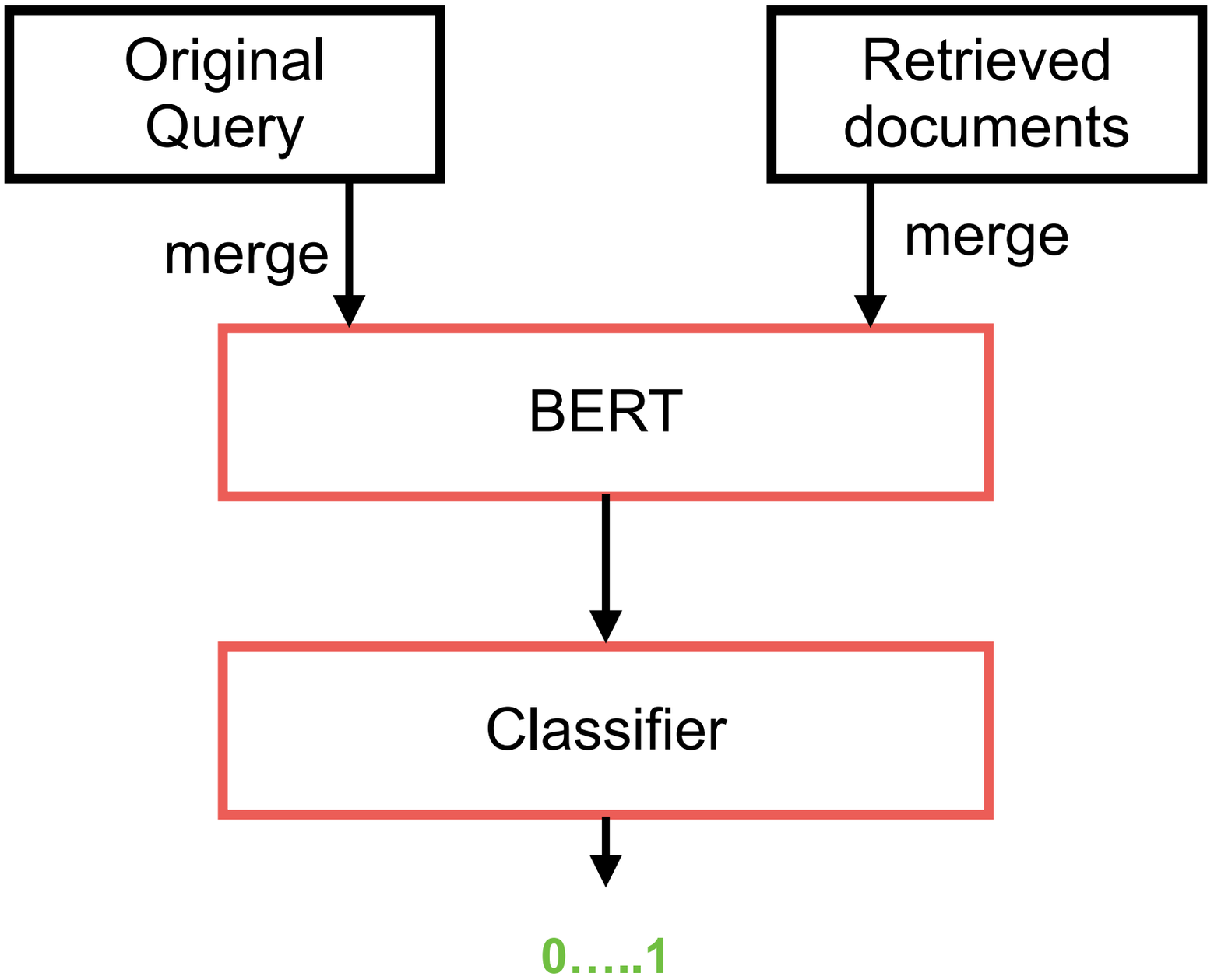
How to ordering relevant results !!



Re-rank output
so the most relevant have the highest rank

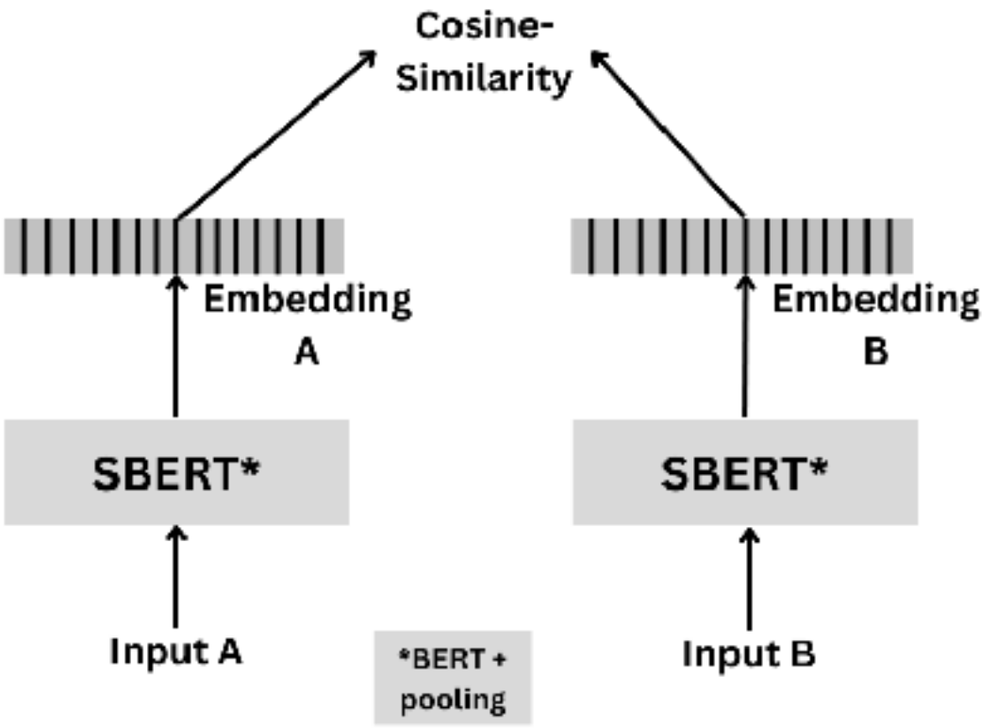
Select the top ranking results

Cross-Encoder

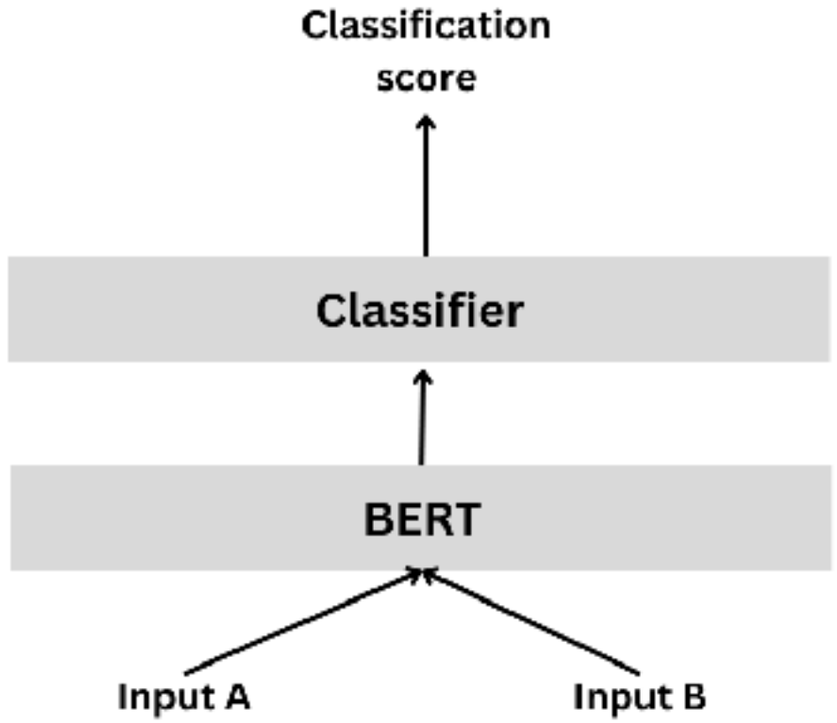


Cross-Encoder !!

Bi-encoder

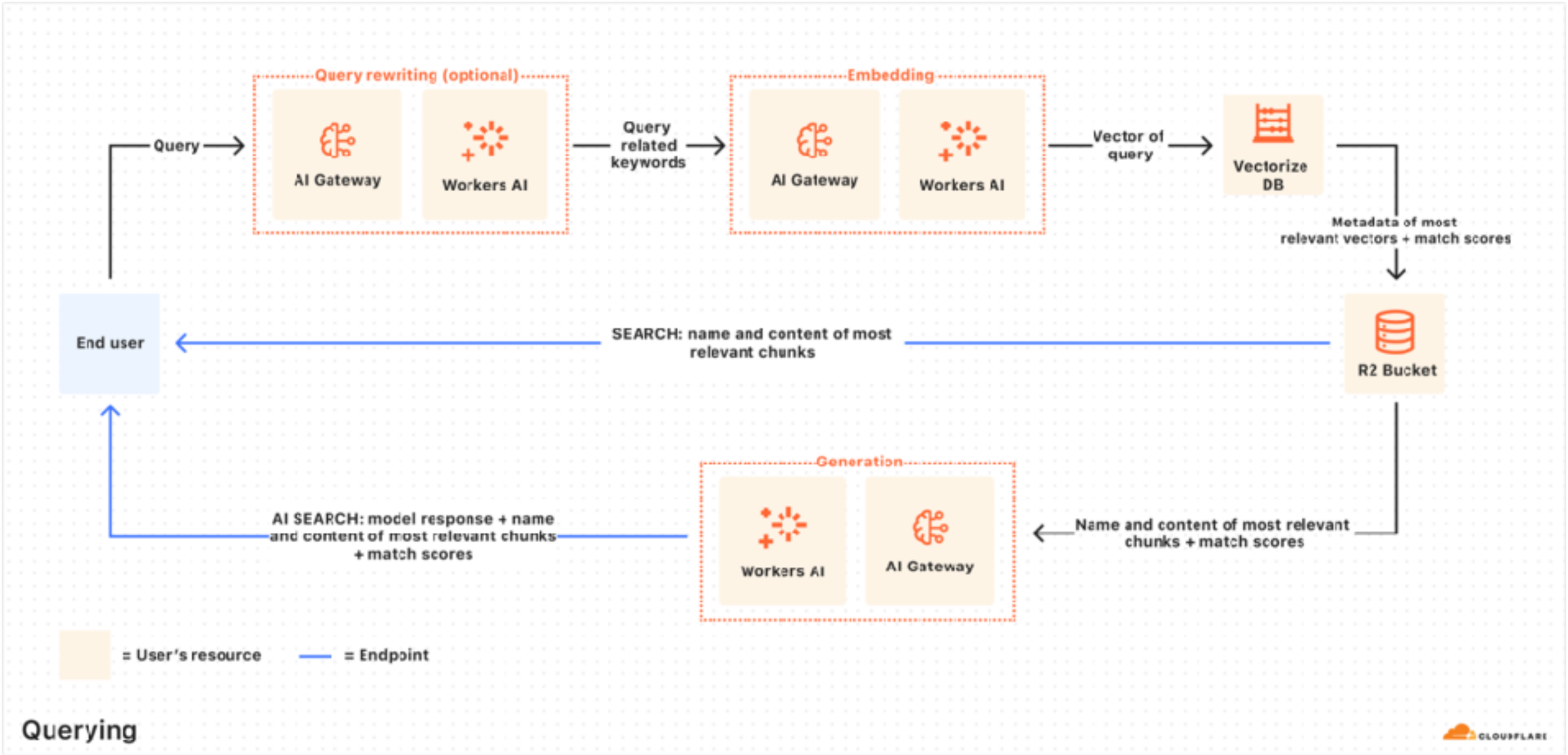


Cross Encoder



https://sbert.net/examples/cross_encoder/applications/README.html

Cloudflare AutoRAG



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

Local LLM

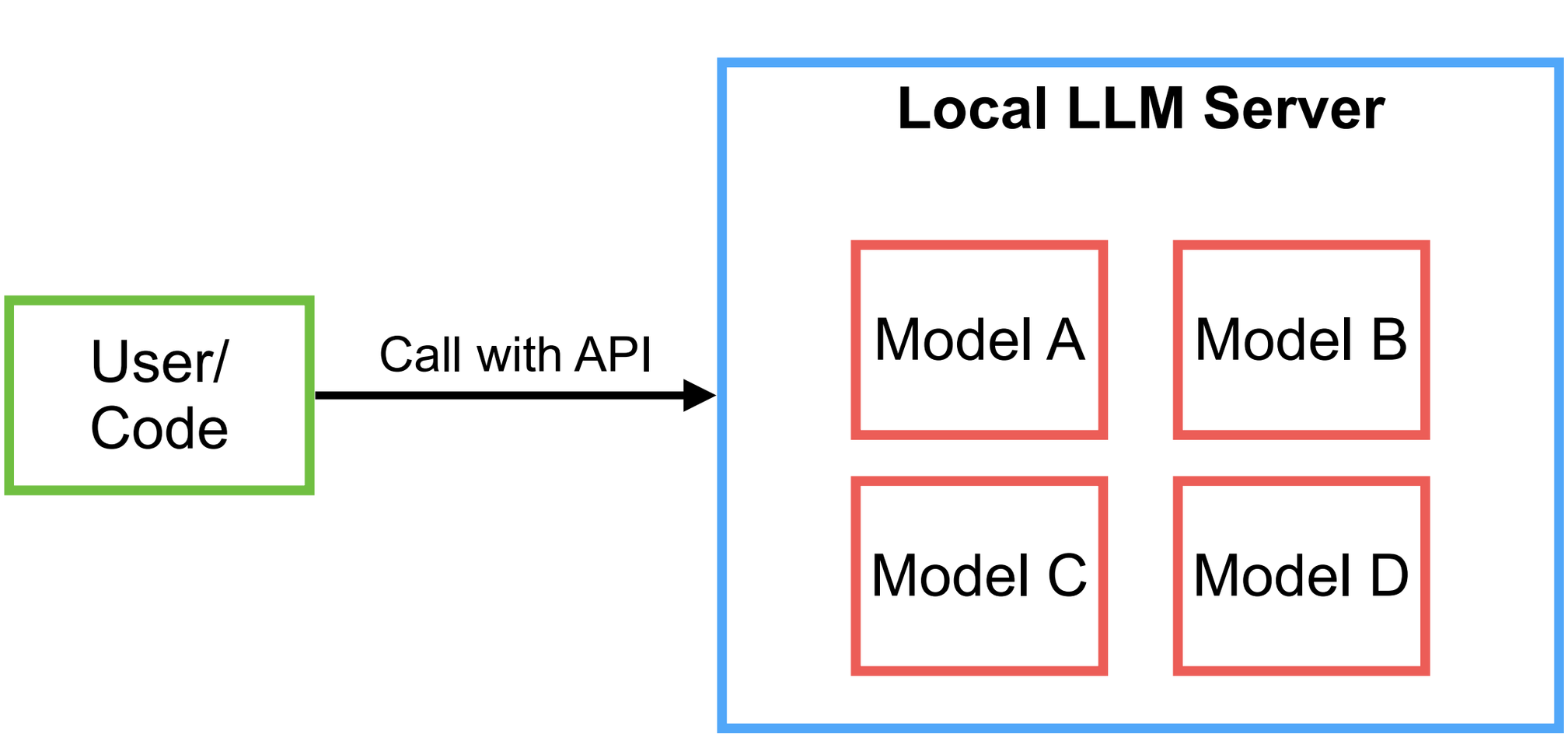
Local LLM

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

- Reduce cost
- Data privacy
- Responsive
- Offline mode

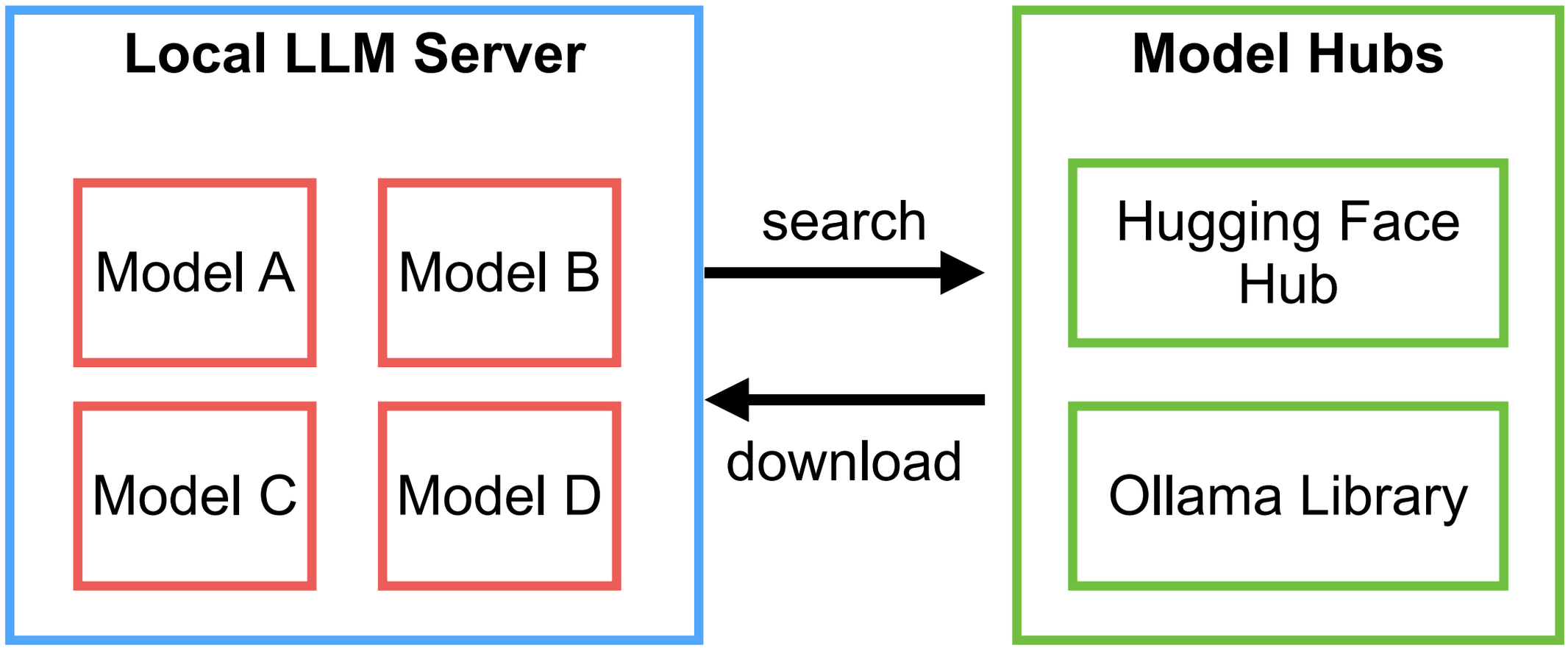
Local LLM

Improve your LLM models, more accurate answer



Models ?

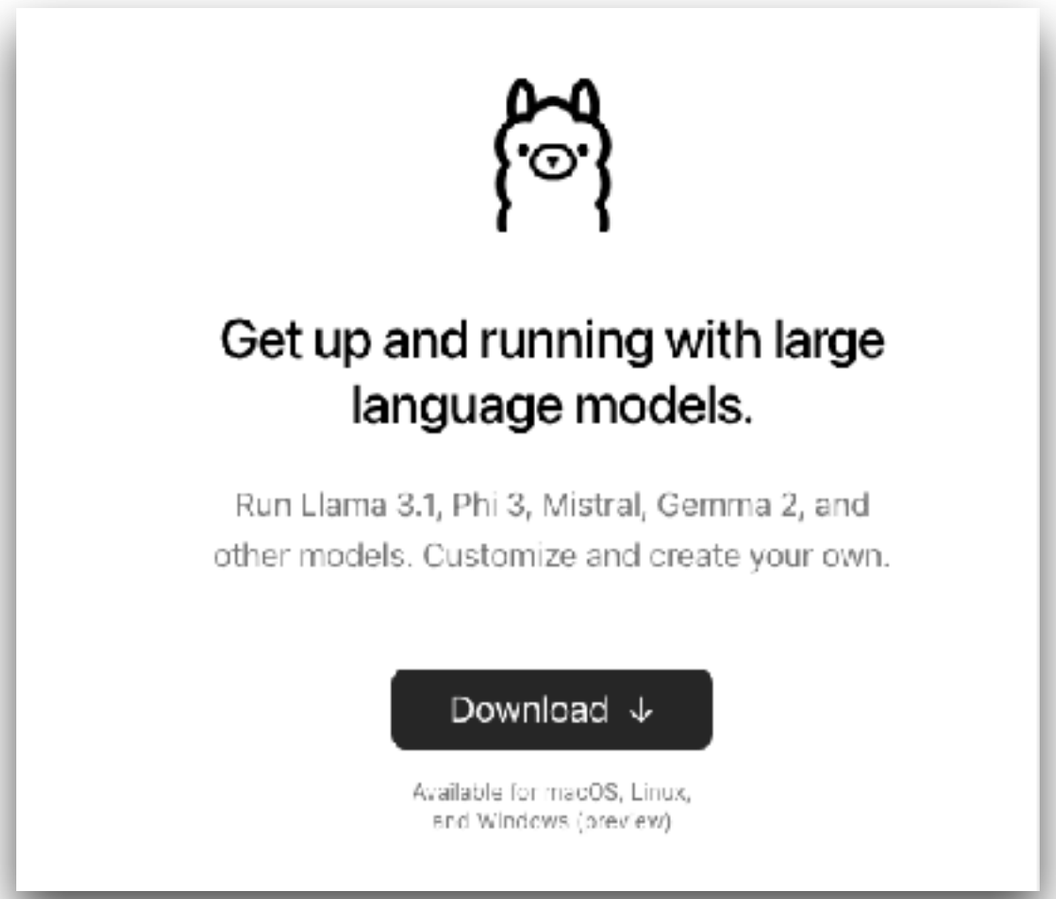
How to download models ?



Local LLM with Ollama

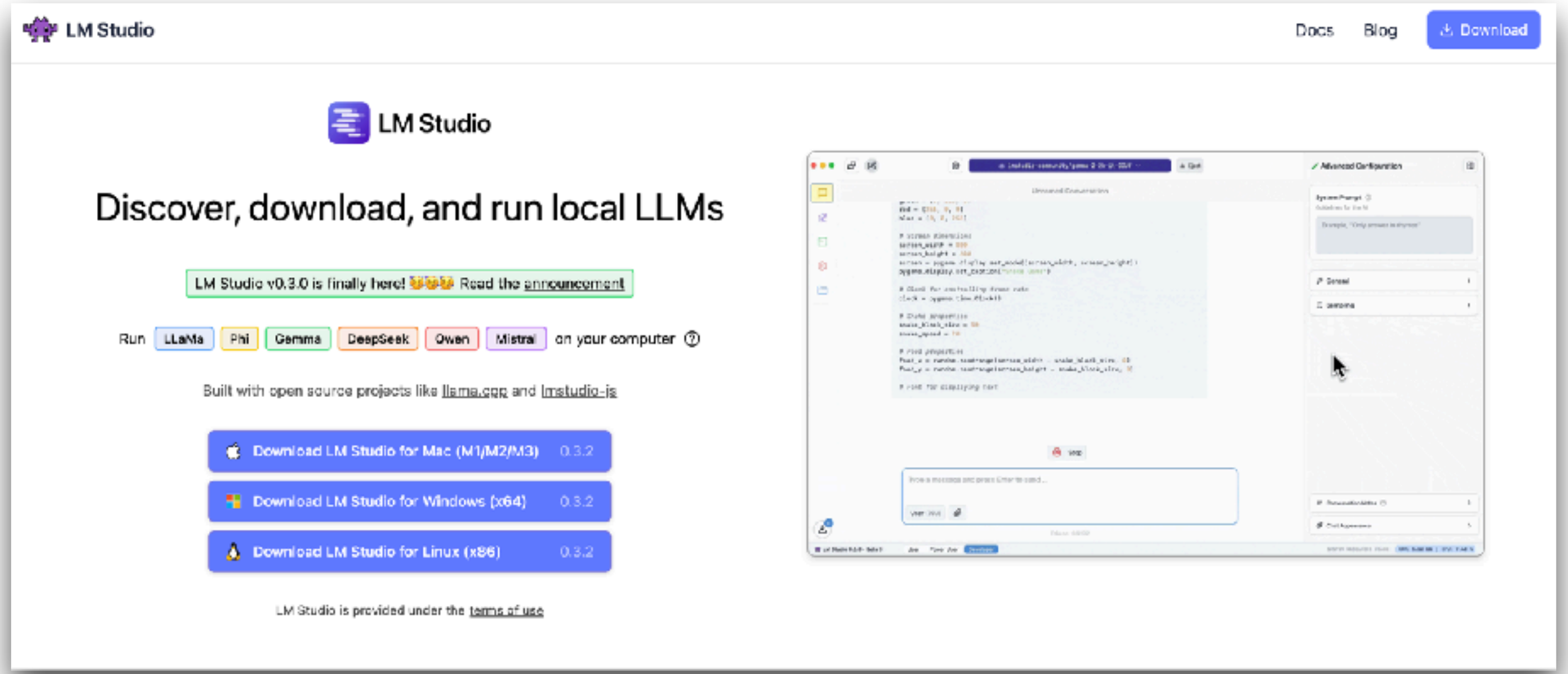
$ollama run **llama3.2**

LLM model



https://ollama.com/

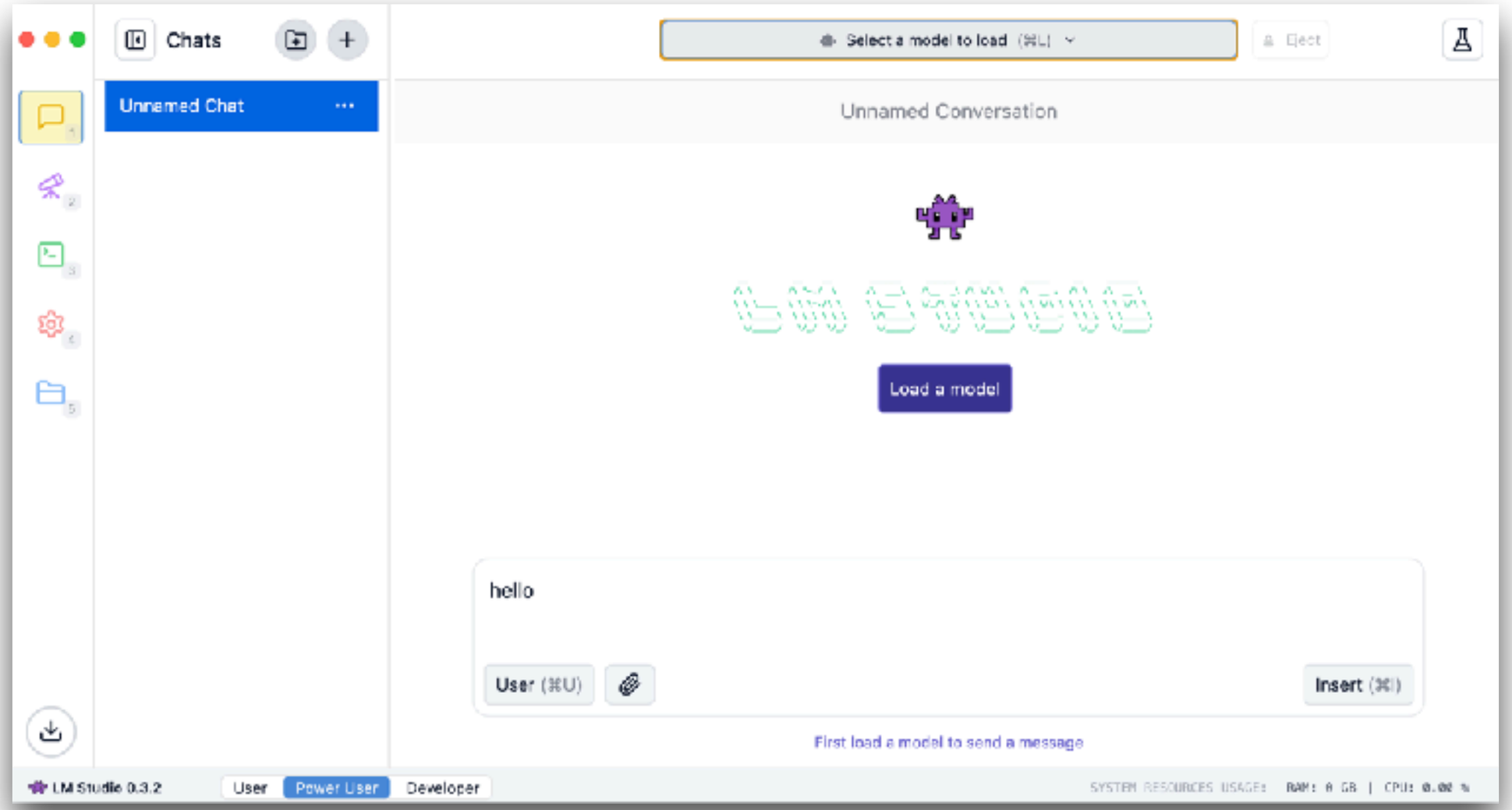
Local LLM with LM Studio



https://lmstudio.ai/

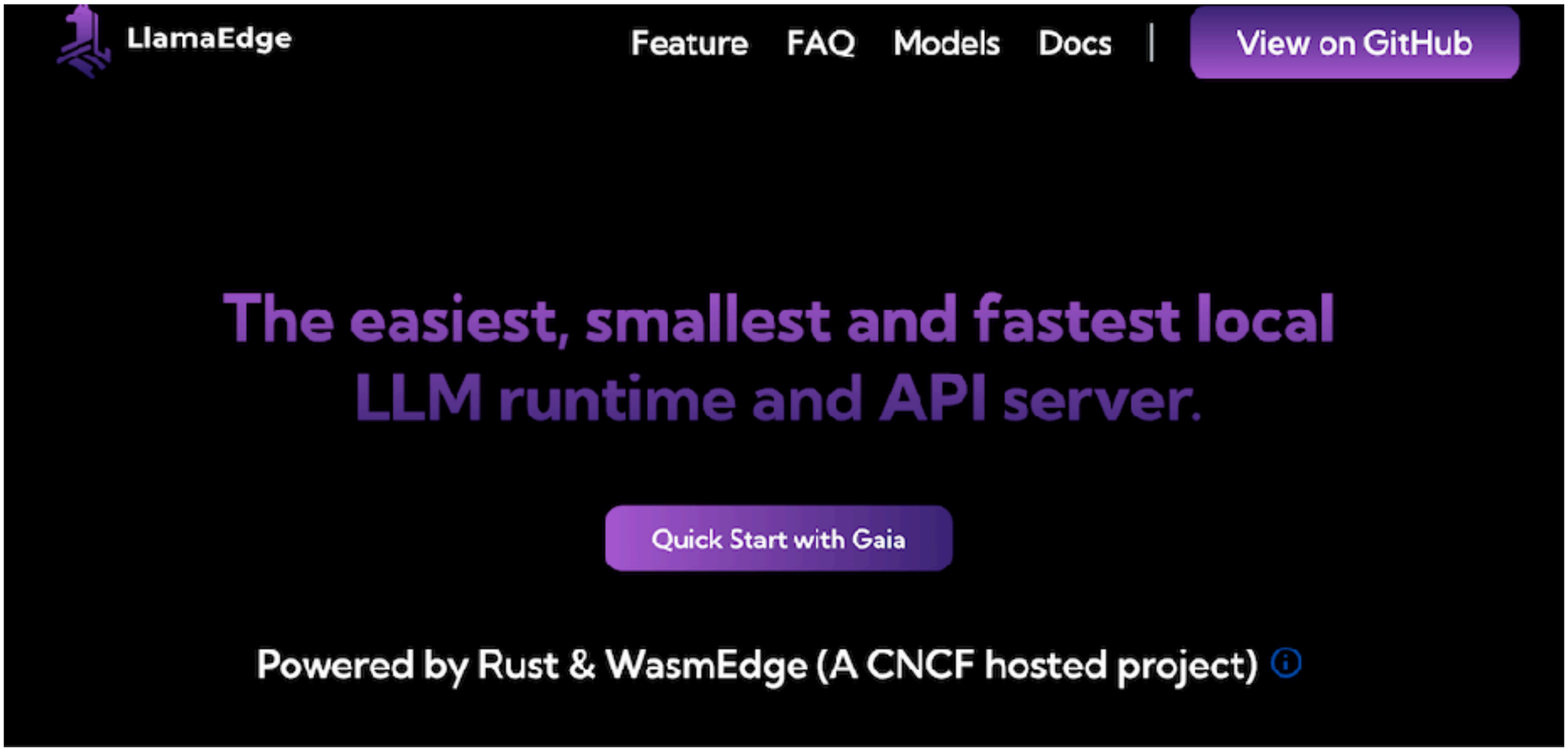
Local LLM with LM Studio

Load model from Hugging Face



https://lmstudio.ai/

Local LLM with LlamaEdge



https://llamaedge.com/

Local LLM with LocalAI

https://localai.io/

More

GPT4All

LlamaFile

Jan.ai

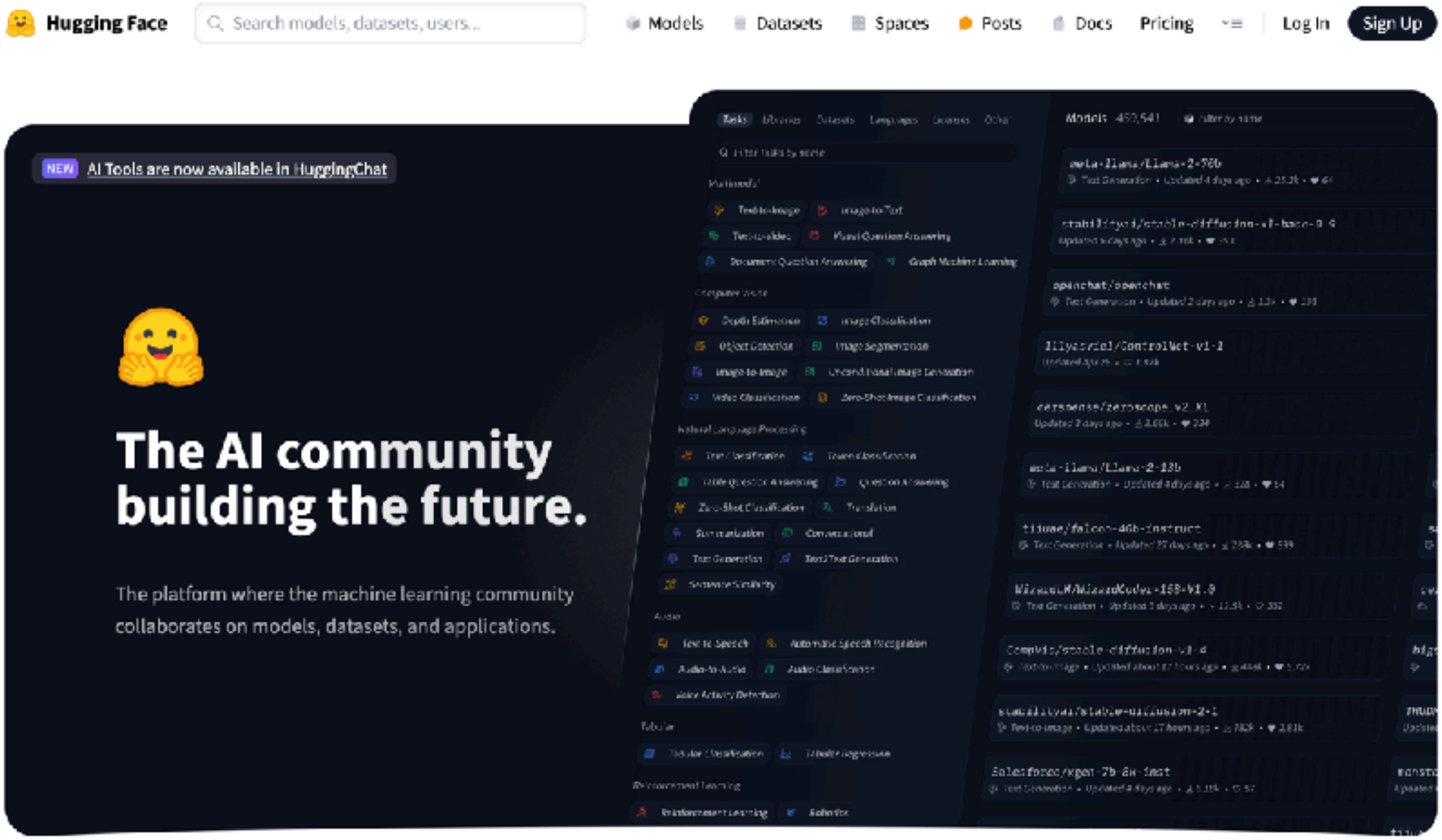
NextChat

Anything LLM

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

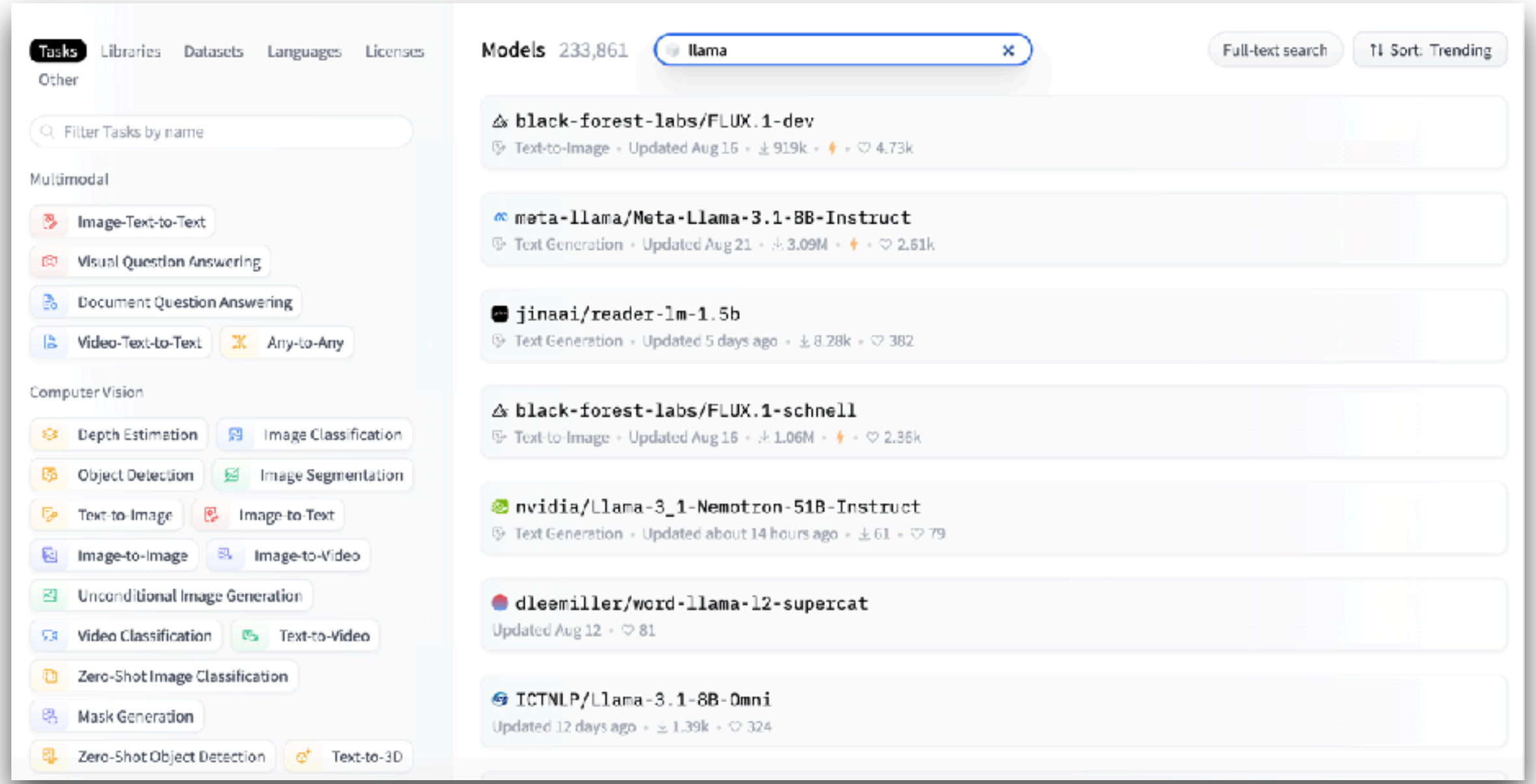
LLM Models

Hugging Face Model Hub



https://huggingface.co/

Hugging Face :: Model



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...

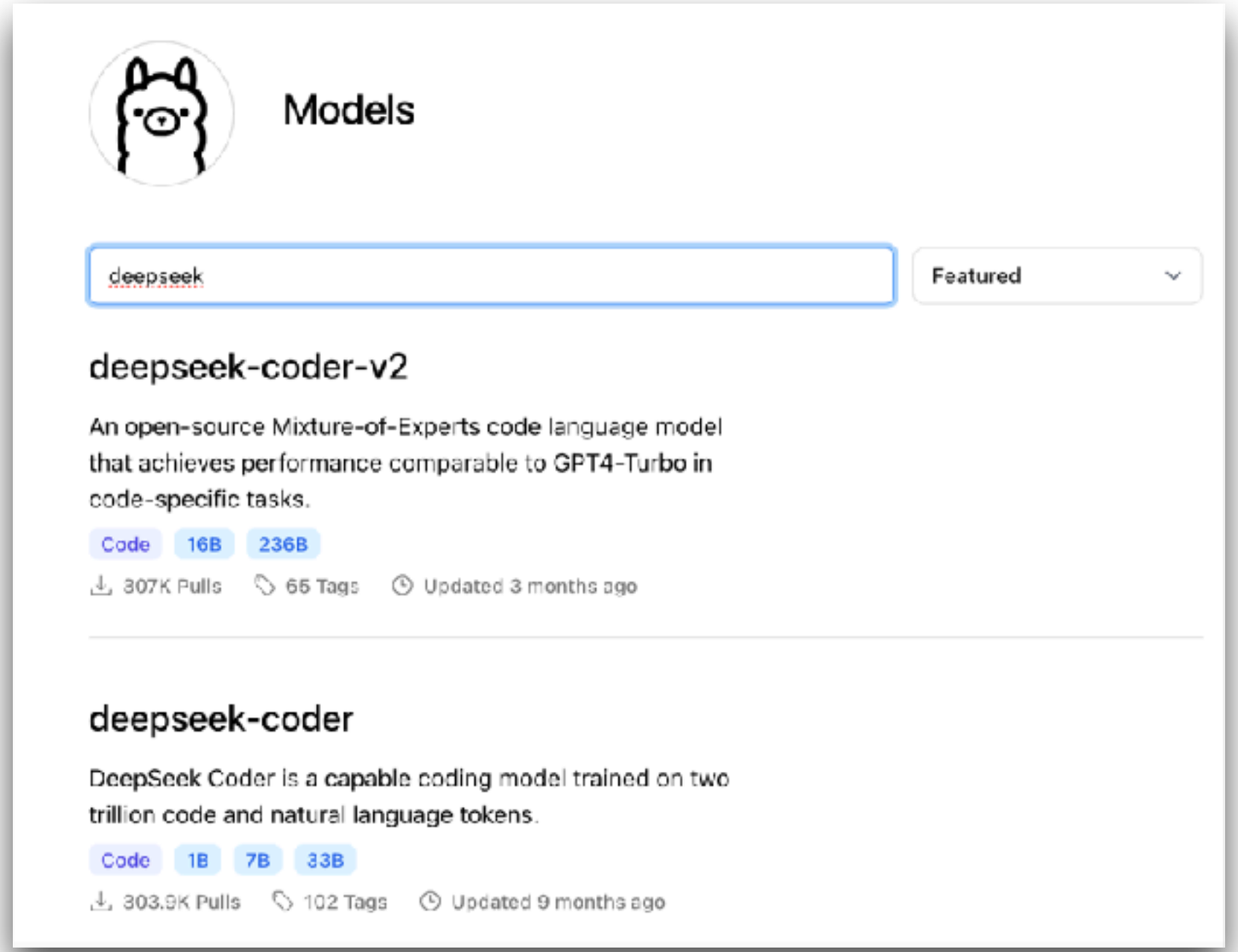
Filter model types

all | 🟢 base | 🔶 instruction-tuned | EXT external-evaluation

T	Model	Win Rate	humaneval-python	java	javascript	cpp
🔶 EXT	OpenCodeInterpreter-DS-33B	55.83	75.23	54.8	69.06	64.47
🔶 EXT	Nxcode-CQ-7B-orpo	55.42	87.23	60.91	71.69	68.84
🔶	CodeQwen1.5-7B-Chat	55.08	87.2	61.04	70.31	67.85
🔶 EXT	CodeFuse-DeepSeek-33b	54.33	76.83	60.76	66.46	65.22
🔶 EXT	DeepSeek-Coder-33b-instruct	52	80.02	52.03	65.13	62.36
🔶 EXT	Artigenz-Coder-DS-6.7B	51.5	70.89	56.84	66.16	59.75
🔶 EXT	DeepSeek-Coder-7b-instruct	50.33	80.22	53.34	66.8	59.66
🔶 EXT	OpenCodeInterpreter-DS-6.7B	49.67	73.2	51.41	63.85	60.81

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

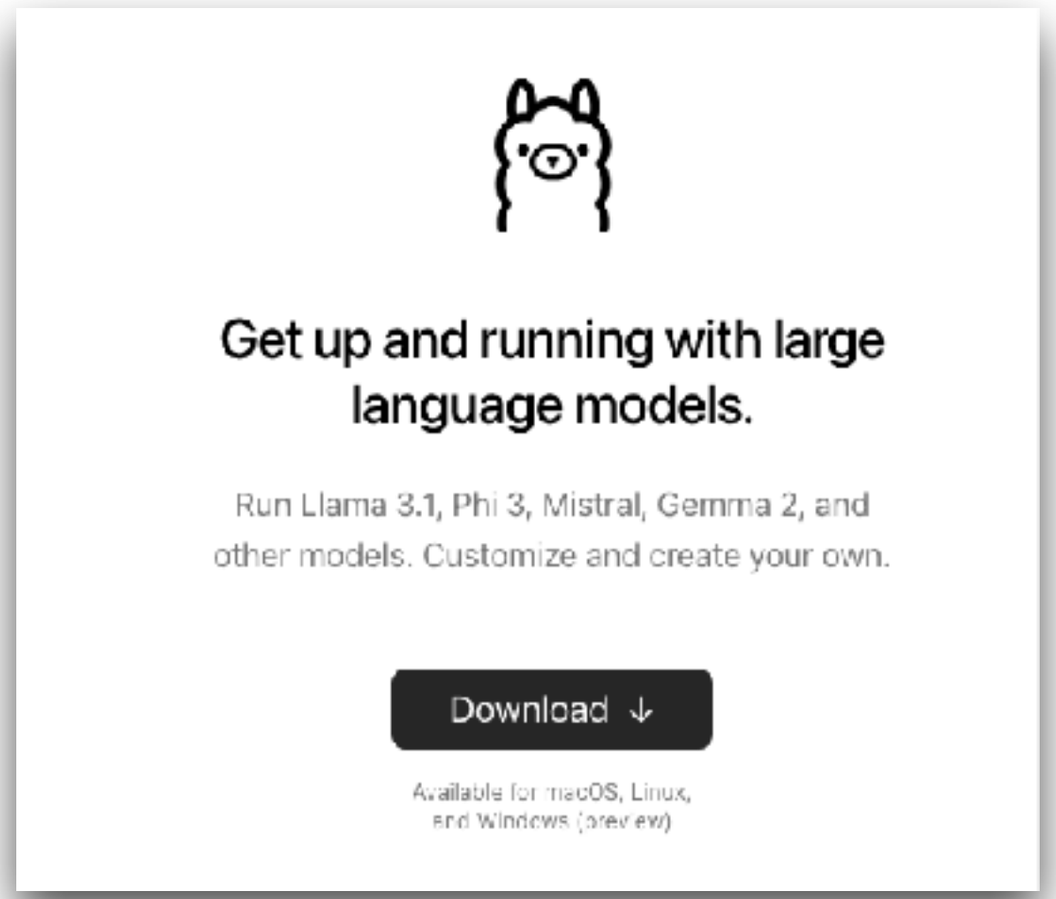
Model in Ollama



https://ollama.com/library

Workshop with Ollama

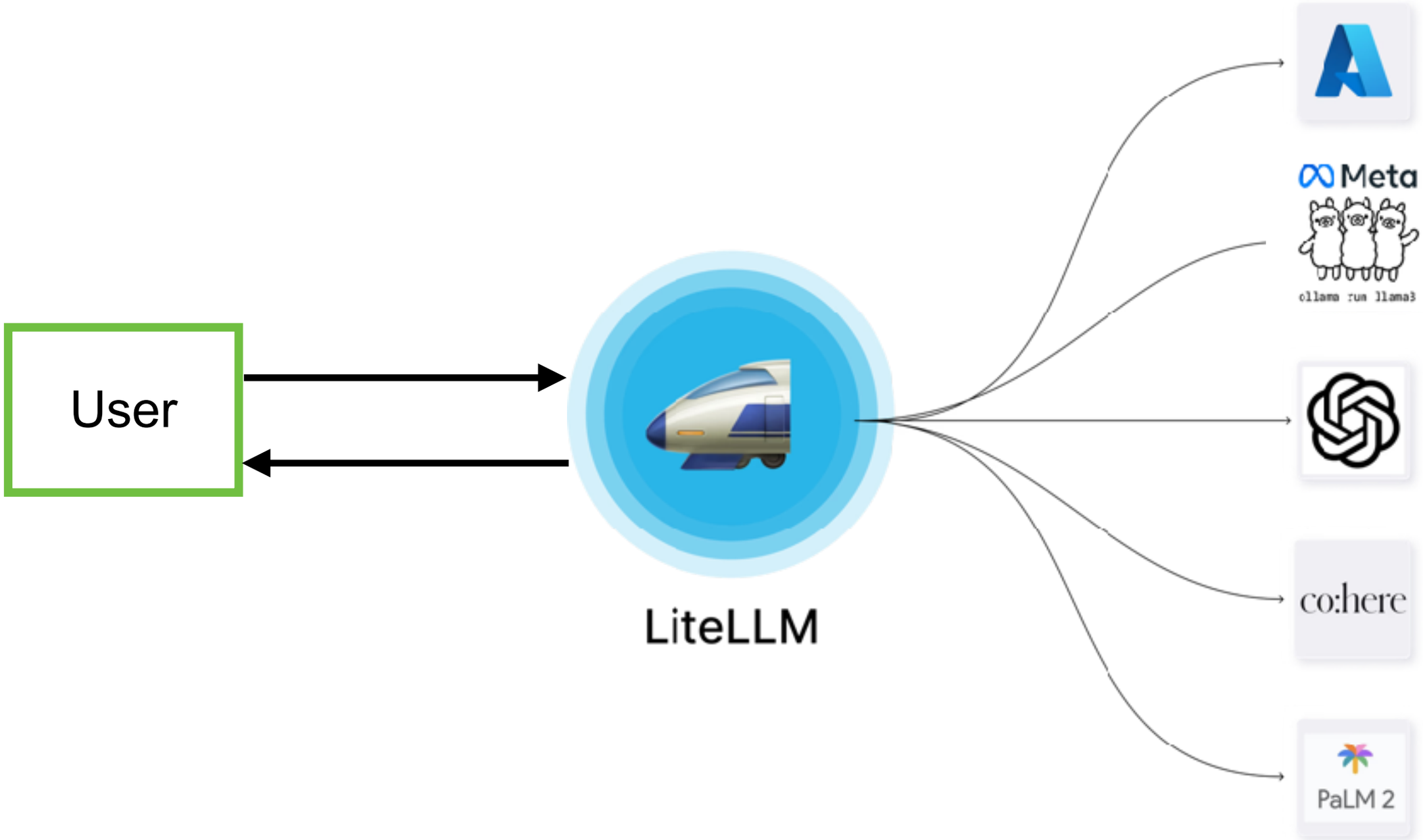
$ollama run **llama3.1**



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

LiteLLM as a Proxy

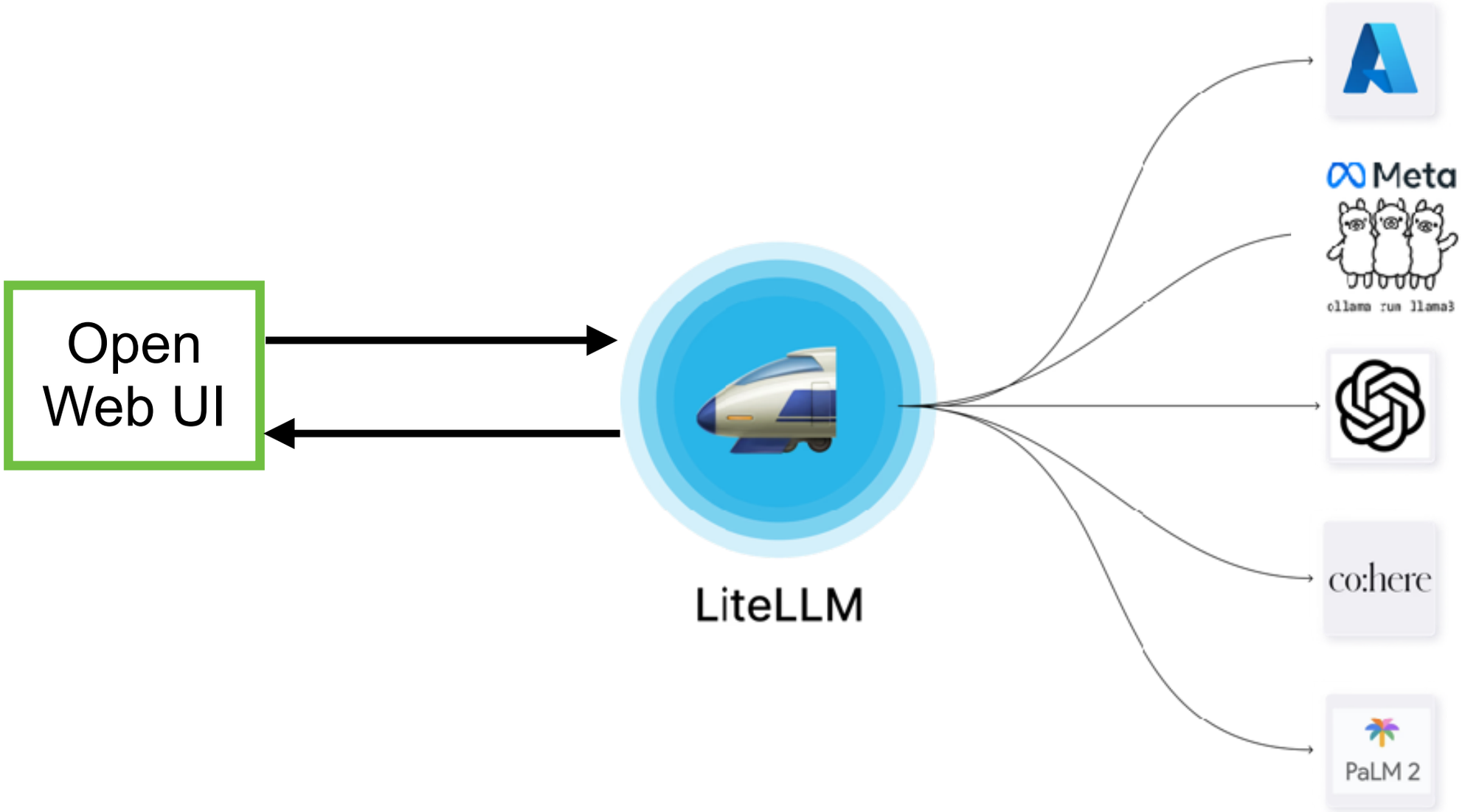
LiteLLM as a Proxy



https://www.litellm.ai/

Workshop

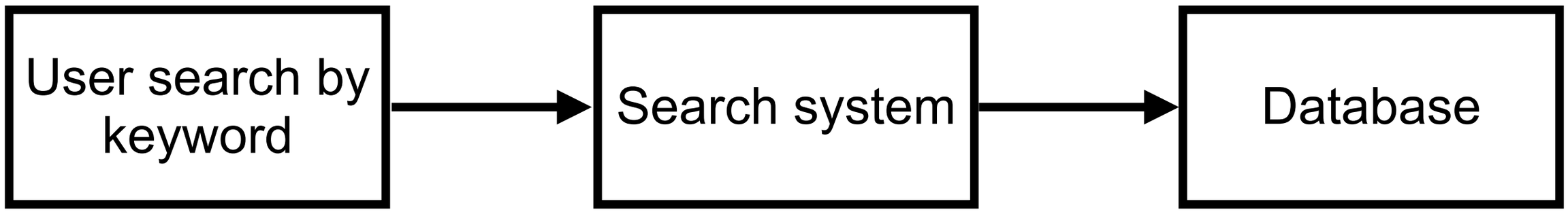
Use docker compose to build and run



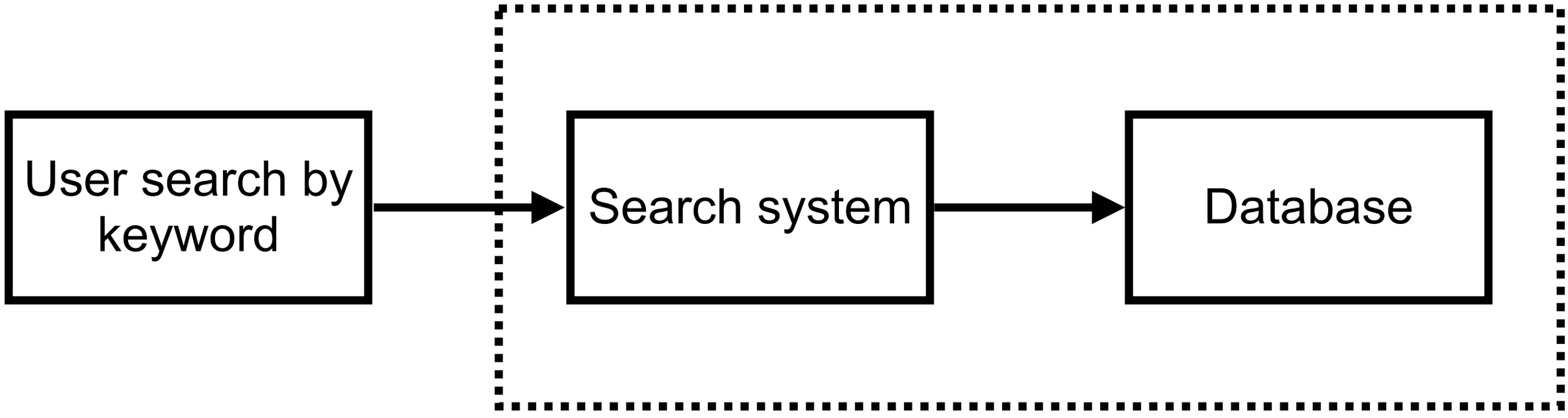
https://github.com/up1/workshop-ai-with-technical-team/wiki/LiteLLM-and-WebUI

Flow of search system

Search from user



Search from user



Improve search ?
Keyword, semantic, hybrid ?

Search Techniques ?

NAME LIKE %ข้อมูล%

Full-text search (FTS)

N-gram

Vectorization or semantic search

Graph search

etc…

Need more accuracy !!

Start your journey