



# LLM Application workshop

Prompt engineering  
RAG (Retrieval Augmented Generation)





Facebook somkiat.cc

Page Messages Notifications 3 Insights Publishing Tools Settings Help

somkiat.cc  
@somkiat.cc

Home Posts Videos Photos

Liked Following Share ...

+ Add a Button



# Topics

Basic LLM (Large Language Model)

Problems and solutions

Prompt engineering

RAG (Retrieval Augmented Generation)

RAG processes and techniques

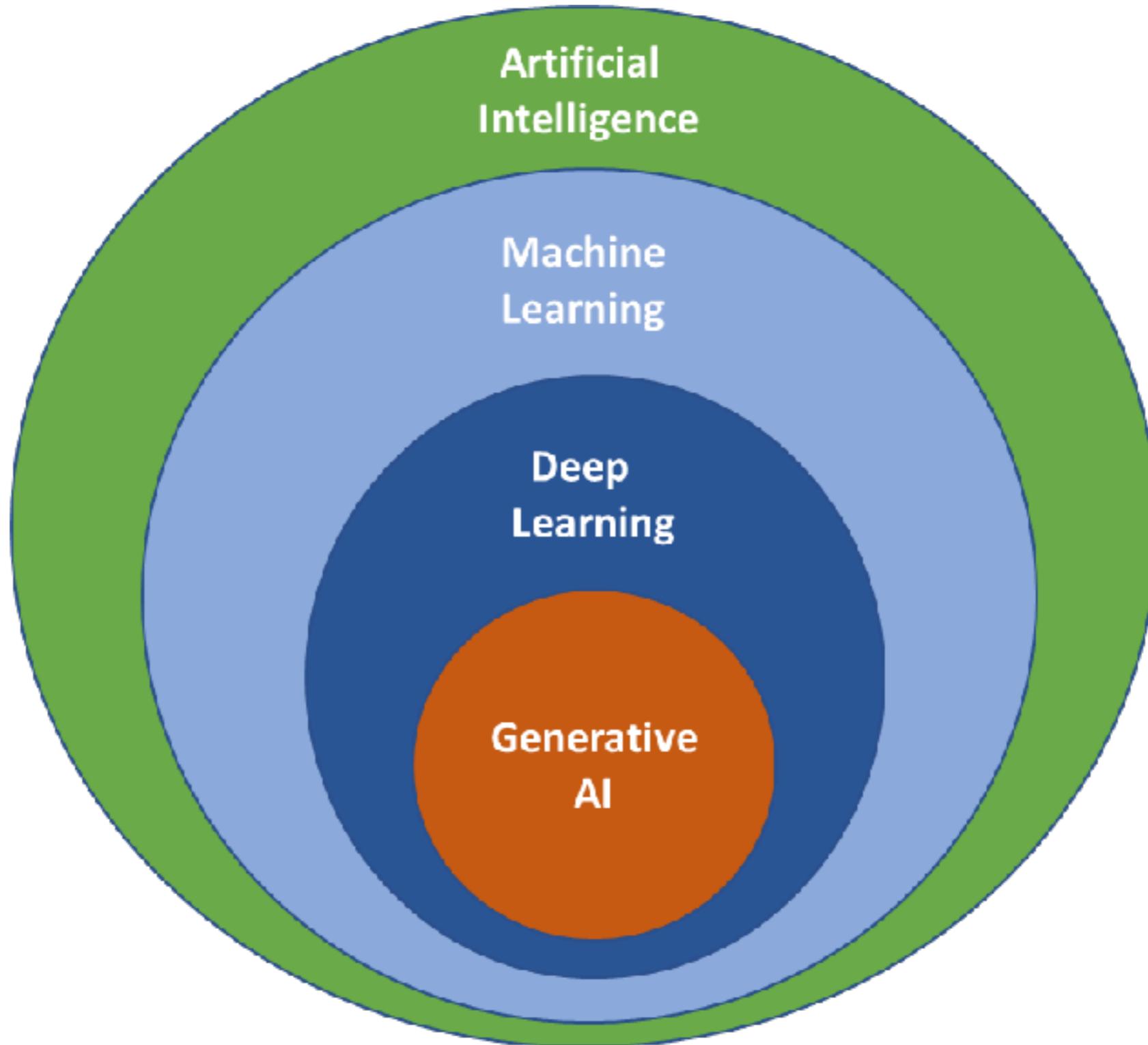
Workshop

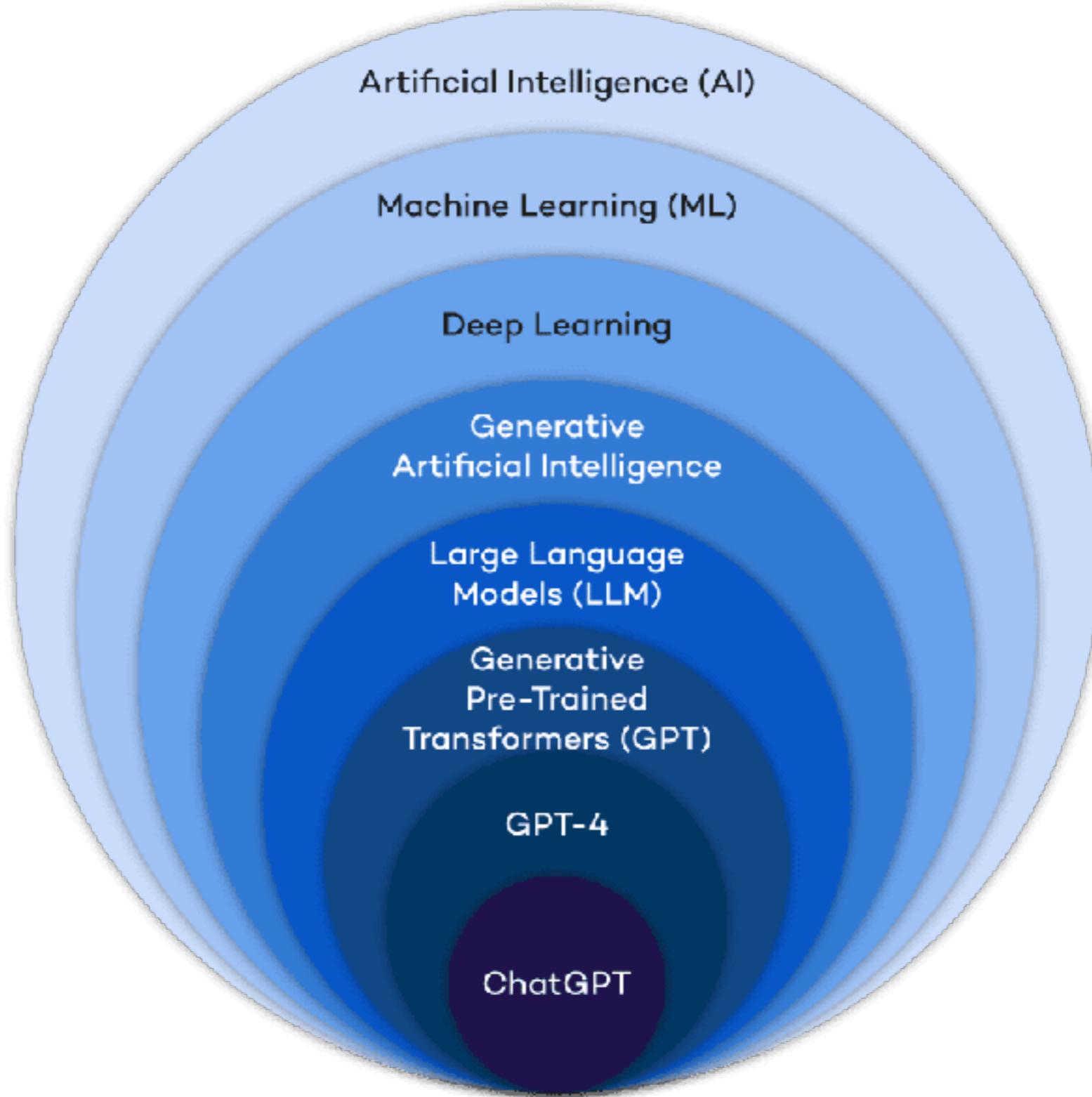


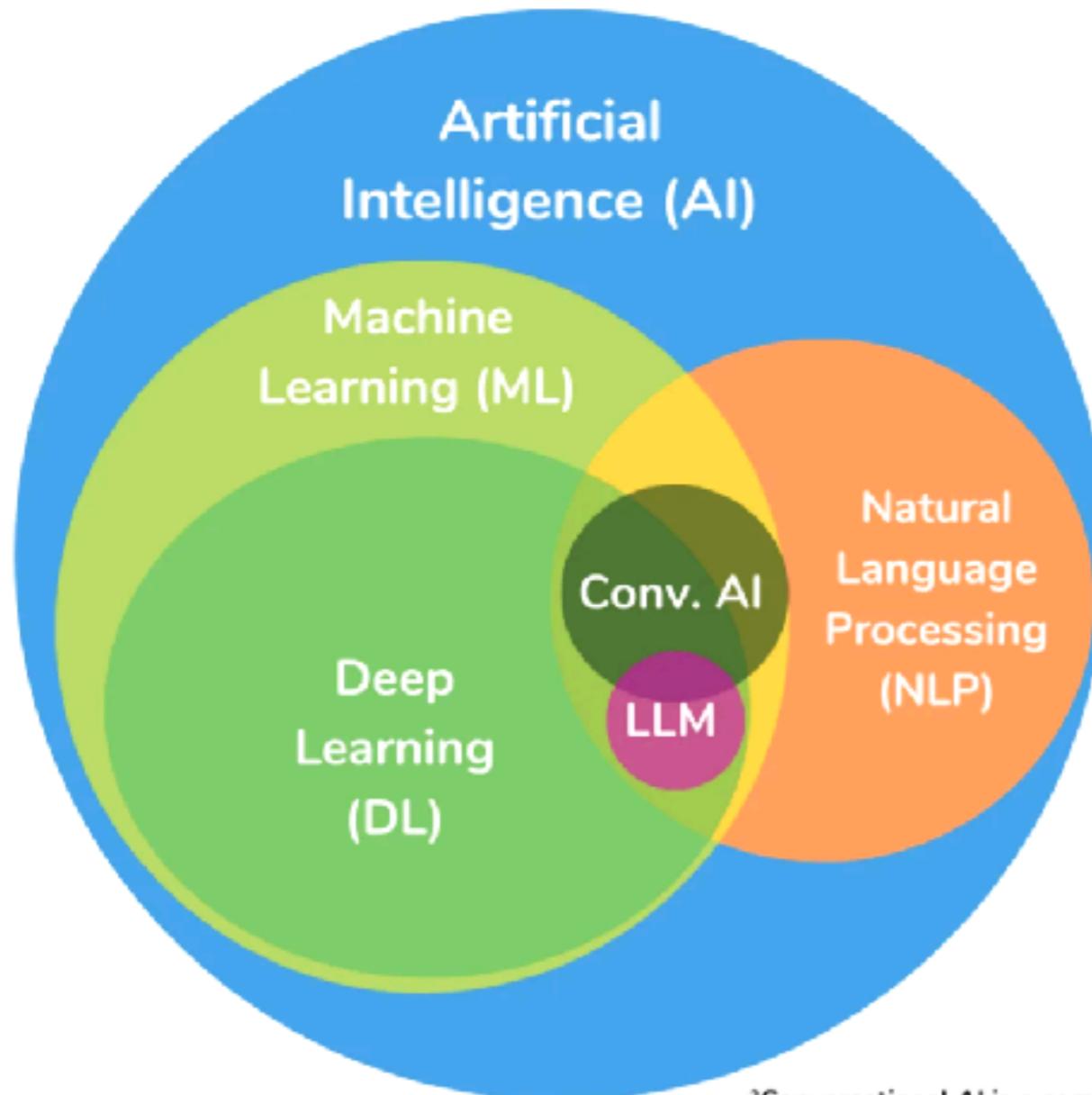
# AI

# (Artificial Intelligence)







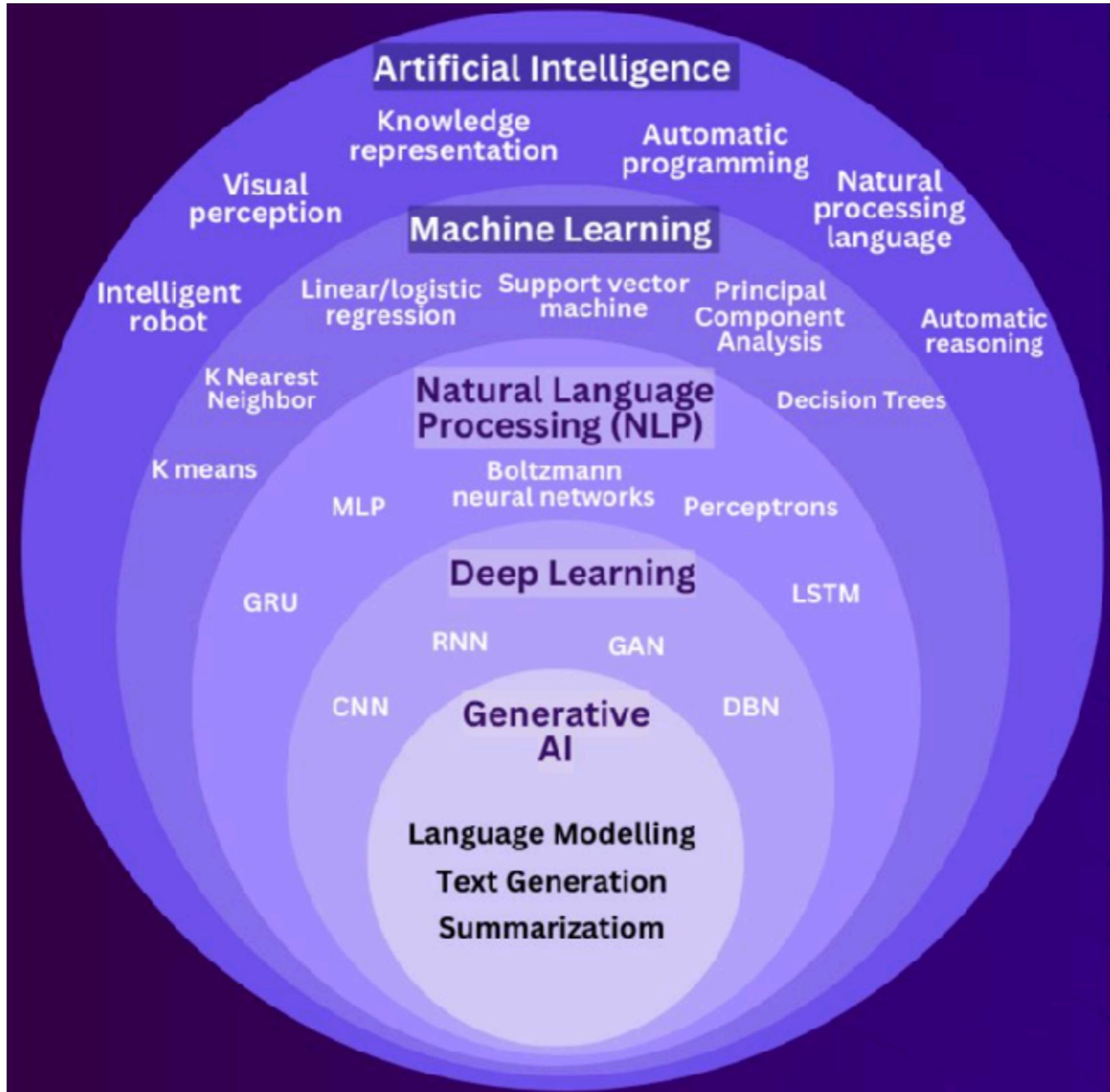


- Artificial Intelligence (AI)
- Machine Learning (ML)
- Deep Learning (DL)
- Natural Language Processing (NLP)
- Large Language Model (LLM)<sup>1</sup>
- Conversational AI (Conv. AI)<sup>2</sup>

<sup>1</sup>LLM is an intersection of DL and NLP

<sup>2</sup>Conversational AI is a combination of ML and NLP. It may include DL and LLM, but that isn't always the case.





# Deep Learning vs LLM

Deep Learning	LLM
A subset of ML using multi-layer neural networks	A specific type of DL model trained on massive text data
Broad (images, audio, time series, language, etc.)	Focused mainly on natural language understanding and generation
Can handle various types (image, video, audio, tabular)	Primarily trained on <b>textual data</b>
Size of data from small to large	Extremely large (billions to trillions of parameters)
Task-specific (e.g., classification, regression)	General-purpose language modeling (e.g., next-word prediction)



# LLM (Large Language Model)

Type of AI model designed to understand, generate and interact using human language

LLM model

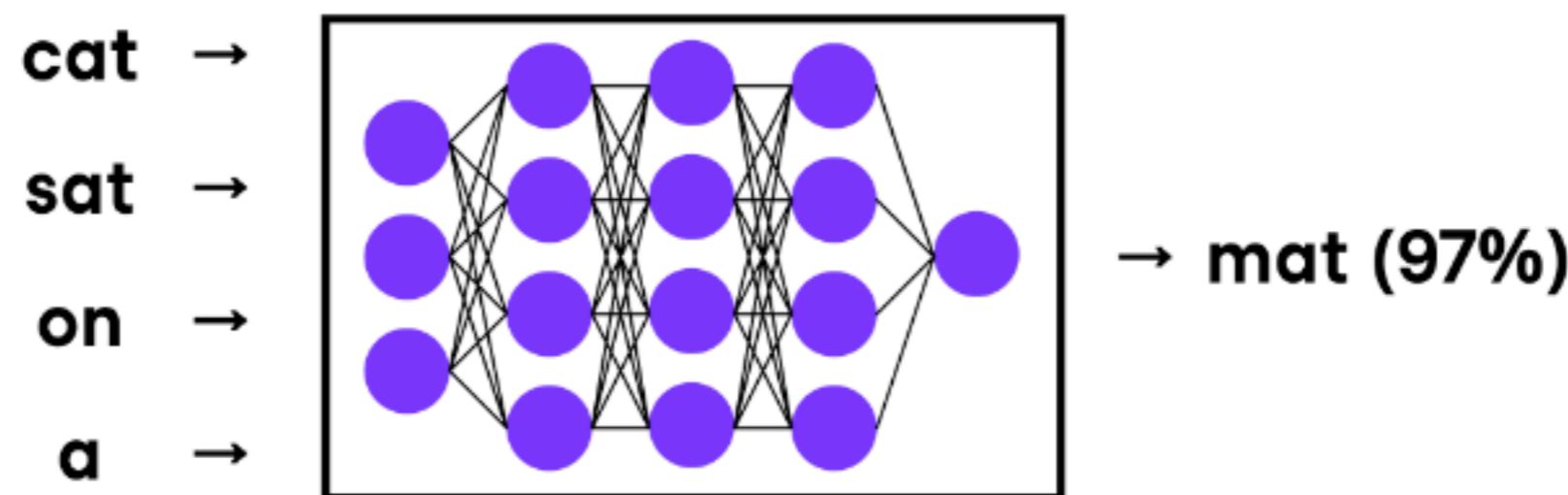
Training data !!



# LLM (Large Language Model)

Neural network

Predicts the next word in a sequence



# ສືເໜືອງ



# ตอนเที่ยงฉันจะกิน

— — — — —



# Next word ?

The boy go to the ...

Cafe

Hospital

Park

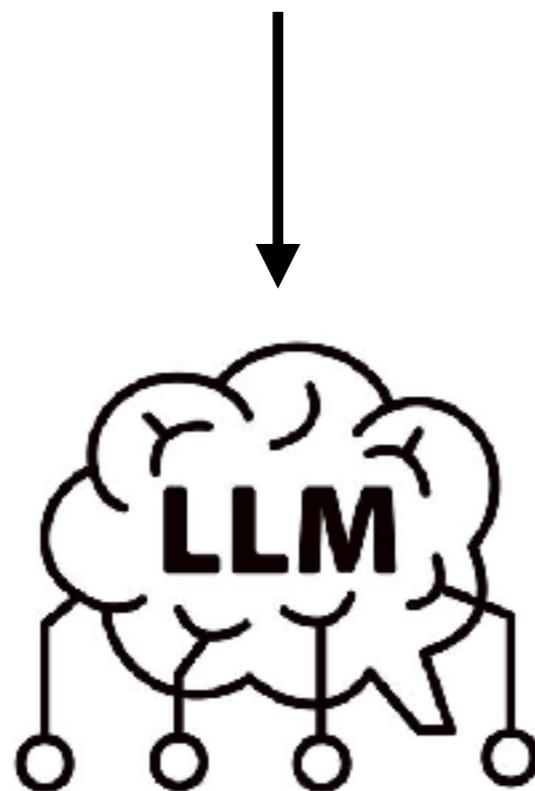
School



# Predict next word ?



The boy go to the ...



Cafe 0.1

Hospital 0.05

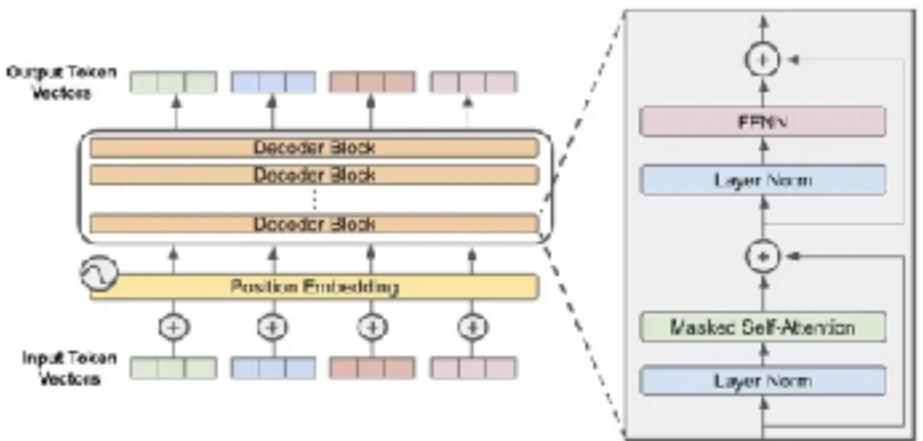
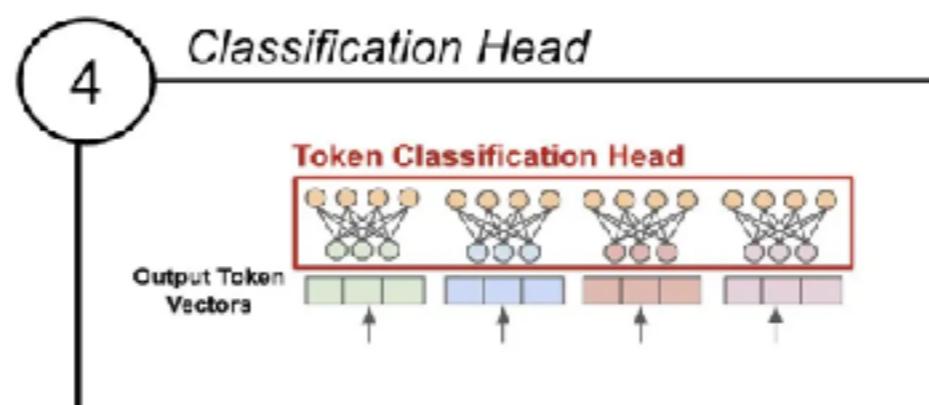
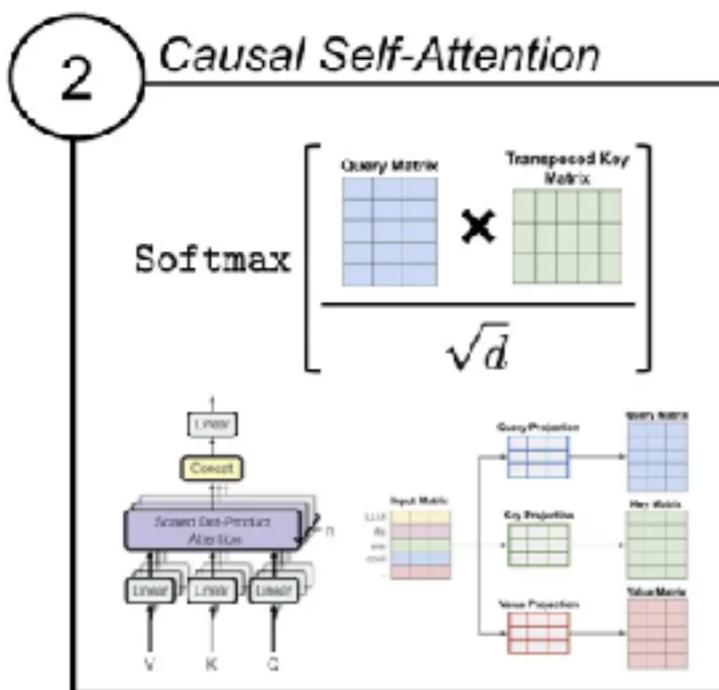
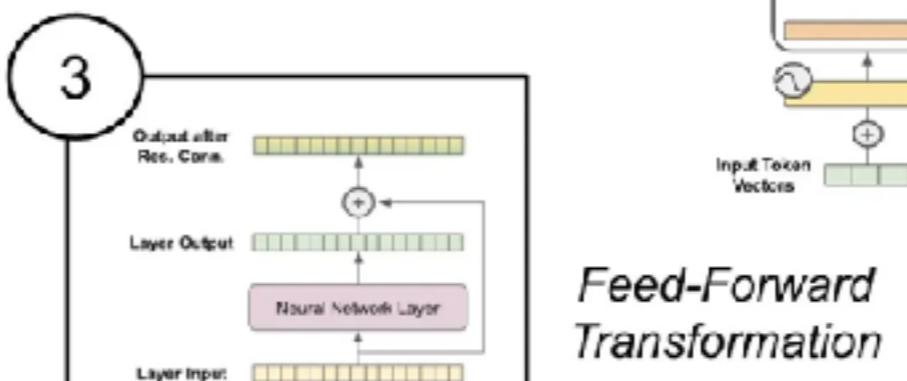
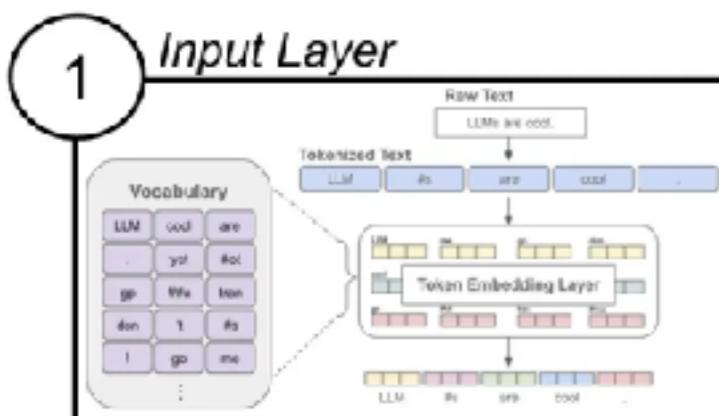
Playground 0.5

School 0.3



# LLM components !!

## Components of the Decoder-only Transformer



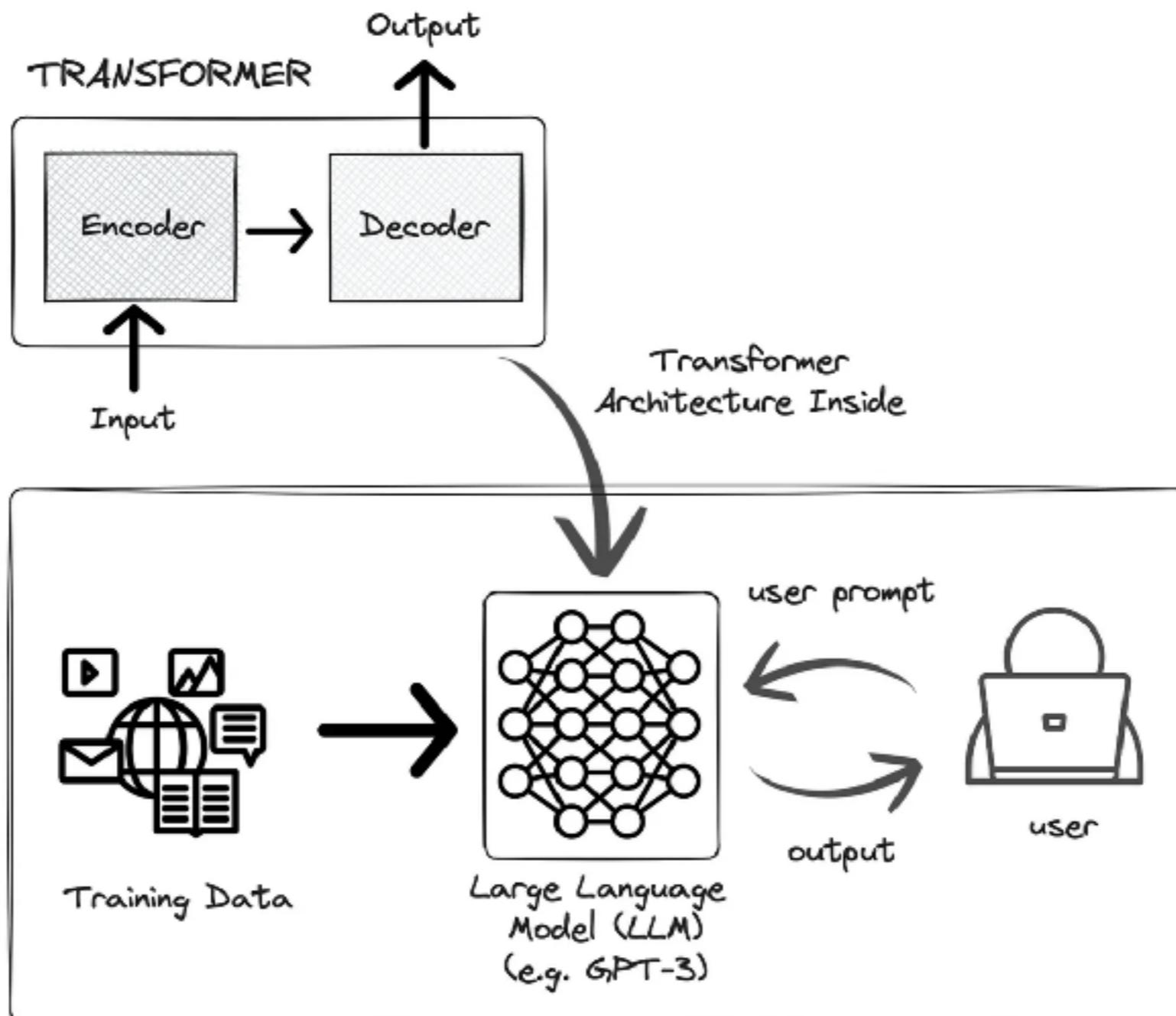
```
from torch import nn
class Block(nn.Module):
    def __init__(self, d, h, T, base, dropout=0.2, **kwargs):
        super().__init__()
        self.ln_1 = nn.LayerNorm(d)
        self.attn = CausalSelfAttention(d, h, T, base, dropout)
        self.ln_2 = nn.LayerNorm(d)
        self.ffn = FFN(d, base,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

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

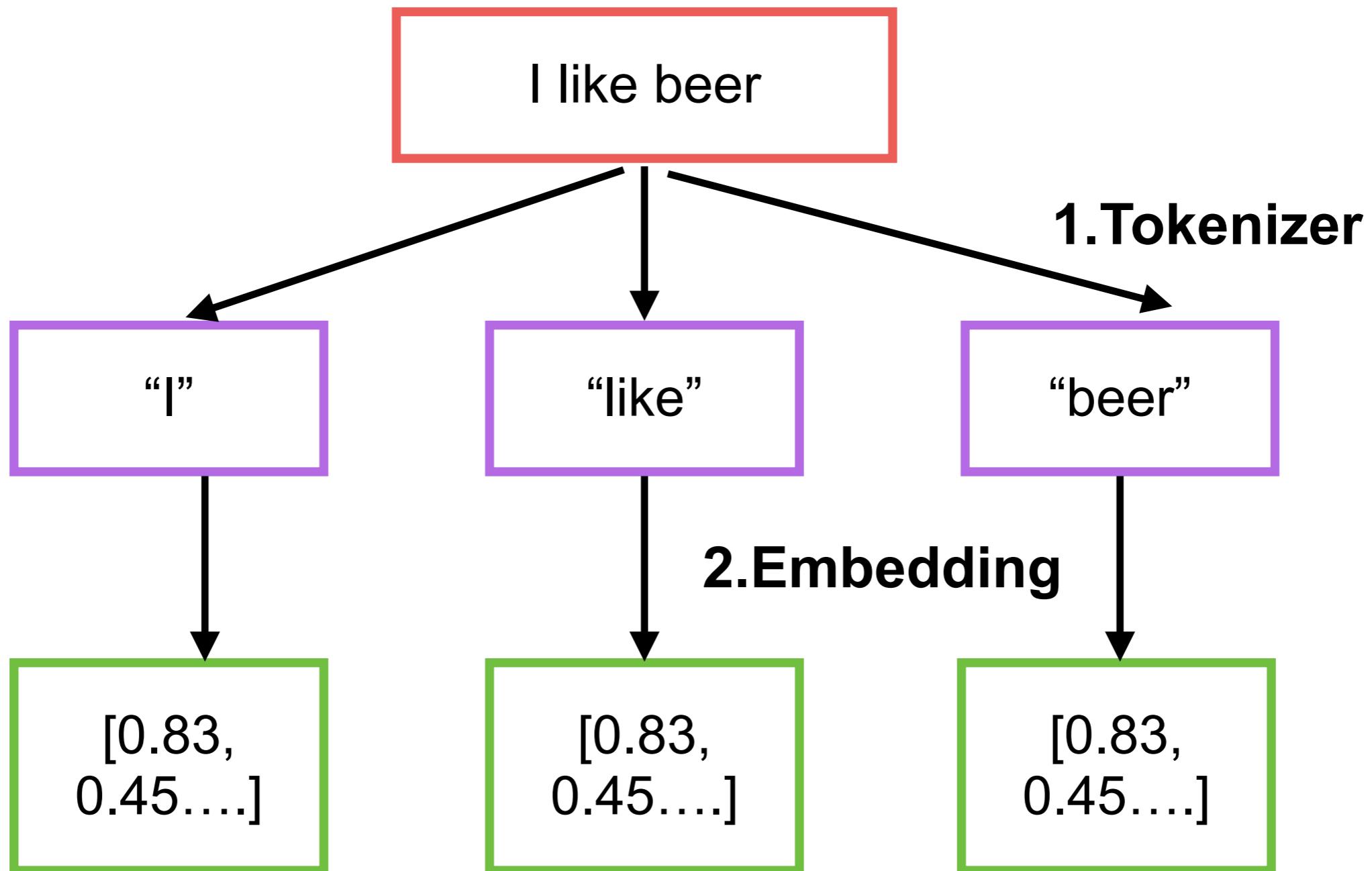
<https://stackoverflow.blog/2024/08/22/lms-evolve-quickly-their-underlying-architecture-not-so-much>



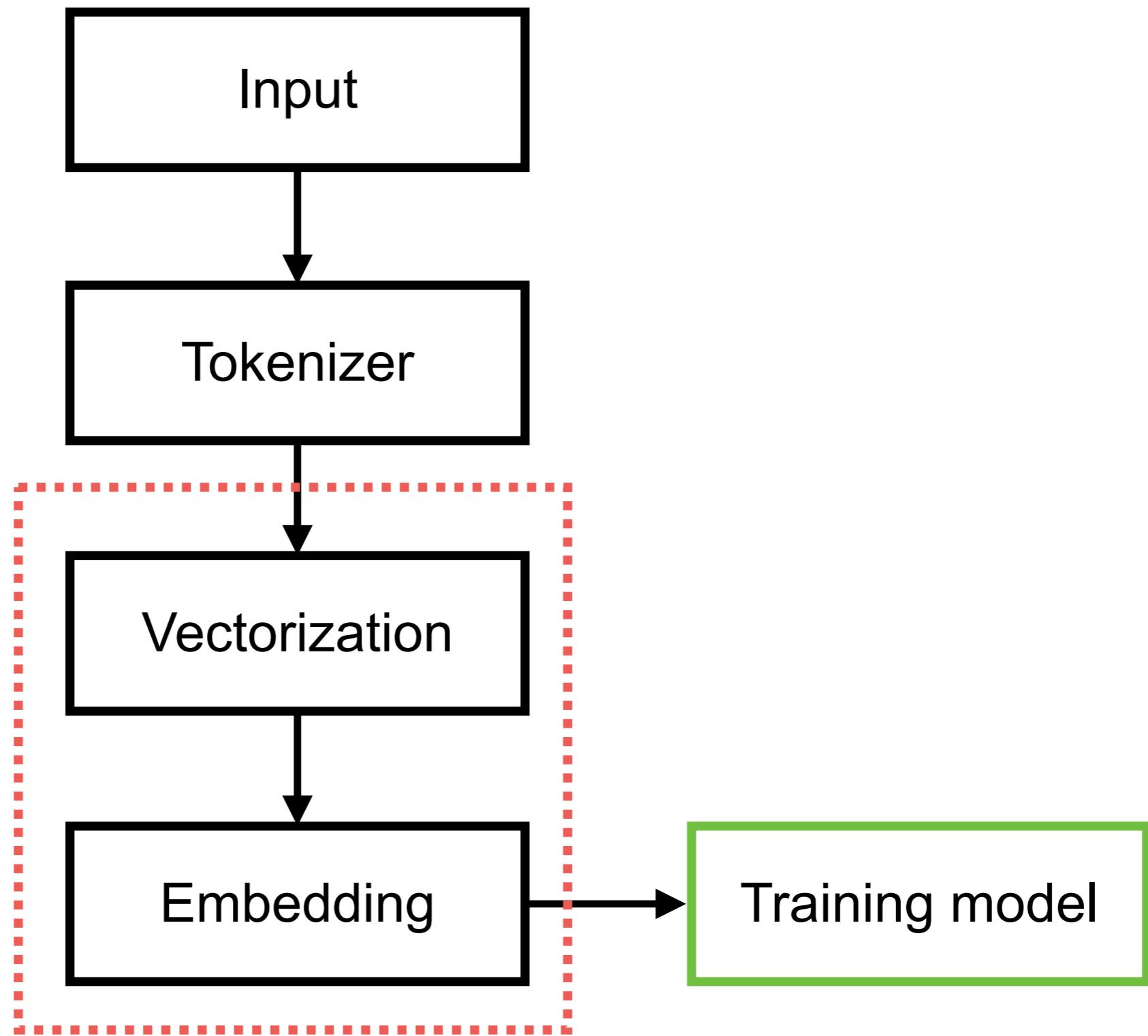
# Transformer inside



# Transformer Process !!



# Processes



# OpenAI Tokenization

The screenshot shows the OpenAI Tokenizer interface. At the top, there are three tabs: "GPT-4o & GPT-4o mini (coming soon)", "GPT-3.5 & GPT-4" (which is highlighted in green), and "GPT-3 (Legacy)". Below the tabs is a text input field containing the Thai text "ประเทศไทย". Underneath the input field are two buttons: "Clear" and "Show example". Further down, there are two sections labeled "Tokens" and "Characters", each showing the value "10" and "9" respectively. Below these sections is a preview area containing the input text "ประเทศไทย" with each character highlighted by a colored box (purple, red, blue, green, yellow). At the bottom of the interface are two buttons: "Text" and "Token IDs".

<https://platform.openai.com/tokenizer>

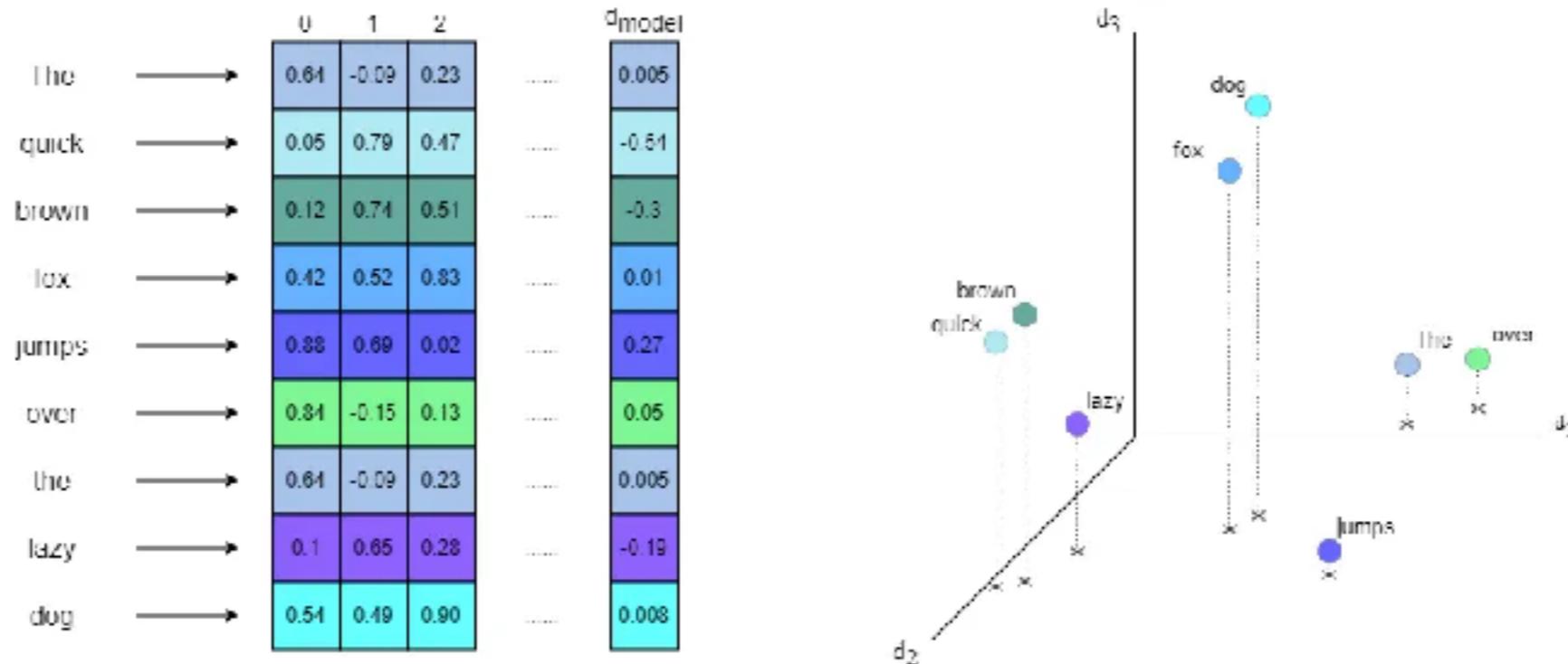


# Embedding

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

Semantic meaning

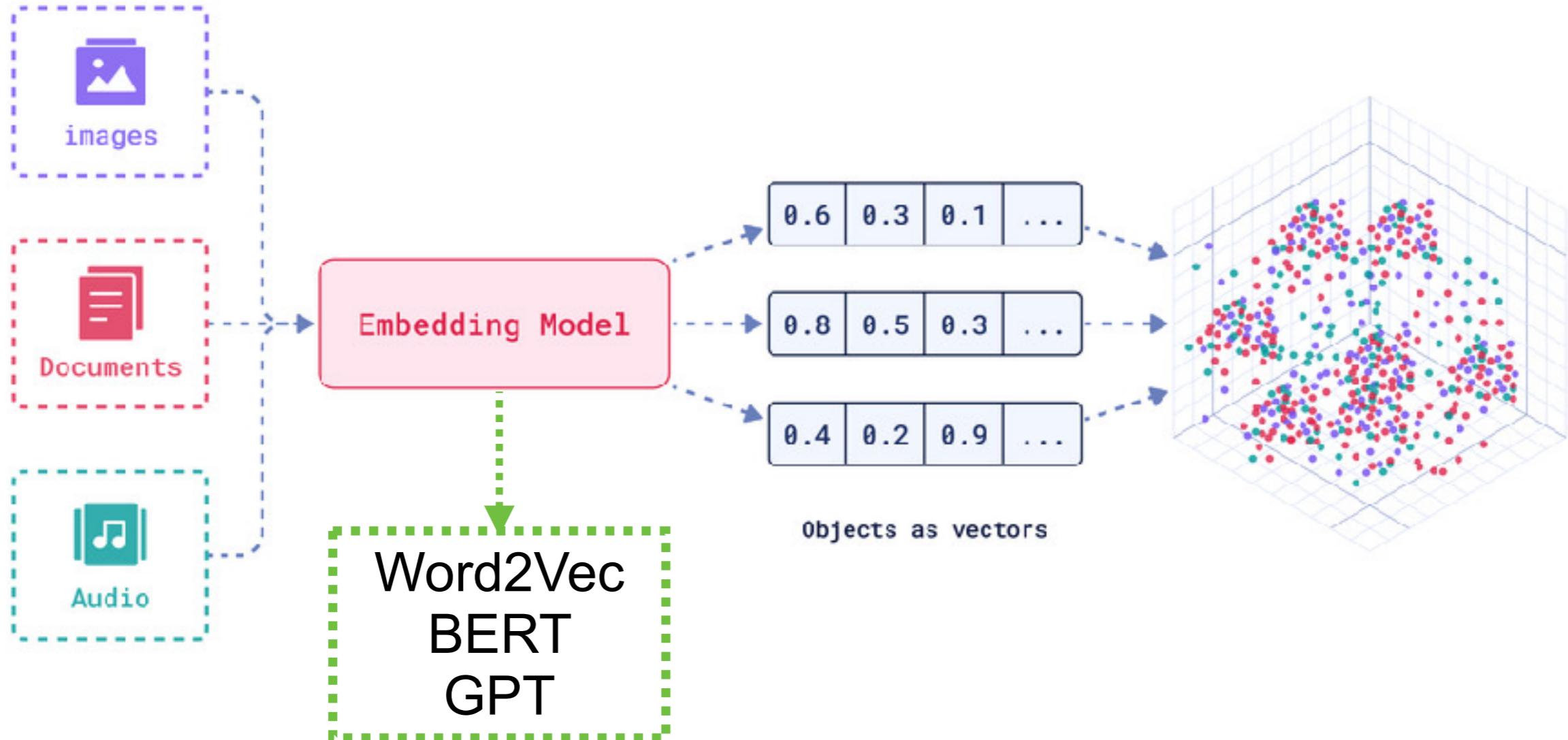
Improvement in ML tasks



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



# Embedding Models



<https://qdrant.tech/articles/what-are-embeddings/>



# eg. Embedding space

Synonym words in similar context !!  
Embeddings are about **semantics**

pretty attractive  
lovely nice cute  
beautiful elegant

dirty grisly  
awful ugly hideous  
grotesque messy  
disgusting

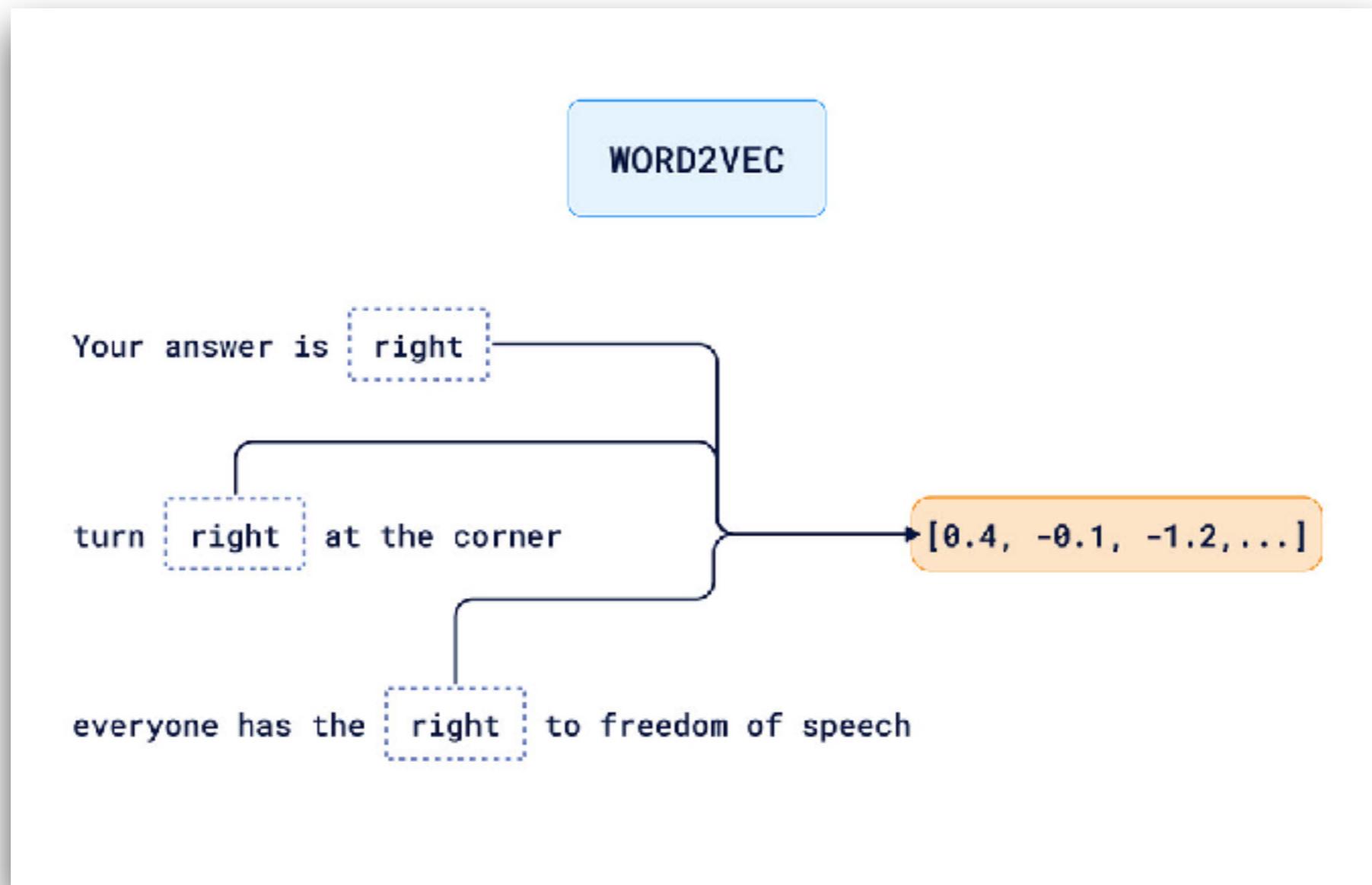
neutral impartial  
autonomous fair independent  
indifferent sovereign hands-off  
nonpartisan



<https://qdrant.tech/articles/what-are-embeddings/>

# Embedding model with Word2Vec

## Static embedding



# Problems ?

Limited context

Short context

Word  
disambiguation  
issues

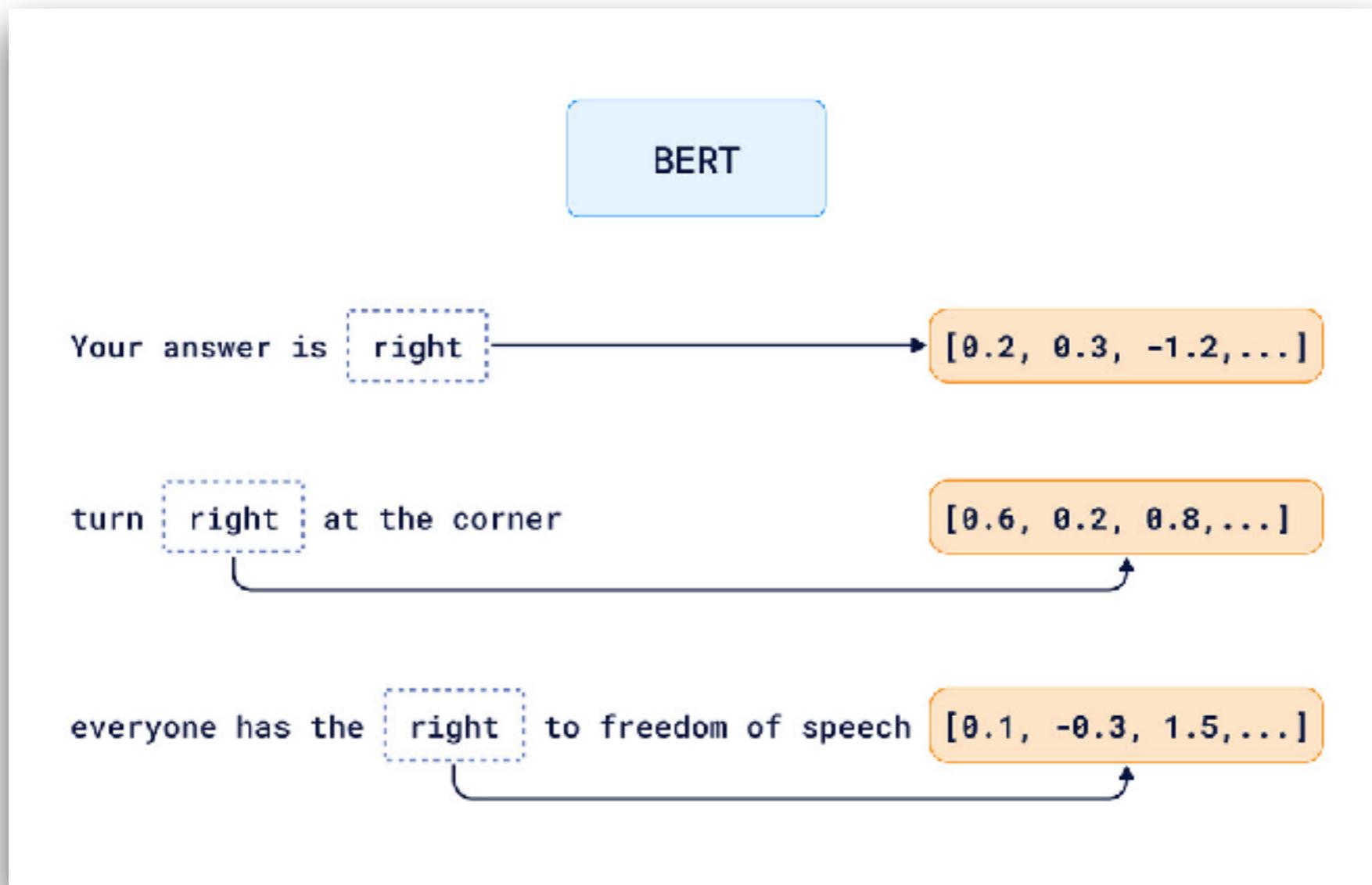
Generalization  
challenges  
(unseen situation)

<https://www.datacamp.com/blog/attention-mechanism-in-langs-intuition>



# Embedding model with BERT, GPT

Different context, different embedding



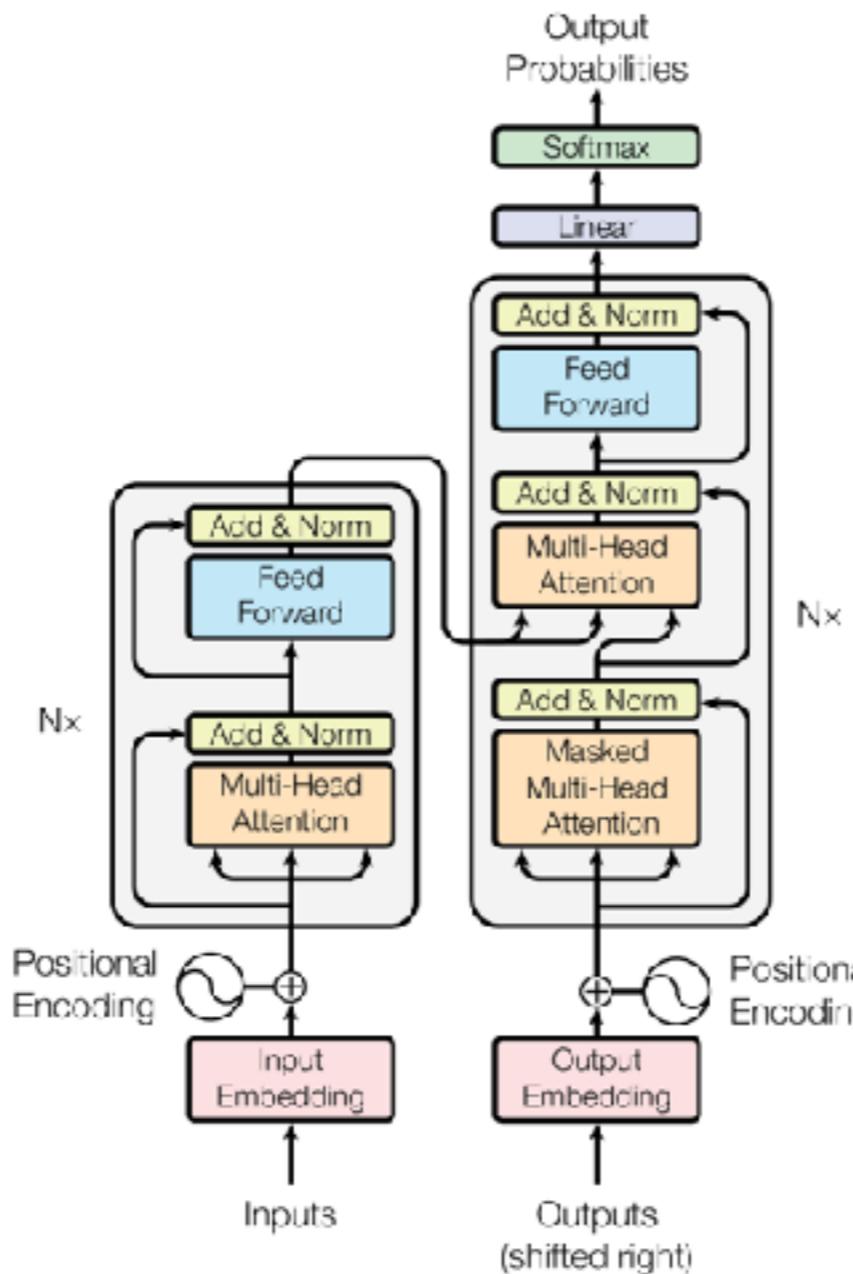
# Attention in LLM !!

<https://arxiv.org/abs/1706.03762>



# Transformer model architecture

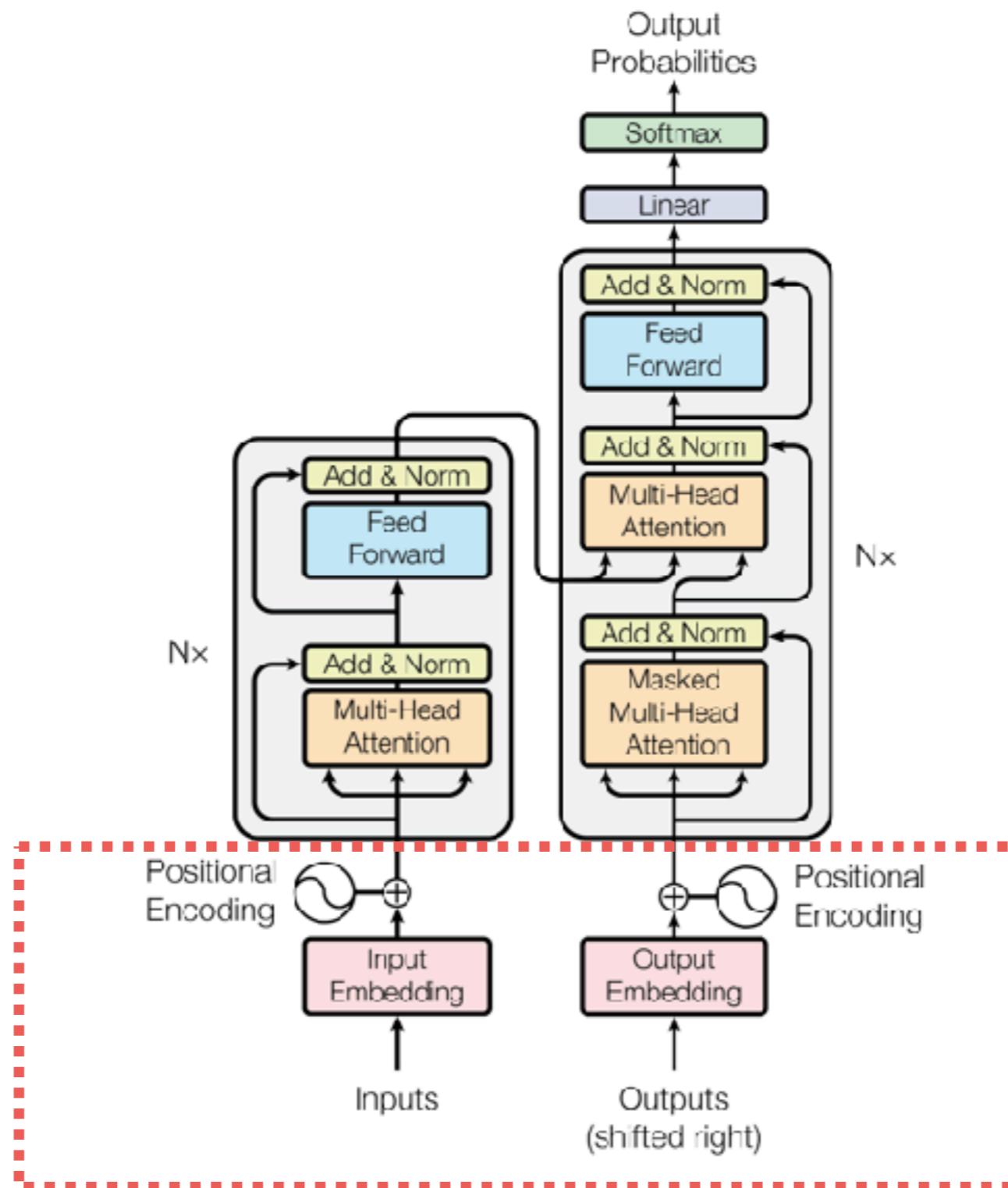
## Attention Is All You Need



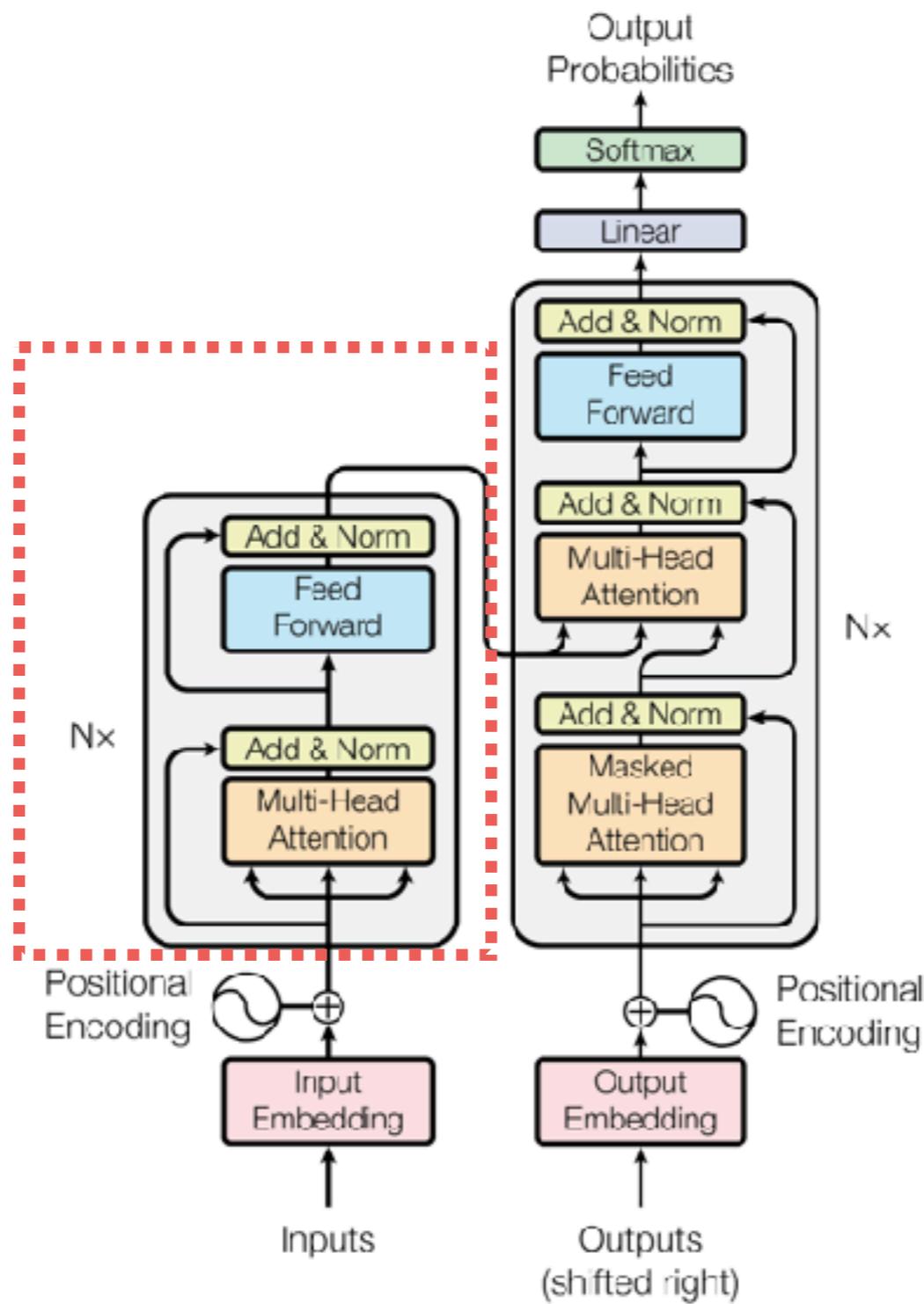
<https://arxiv.org/abs/1706.03762>



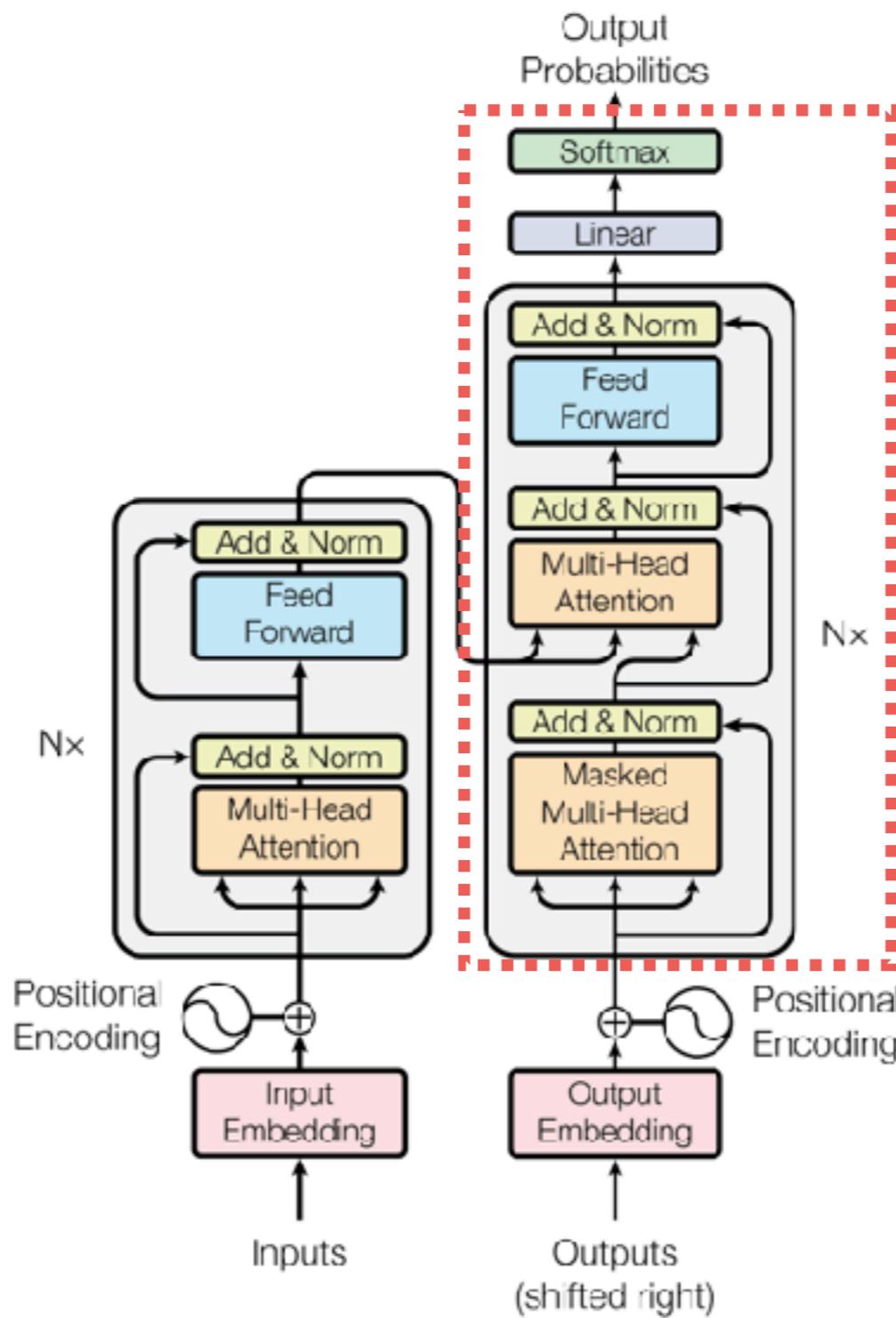
# 1. Pre-processing stage



# 2. Encoding stage



# 3. Decoding stage



# Example with attention

Context-aware embedding vectors

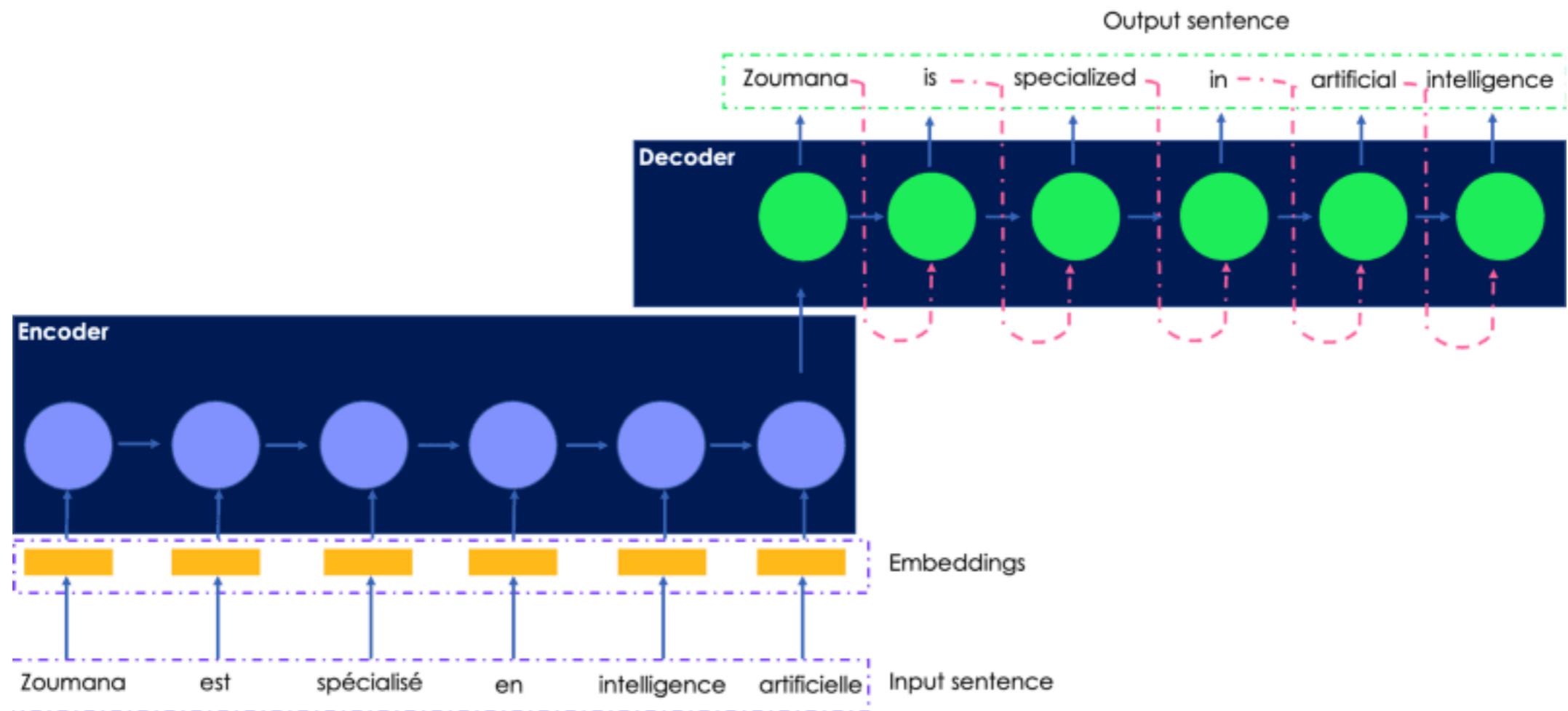
Swing the bat



The **bat** flew at night



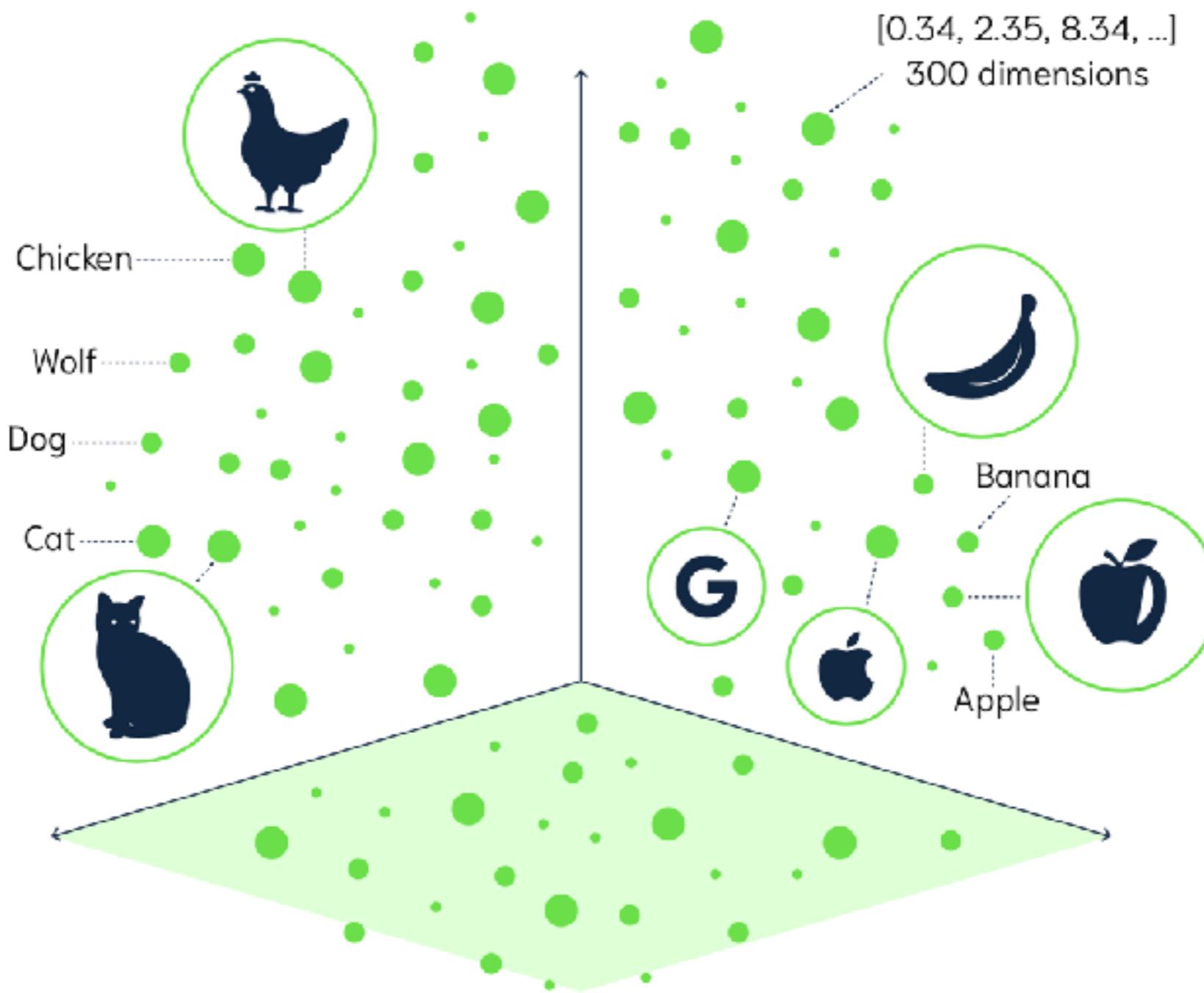
# Example with translation



<https://www.datacamp.com/tutorial/an-introduction-to-using-transformers-and-hugging-face>



# Visual of Vector space



# **Food recommendations ?**

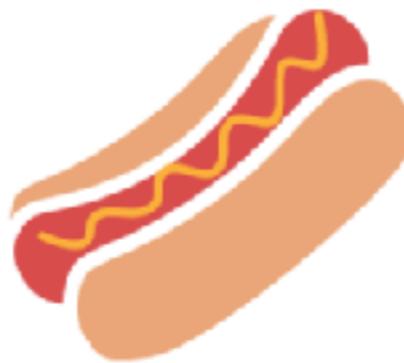


# Food recommendations

borsch



hot dog



salad



pizza



shawarma

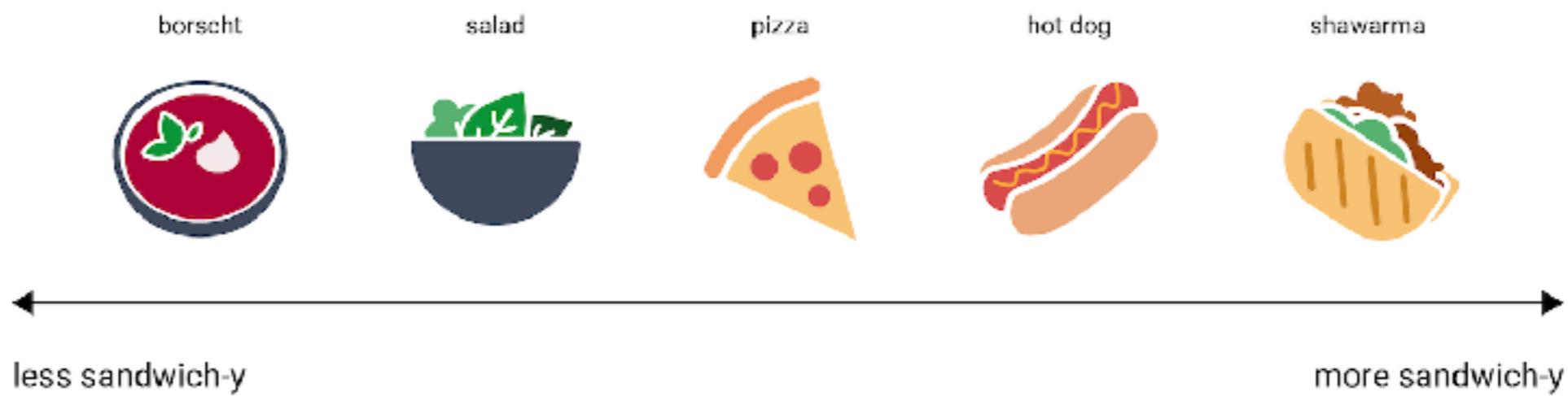


<https://developers.google.com/machine-learning/crash-course/embeddings>



# Embedding space (1)

Less and more sandwich (1D)

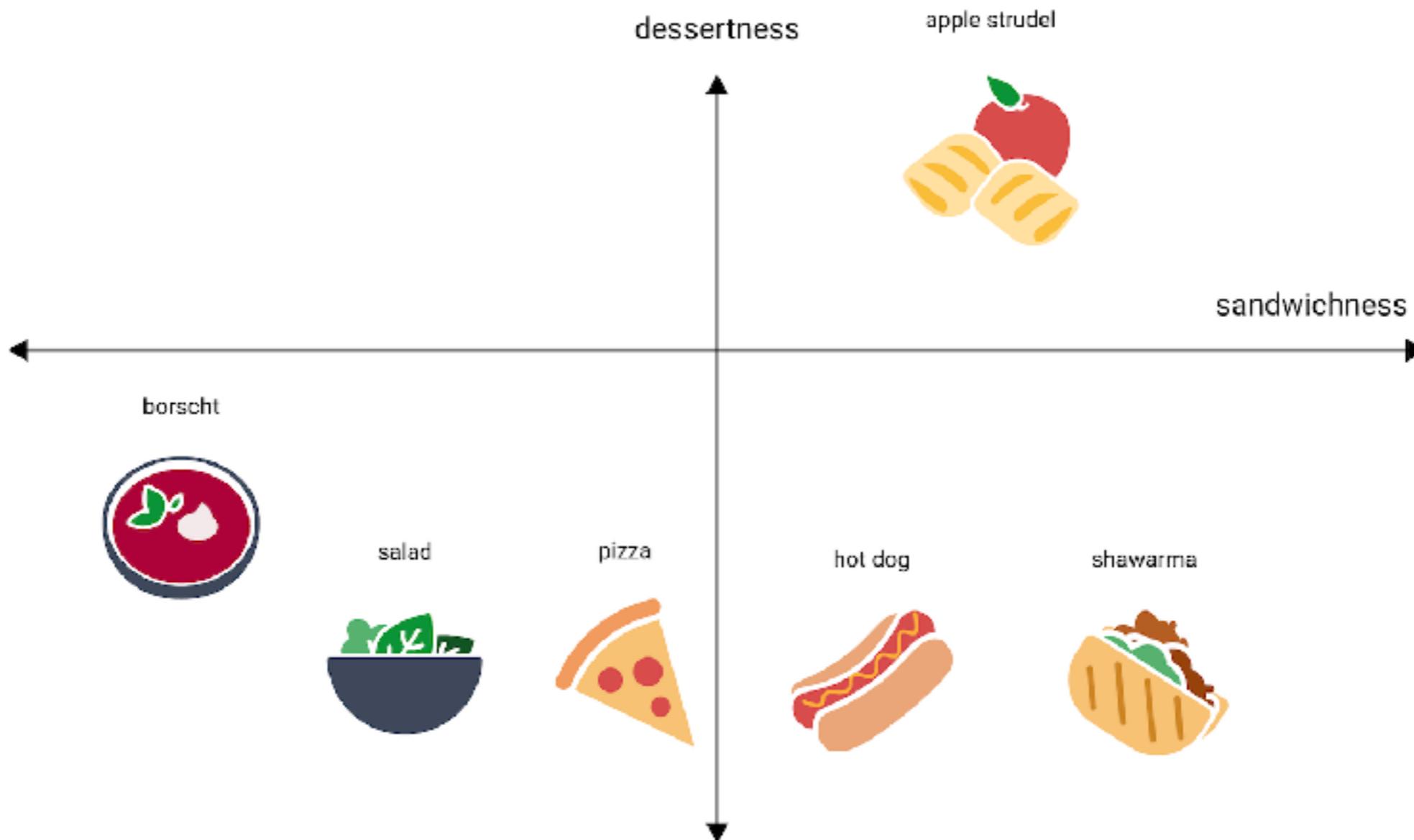


<https://developers.google.com/machine-learning/crash-course/embeddings/embedding-space>



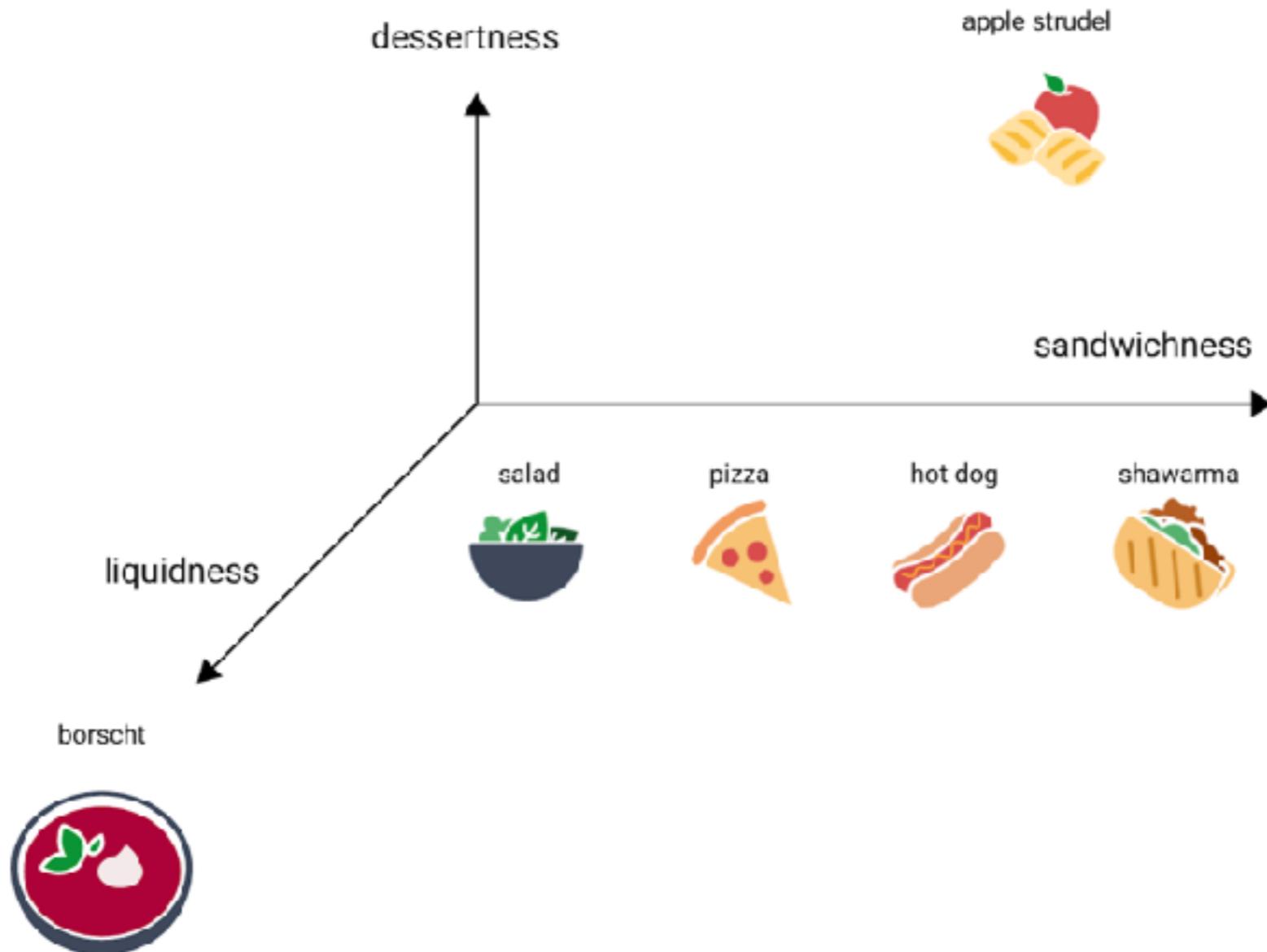
# Embedding space (2)

Sandwichness and dessertness (2D)

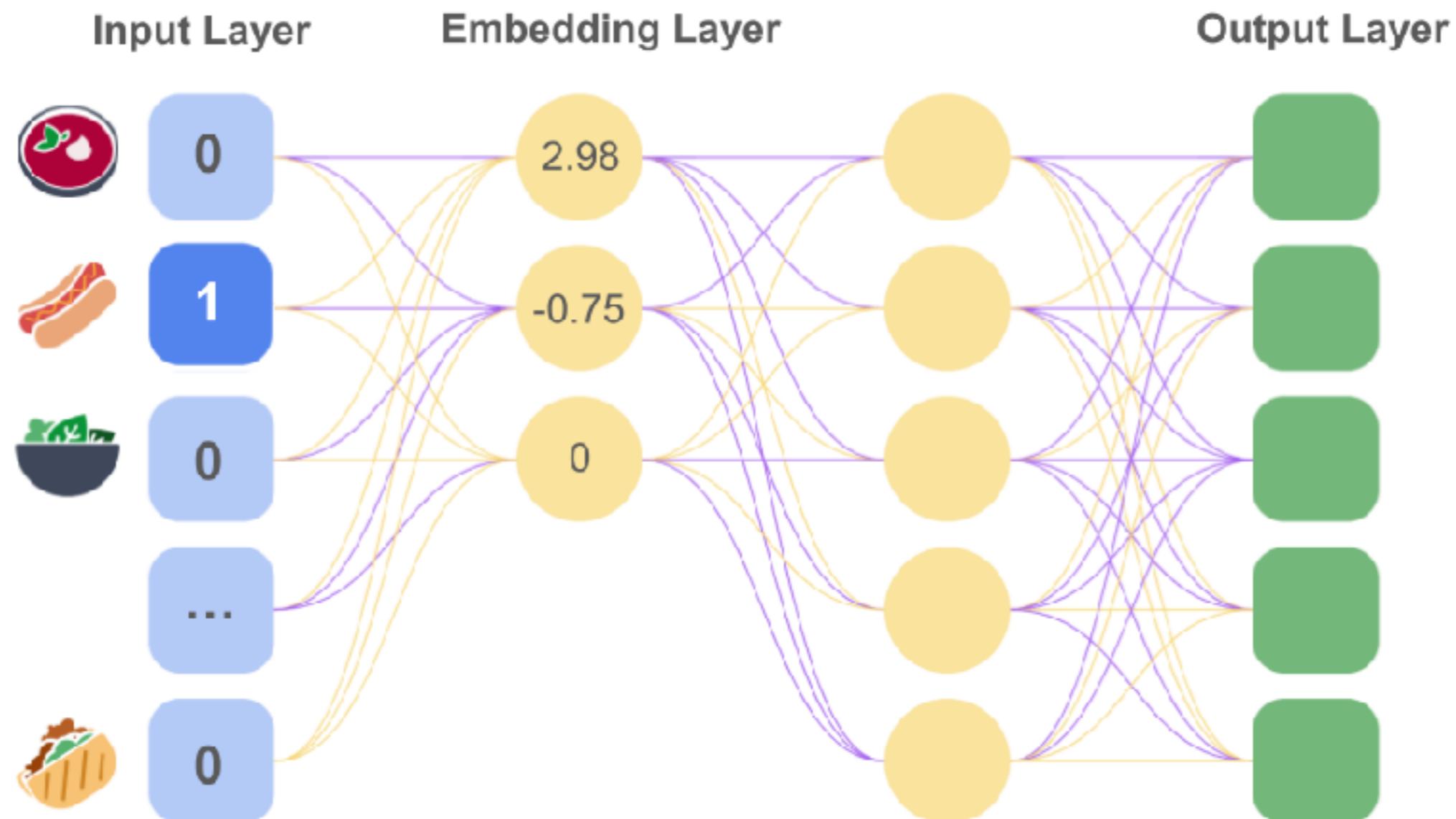


# Embedding space (3)

Sandwichness, dessertness and liquidness (3D)

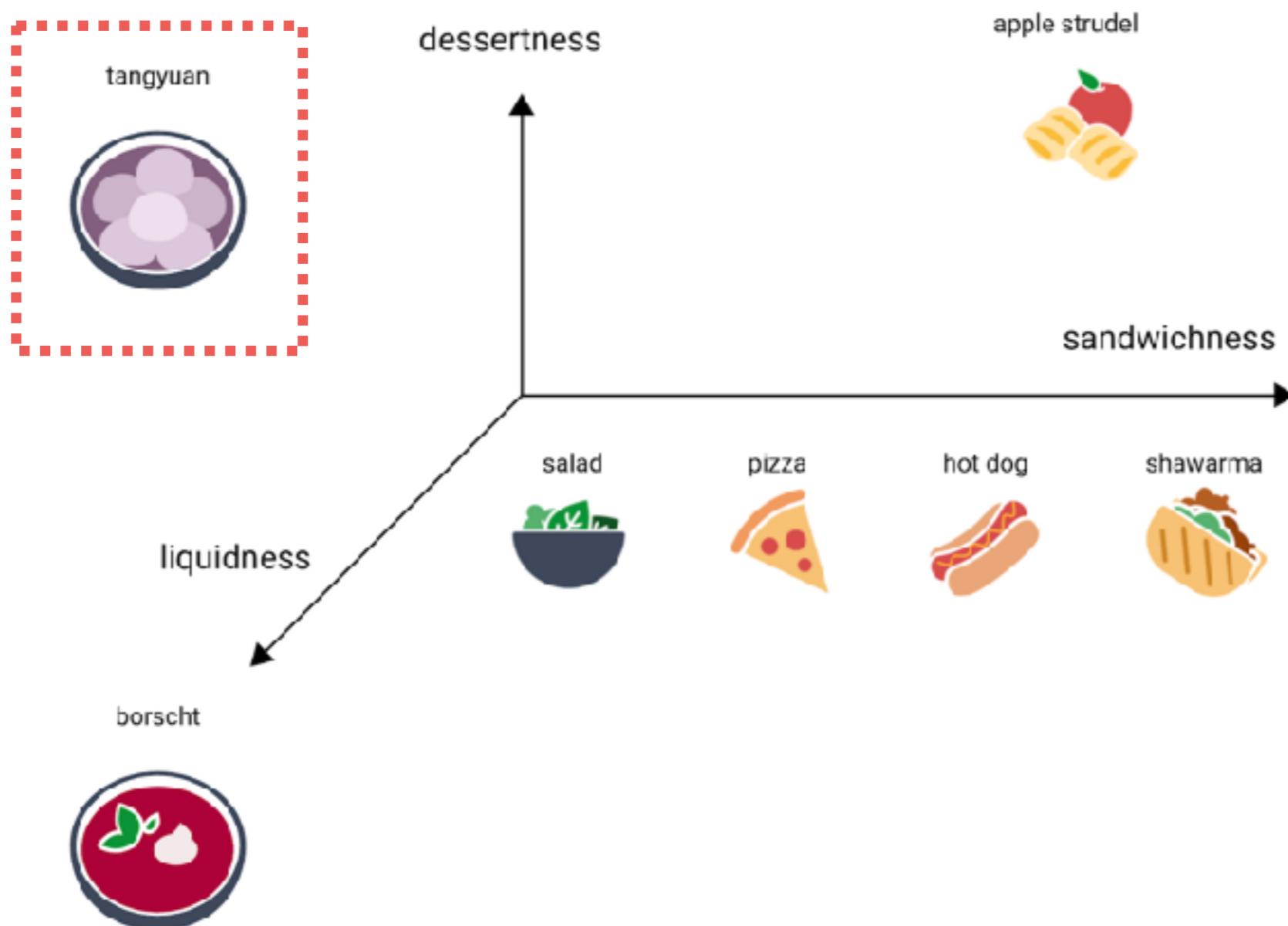


# Embedding process ?



# Embedding space (4)

New food ?





รายการอาหารที่มีผัก กินง่าย ๆ อร่อย ๆ ให้หน่อยสิ ?



# Use cases for Attention

Translation

Text summarization

Question answering

Sentiment analysis

Content generation



# **Choose right model for your use case**



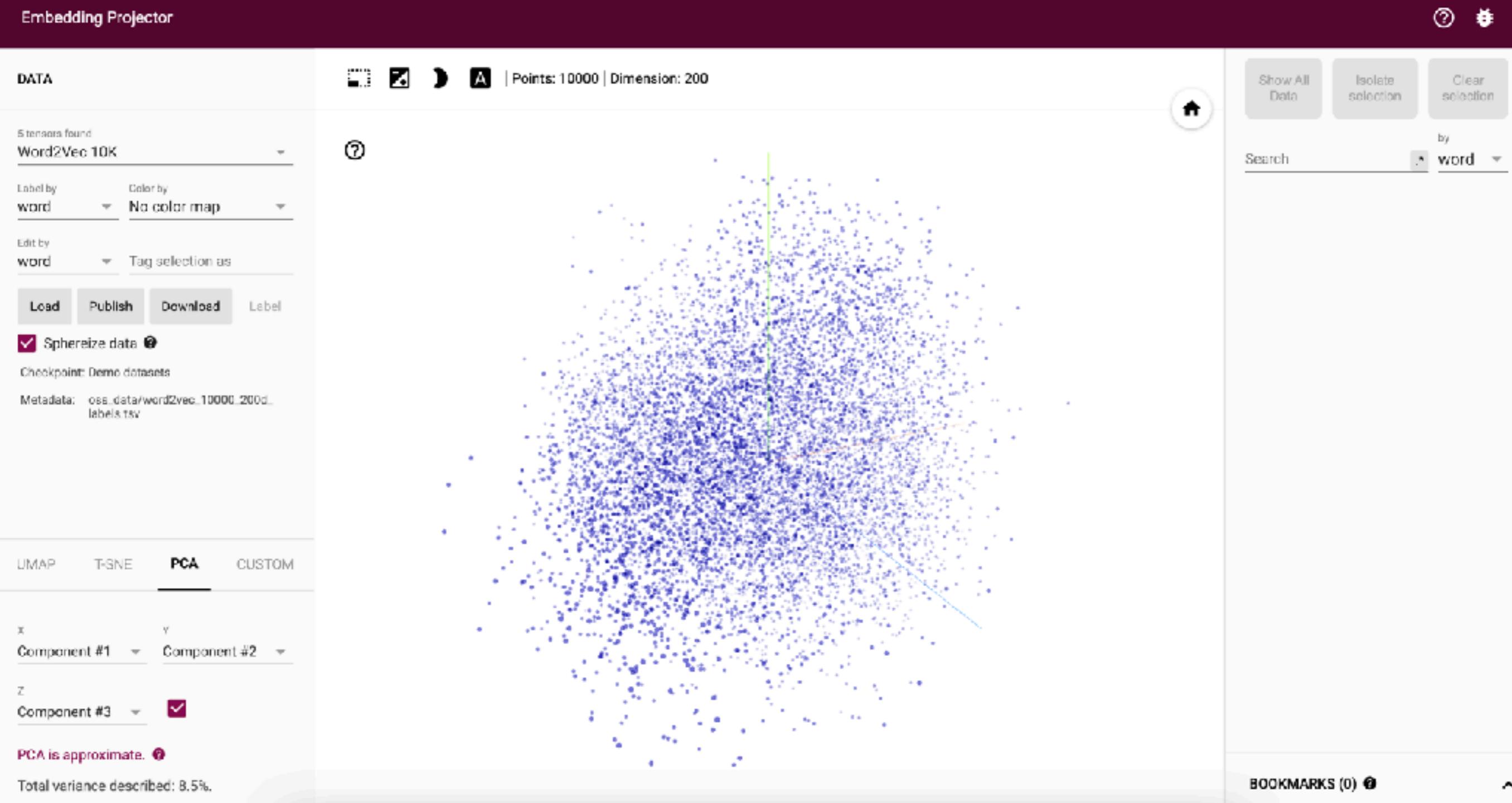
# Embedding models

Model name	Provider name	Dimensions	Max token
text-embedding-3-small	OpenAI	1536	8192
text-embedding-3-large	OpenAI	3072	8192
sentence-transformers/all-MiniLM-L6-v2	Hugging Face	382	256
gemini-embedding	Google	3072	8192

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



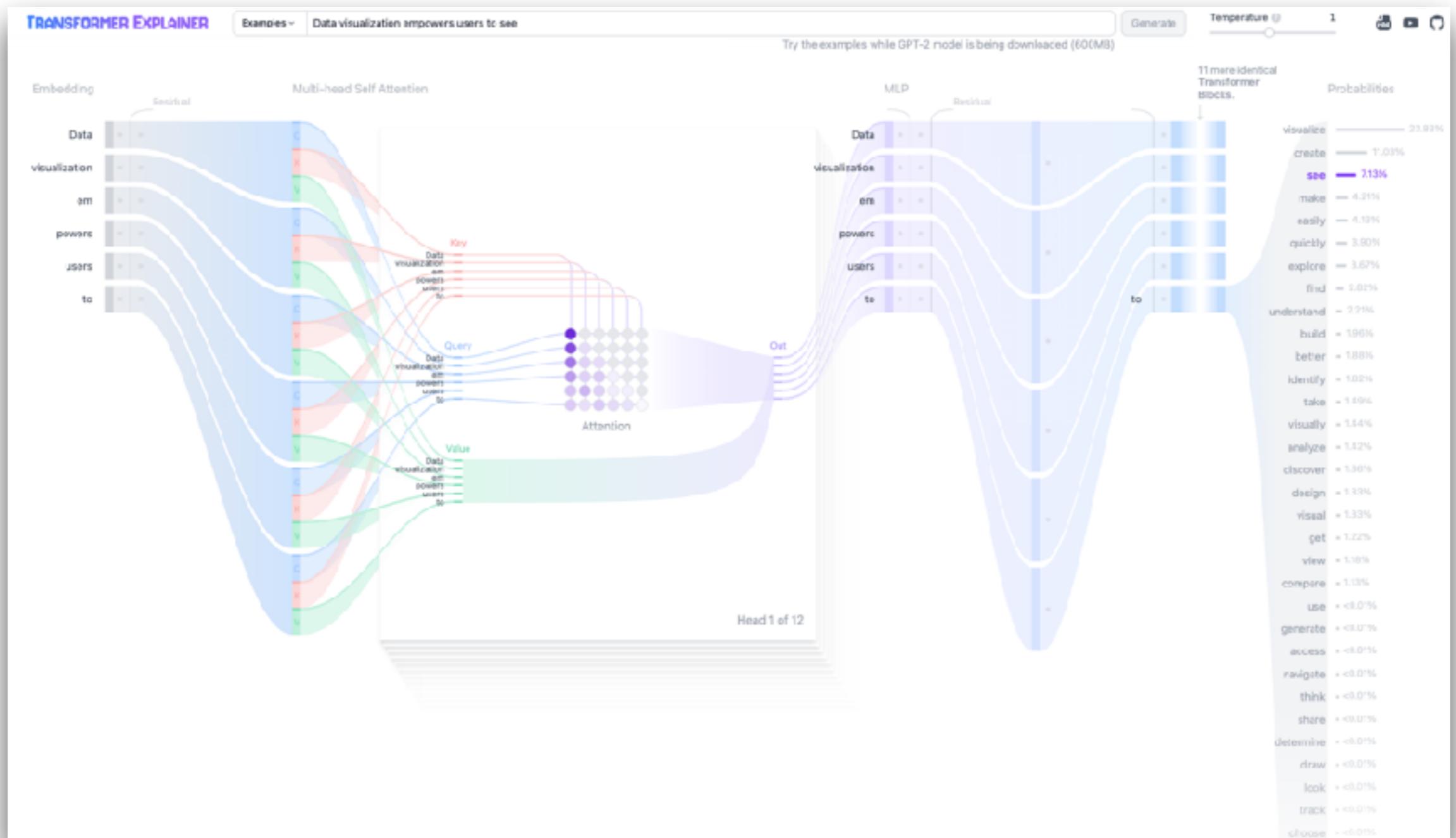
# Embedding Projector



<https://projector.tensorflow.org/>



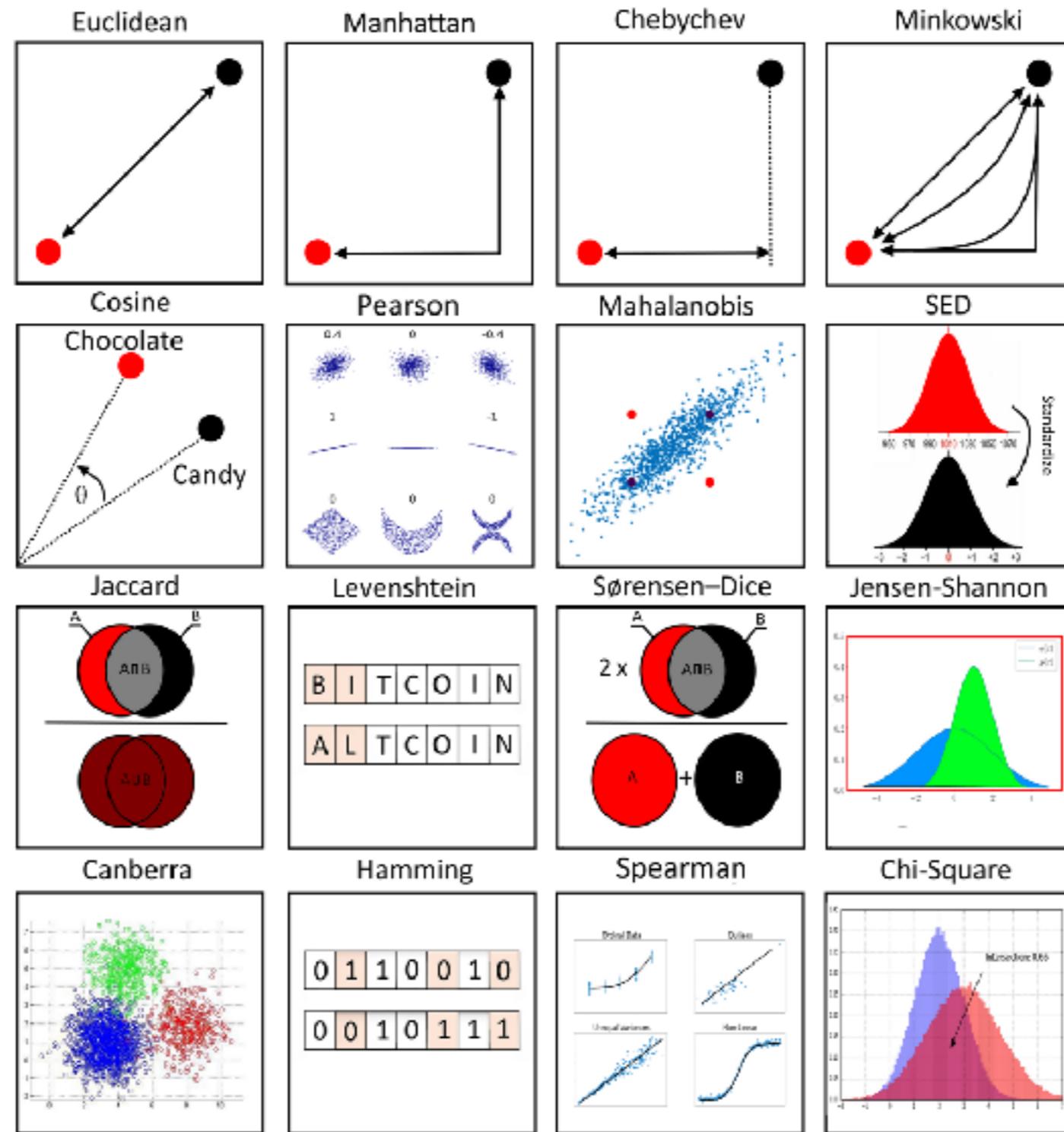
# Transformer Explainer



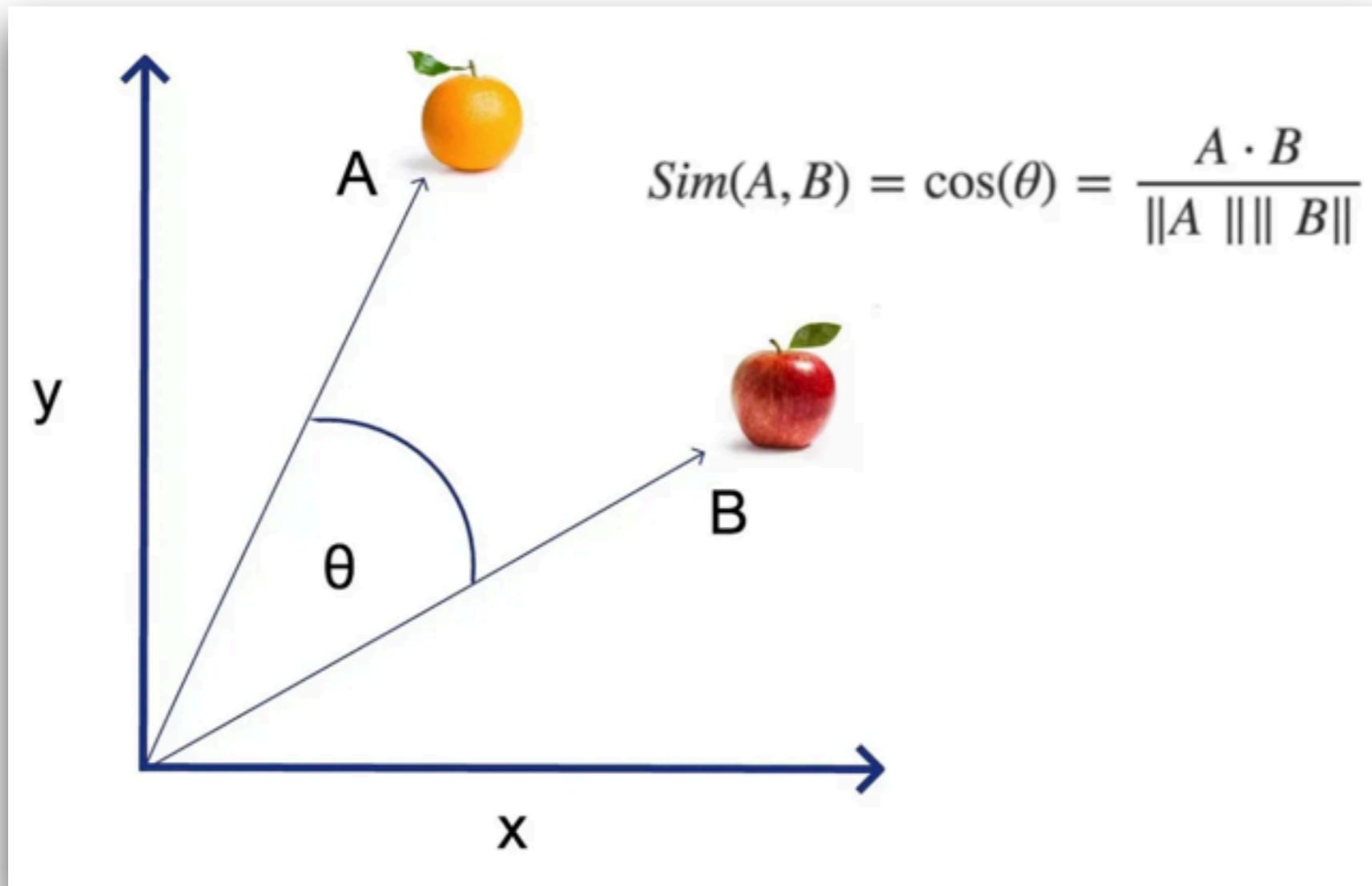
<https://poloclub.github.io/transformer-explainer/>



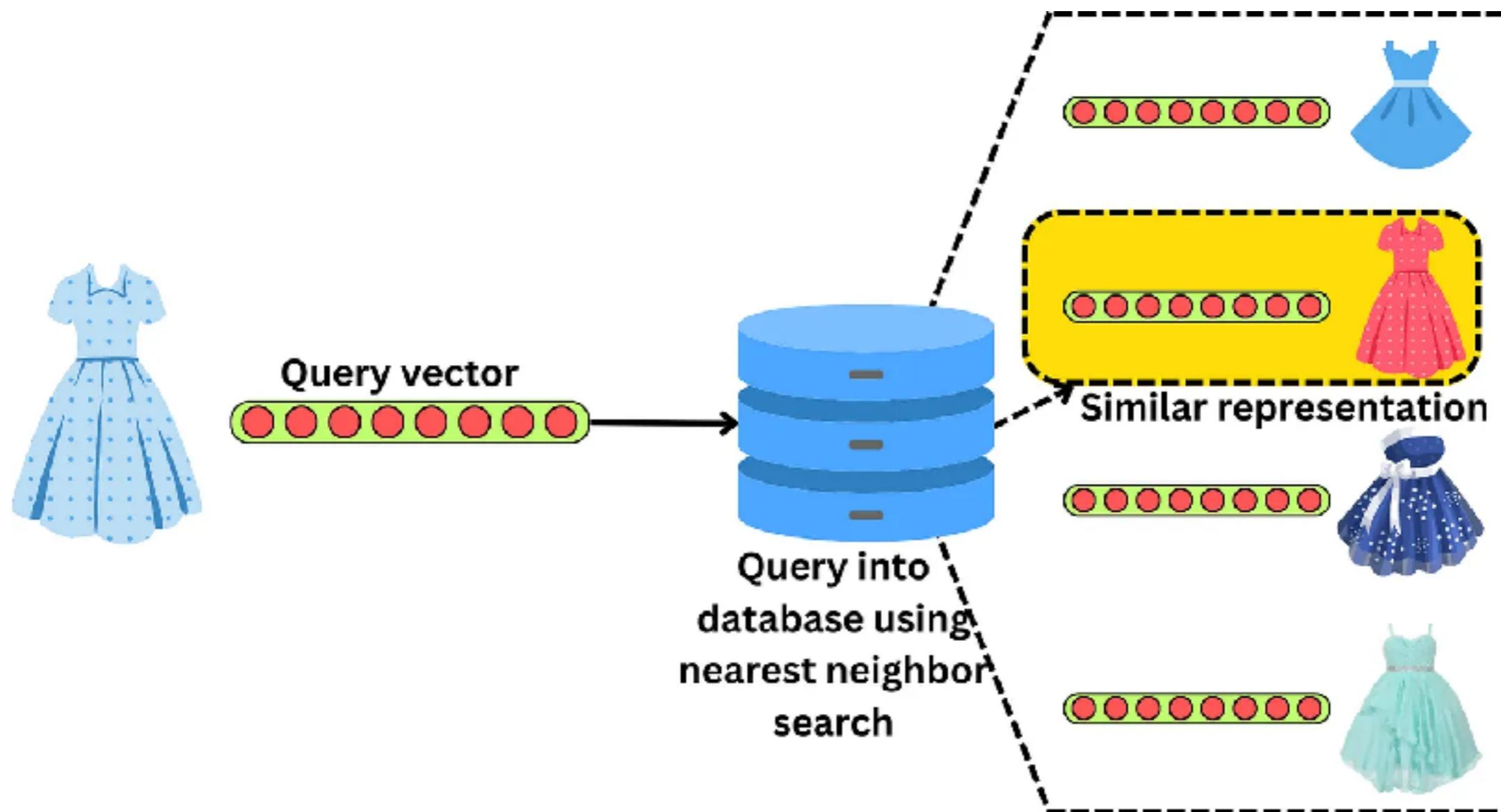
# Distance measure



# Cosine distance/similarity



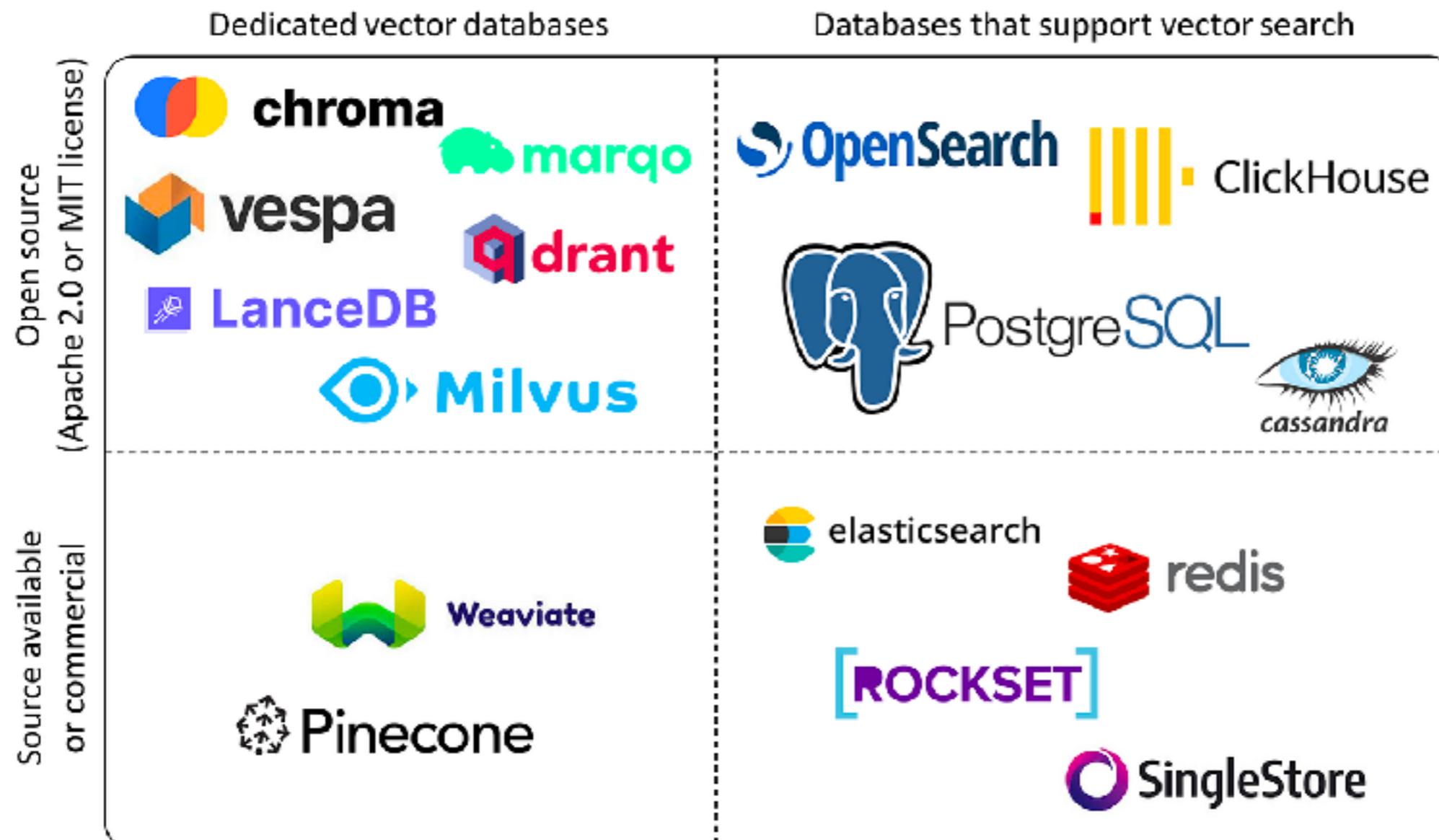
# Search similar items



<https://newsletter.theaiedge.io/p/understanding-how-vector-databases>



# Vector Database



# Vector Database

Capable of organize data by **meaning**

Capable of searching entries by **similarity** to the query

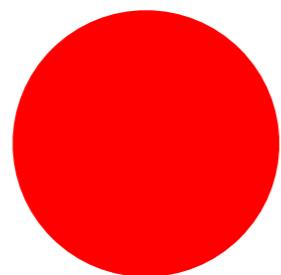
Represent meaning as a series of numbers

Search or filtering by keyword and vector (hybrid)

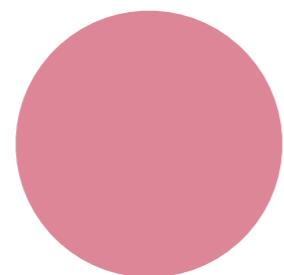
Help to find the **right** information **faster**



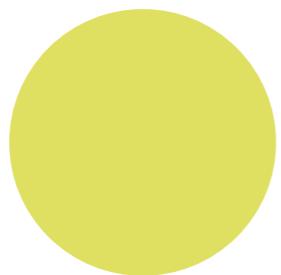
# Example with RGB color



[255, 0, 0]



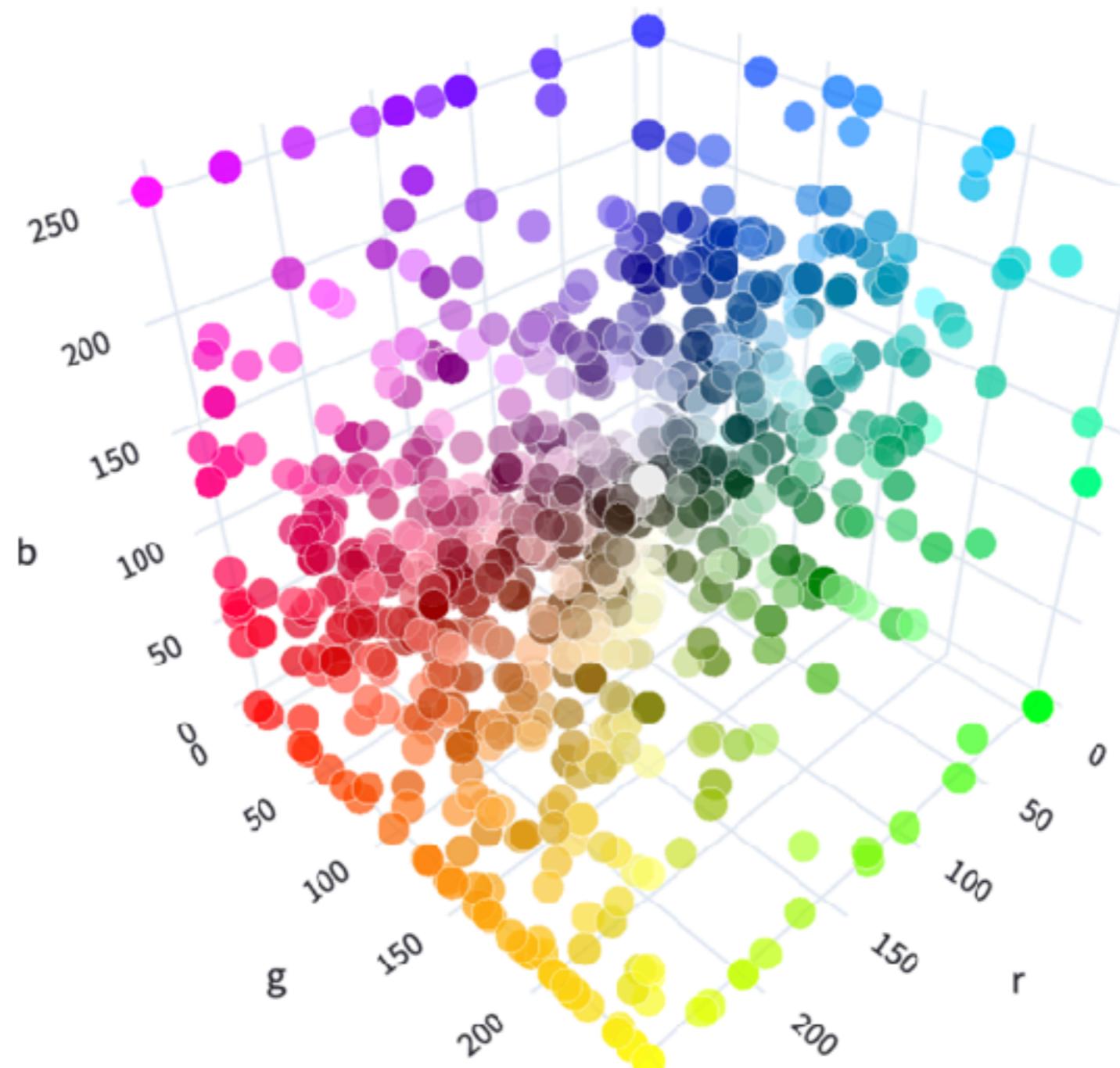
[220, 134, 151]



[223, 223, 97]



# Visualize RGB



[https://huggingface.co/spaces/jphwang/colorful\\_vectors](https://huggingface.co/spaces/jphwang/colorful_vectors)



# Capabilities of LLMs

- Text generation
- Question answering
- Translation
- Summarization
- Conversation
- Sentiment analysis
- Text classification
- Code assistance



# **LLMs in industry**



# LLMs in industry (2024)

Model name	Company
Bidirectional Encoder Representation from Transformers (BERT)	Google AI
Generative Pre-trained transformer-3 (GPT-3)	OpenAI
Generative Pre-trained transformer-4 (GPT-4)	OpenAI
Pathways Language Model-E (PaLM-E)	Google AI
BLOOM	NVIDIA AI
Llama 3	Facebook
Claude 3.7 Sonnet	Anthropic



# LLM Development timeline

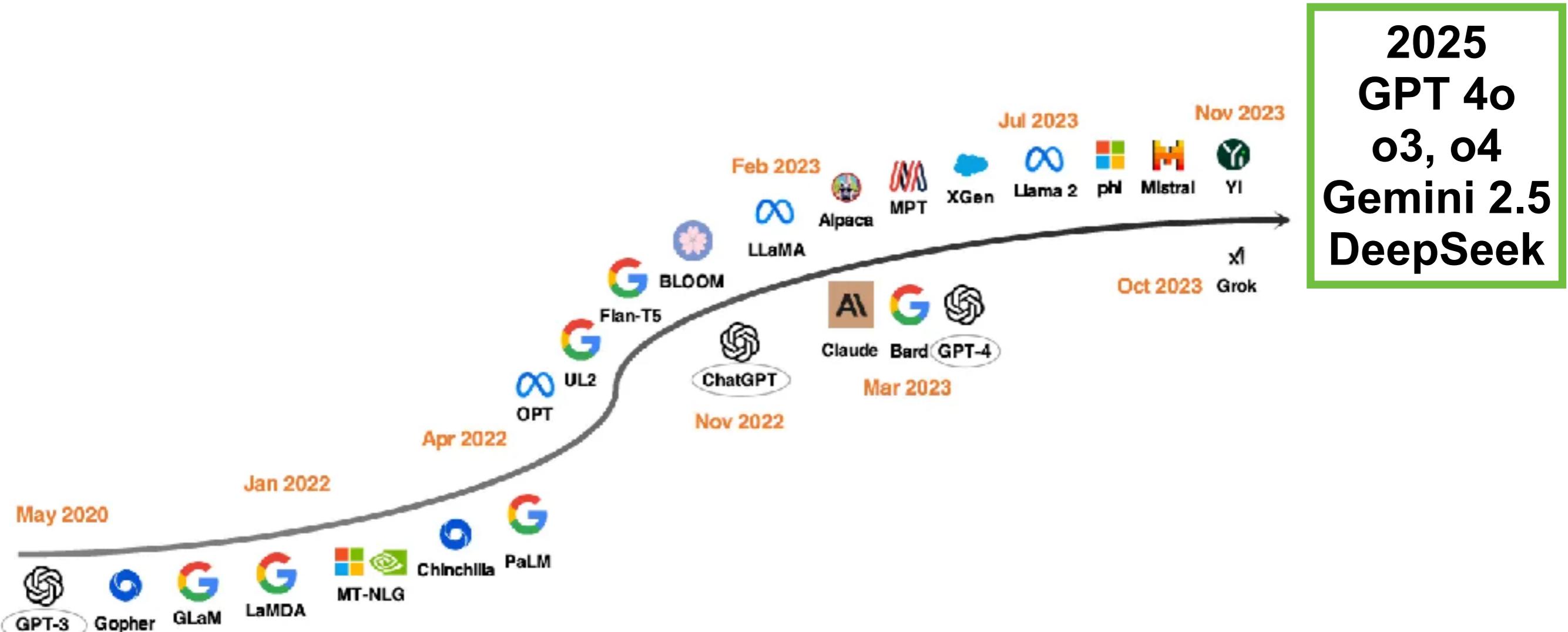
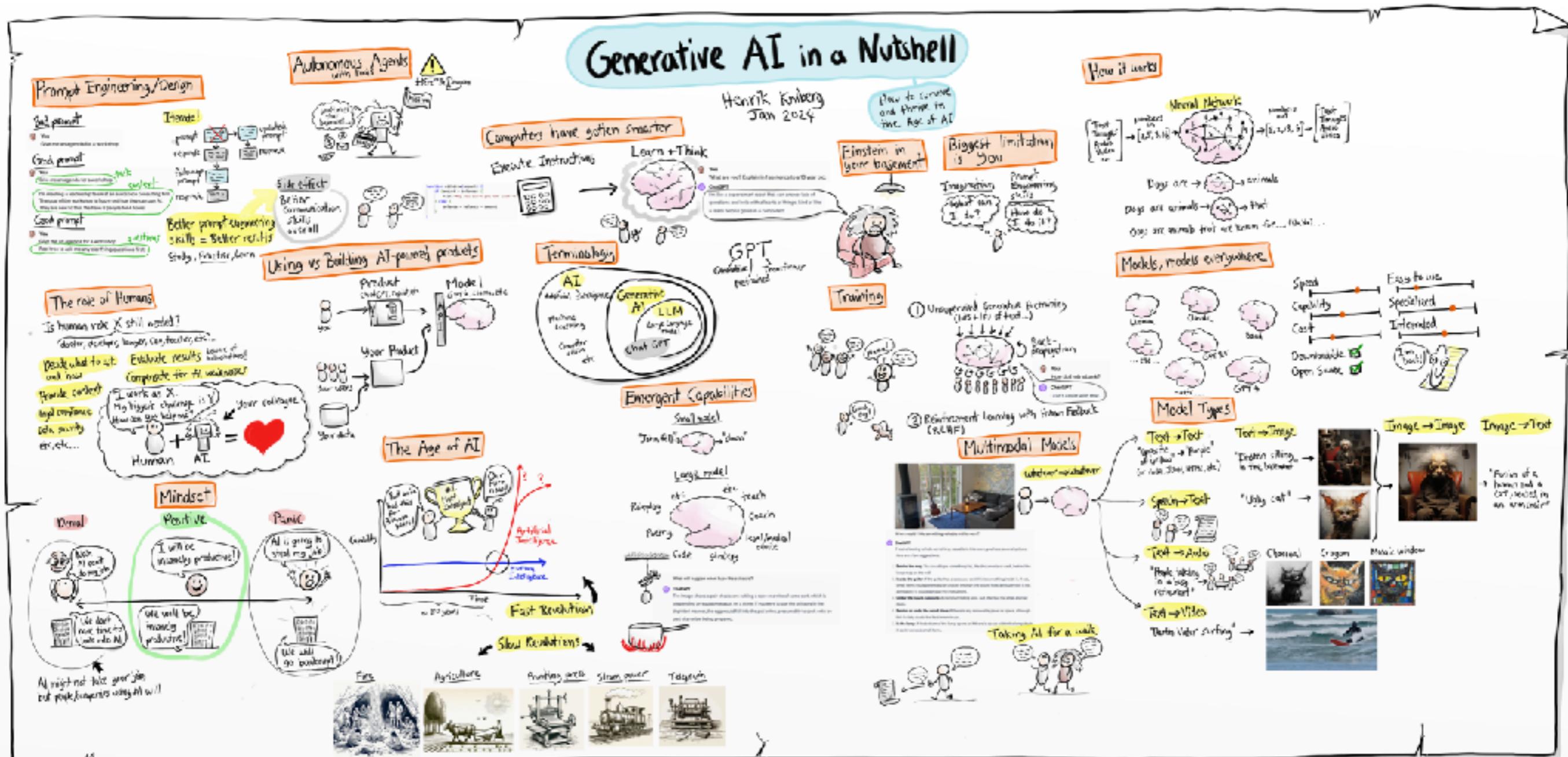


Figure 3: LLM development timeline. The models below the arrow are closed-source while those above the arrow are open-source.

<https://arxiv.org/abs/2311.16989>



# Generative AI in Nutshell



<https://www.youtube.com/watch?v=2IK3DFHRFFw>



# Challenges in Gen AI/LLM

Lack of high-quality data  
Data licenses  
Generation latency  
High computational power

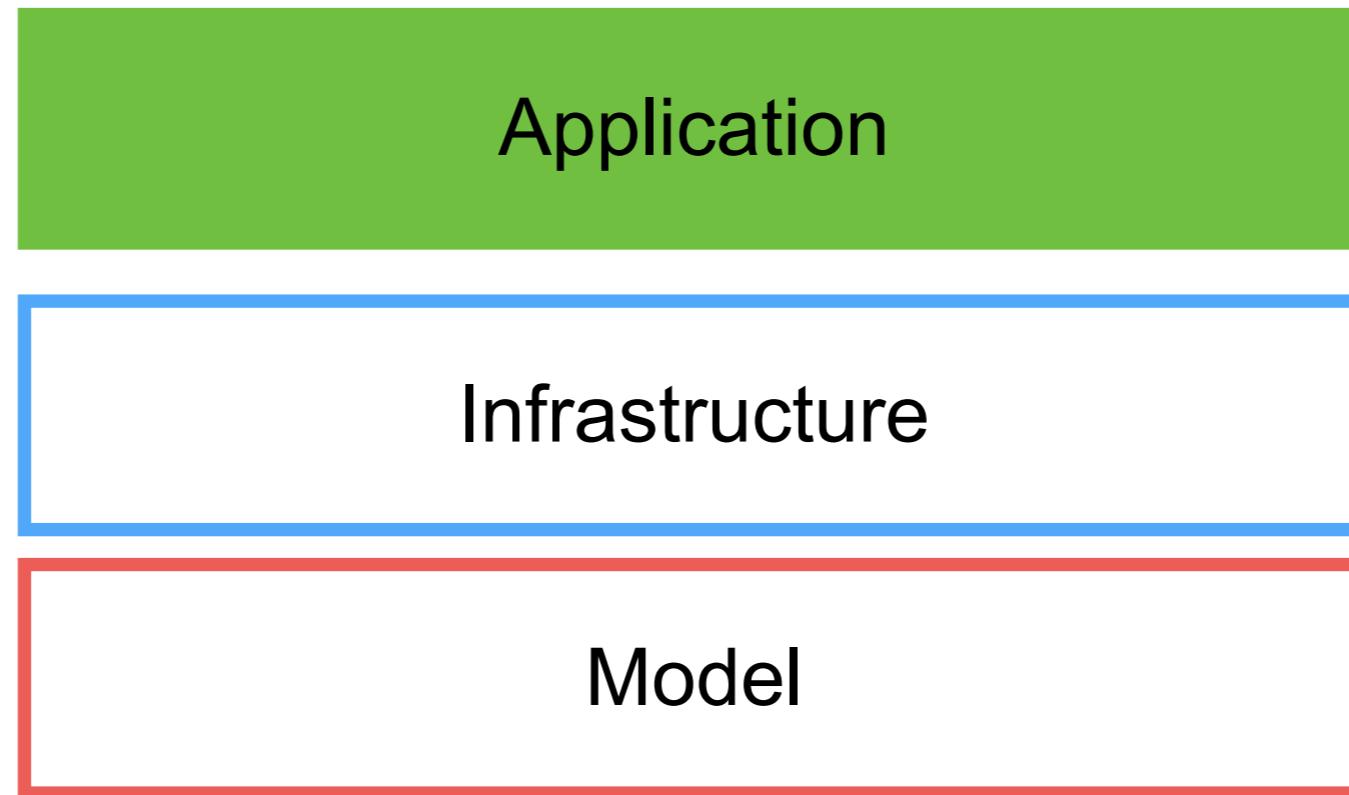
Bias and Ethics

Outdate knowledge

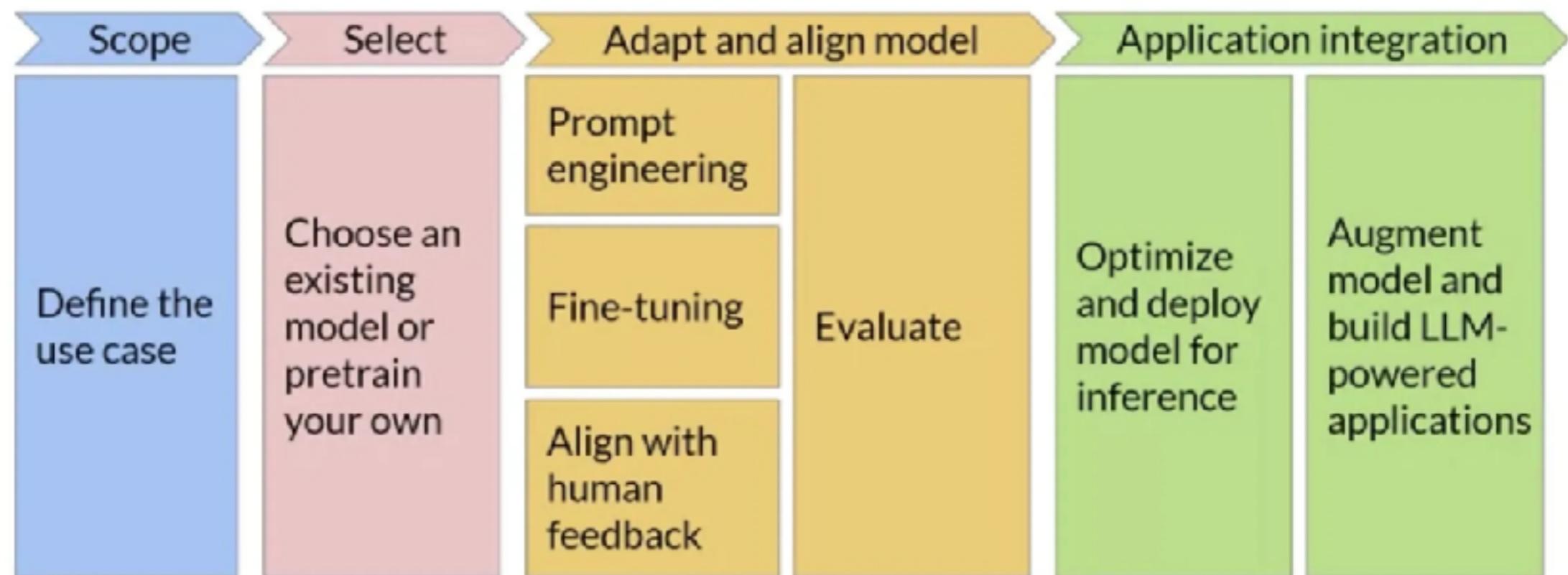
Lack of true understanding



# Challenge in LLMs



# LLM Development Life Cycle

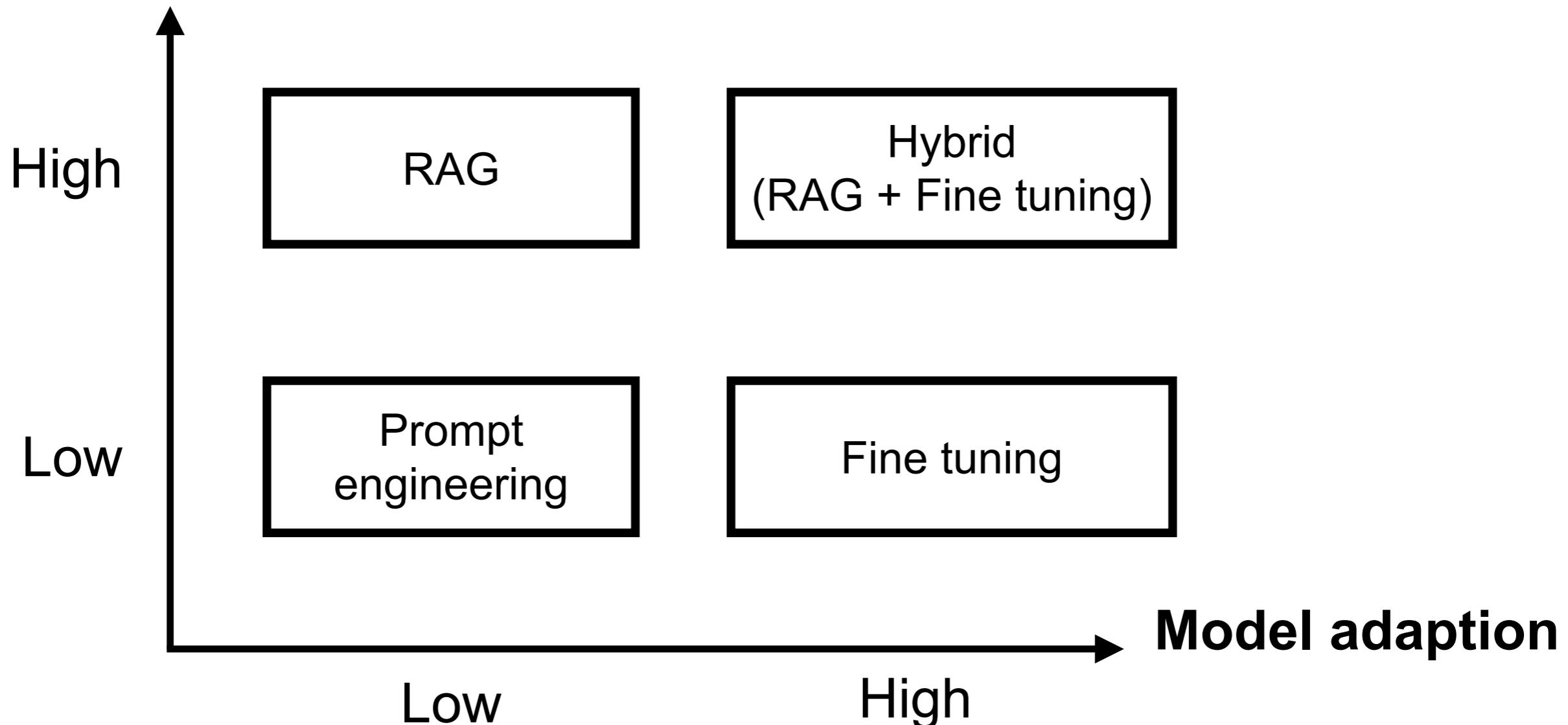


<https://substack.com/home/post/p-147698161>

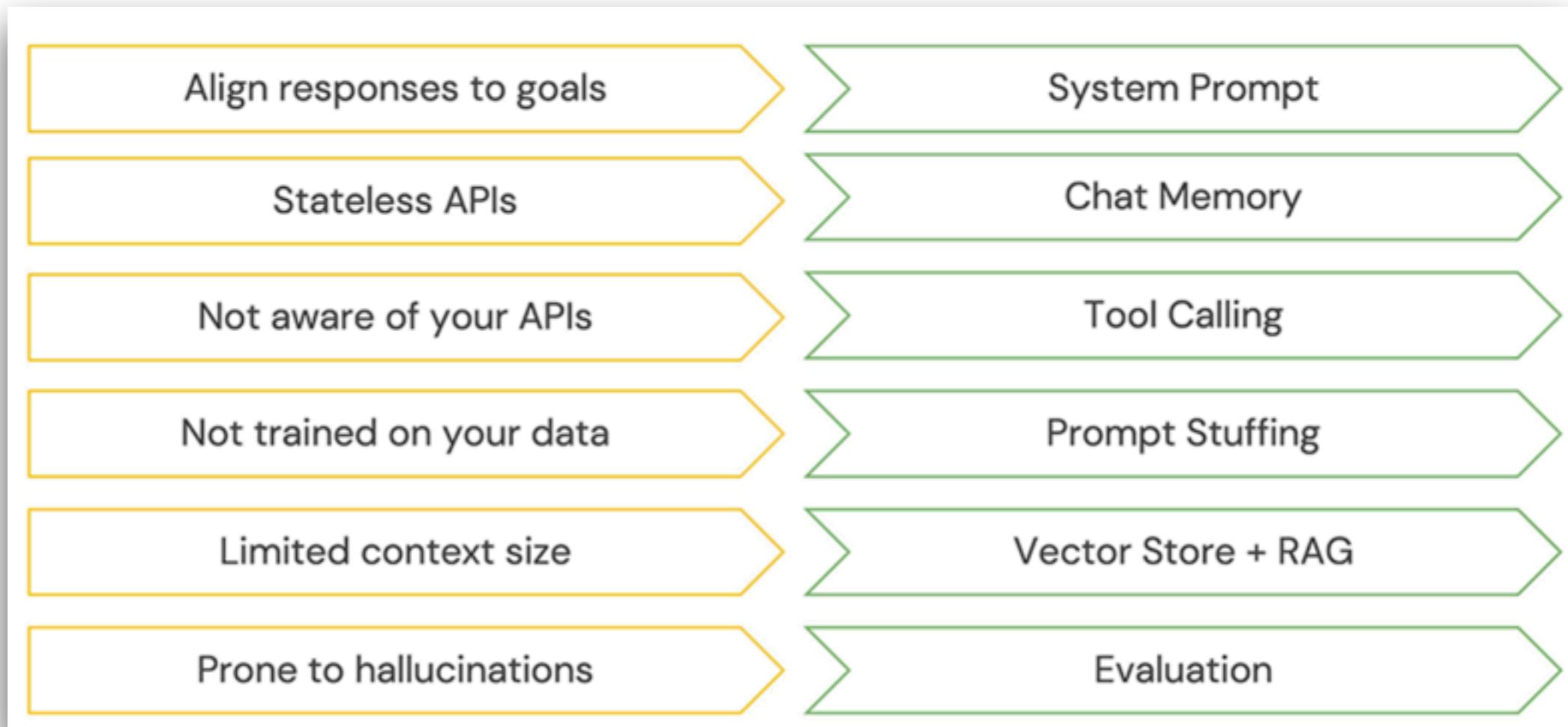


# LLM-based app ?

External knowledge



# Generative AI patterns !!



# Decision factor for LLMs ?

Decision factor	Description
Language and Domain support	Supports Thai language or specialized domains (e.g., finance, legal, medical)
Data ownership	Level of control over data local deployment vs. third-party cloud APIs
Cost and scalability	Open-source may require GPU investment API-based models charge per usage
Latency and SLA	Response time and whether there's an uptime guarantee or service-level agreement
Ecosystem support	SDKs, plug-ins, monitoring tools, and active developer community
Advance features	Support for features like function calling, tool use, and extended context windows

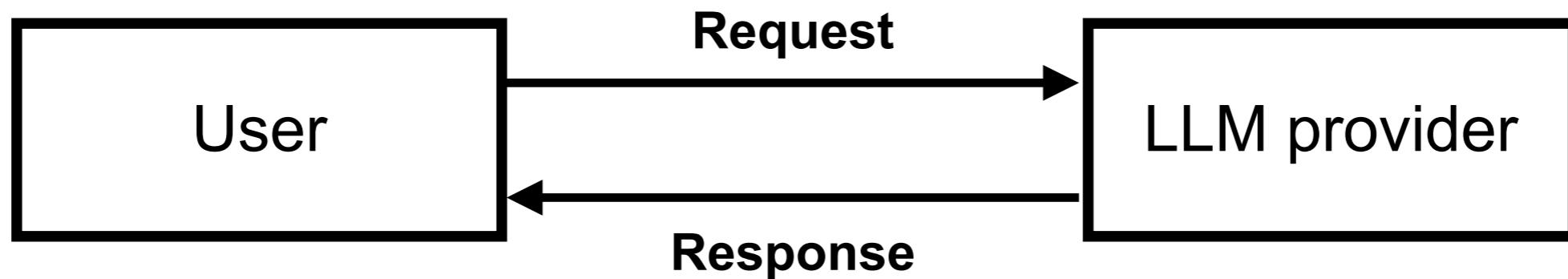


# Working with LLM

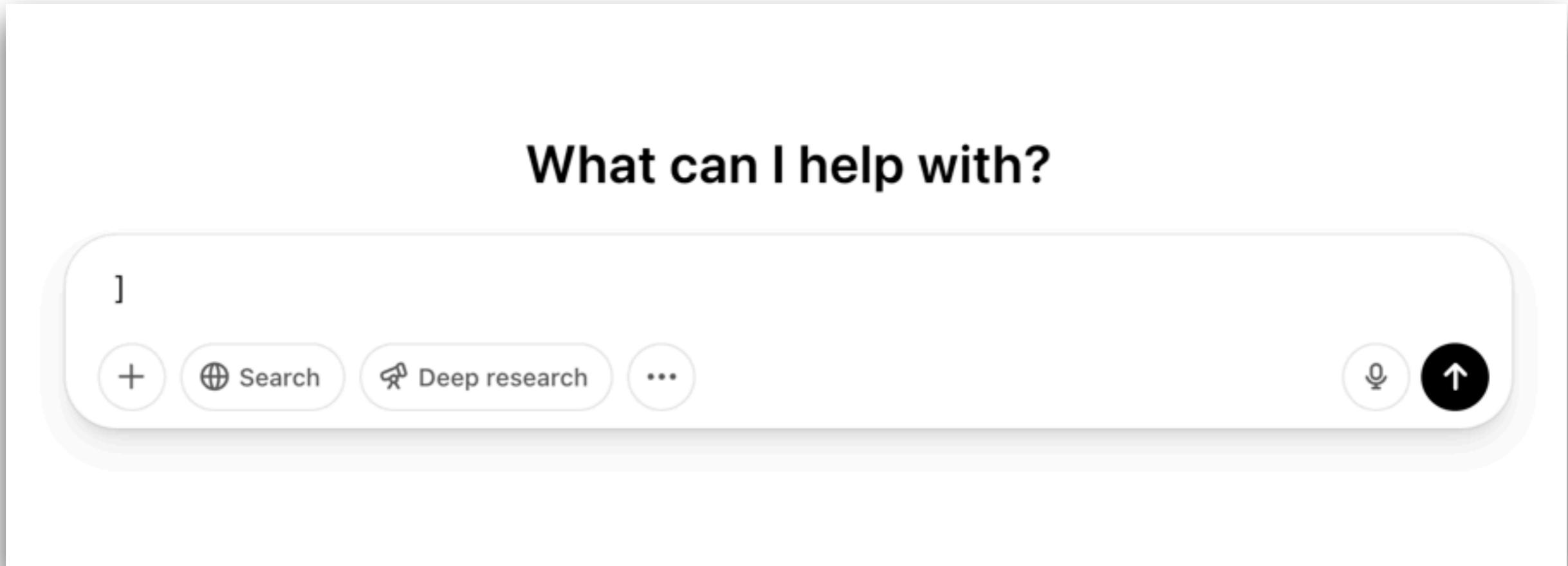


# Working with LLM provider

Prompting and prompt engineering



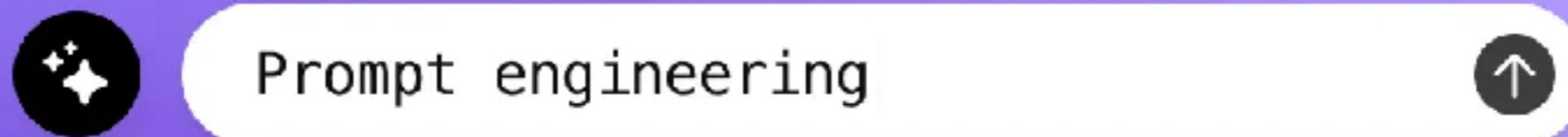
# ChatGPT (web)



<https://chatgpt.com/>



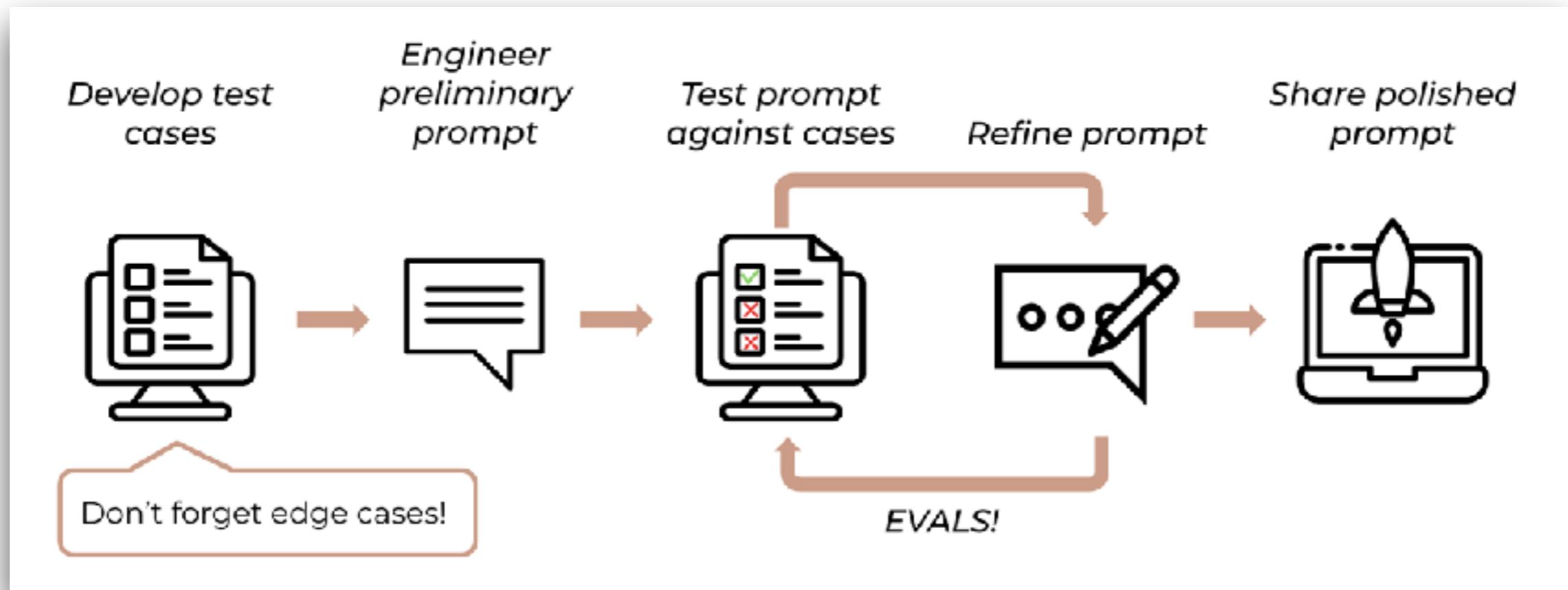
# Prompt Engineering



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



# Prompt Development Life cycle

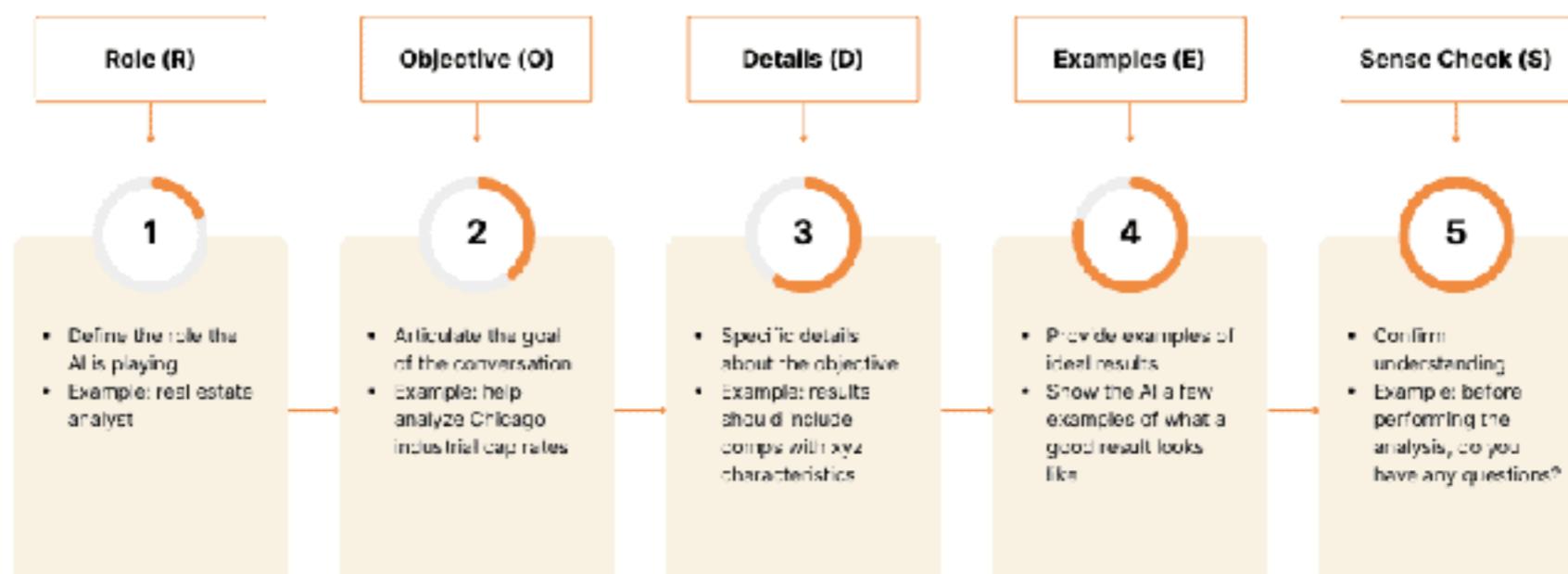


<https://cheatsheet.md/claude/clause-prompt-engineering.en>



# Prompt Engineering

## R.O.D.E.S. Framework AI Prompt Engineering



[https://x.com/sebo\\_gm/status/1687366385620721664](https://x.com/sebo_gm/status/1687366385620721664)



# Chain of Thought (CoT)

Guide LLM to breakdown complex reasoning tasks  
into small steps for human solving

Solving problem with logic, calculation and decision  
making

Improve  
accuracy

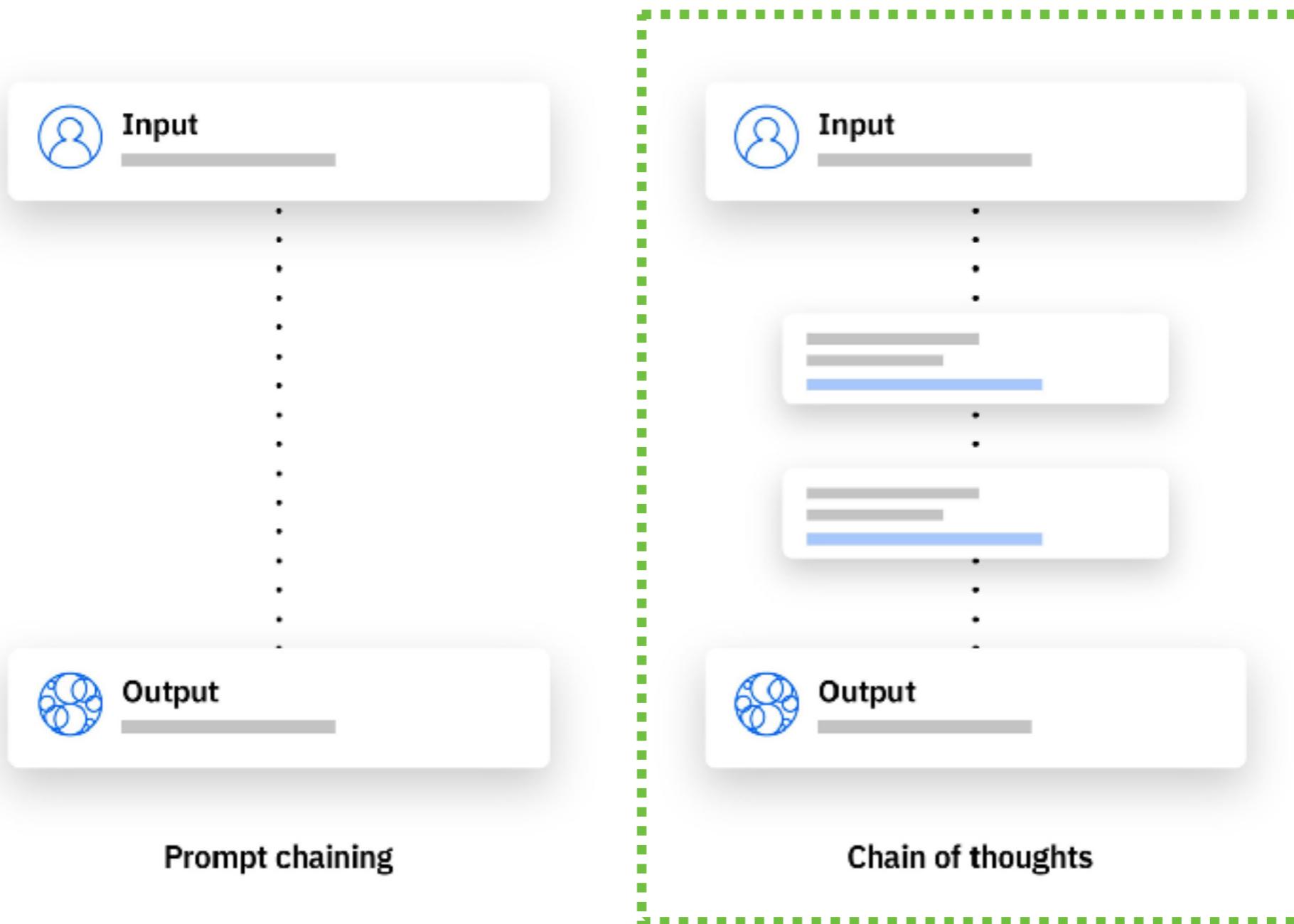
Enhance  
transparency

Better  
reasoning  
capability

<https://arxiv.org/abs/2201.11903>



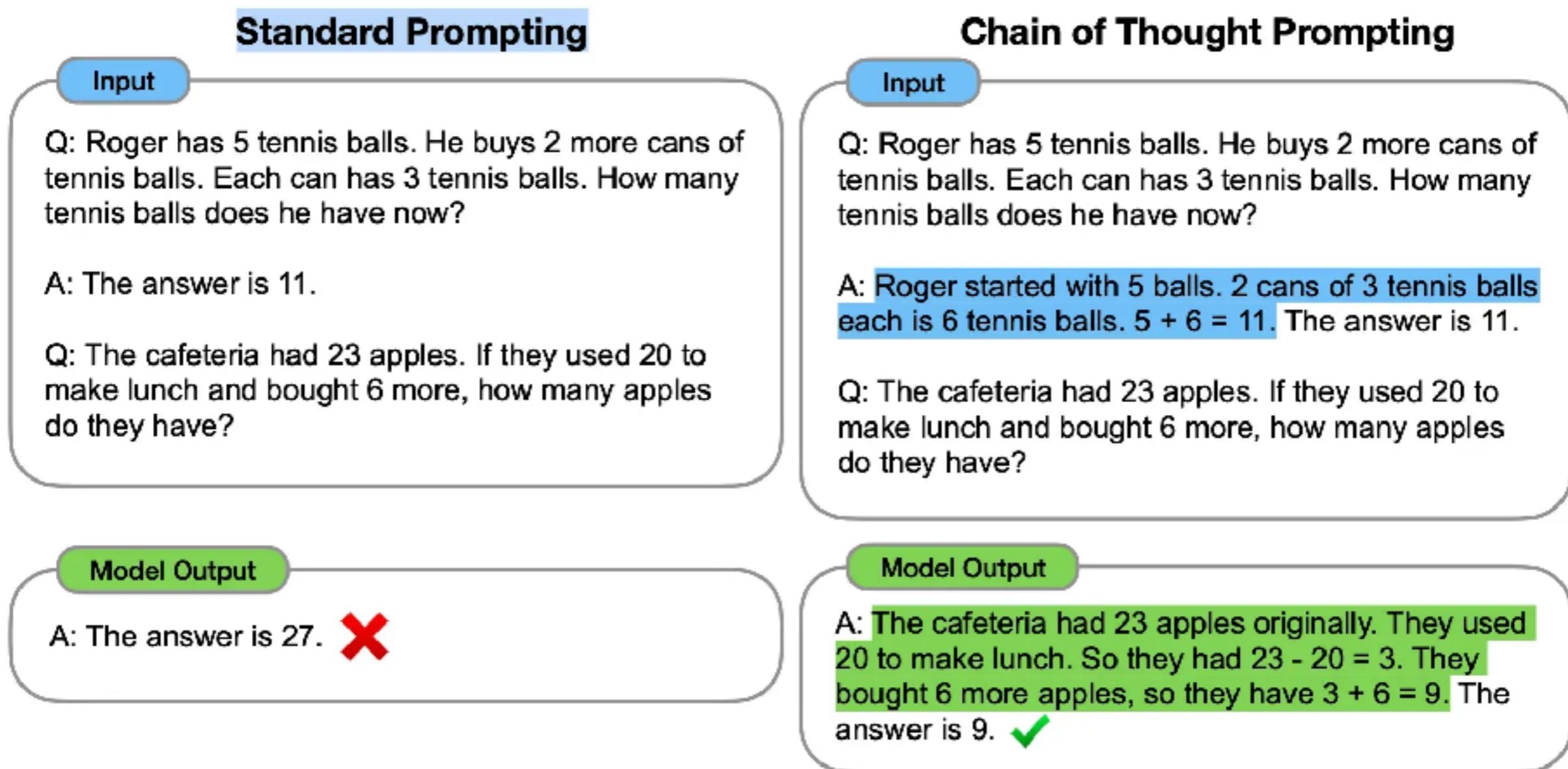
# Chain of Thought (CoT)



<https://www.ibm.com/think/topics/chain-of-thoughts>



# Chain of Thought (CoT)



<https://arxiv.org/abs/2201.11903>



# CoT with Zero-shot

## Break down tasks by LLM model

(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The answer is 8. ✘

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are  $16 / 2 = 8$  golf balls. Half of the golf balls are blue. So there are  $8 / 2 = 4$  blue golf balls. The answer is 4. ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 ✘

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

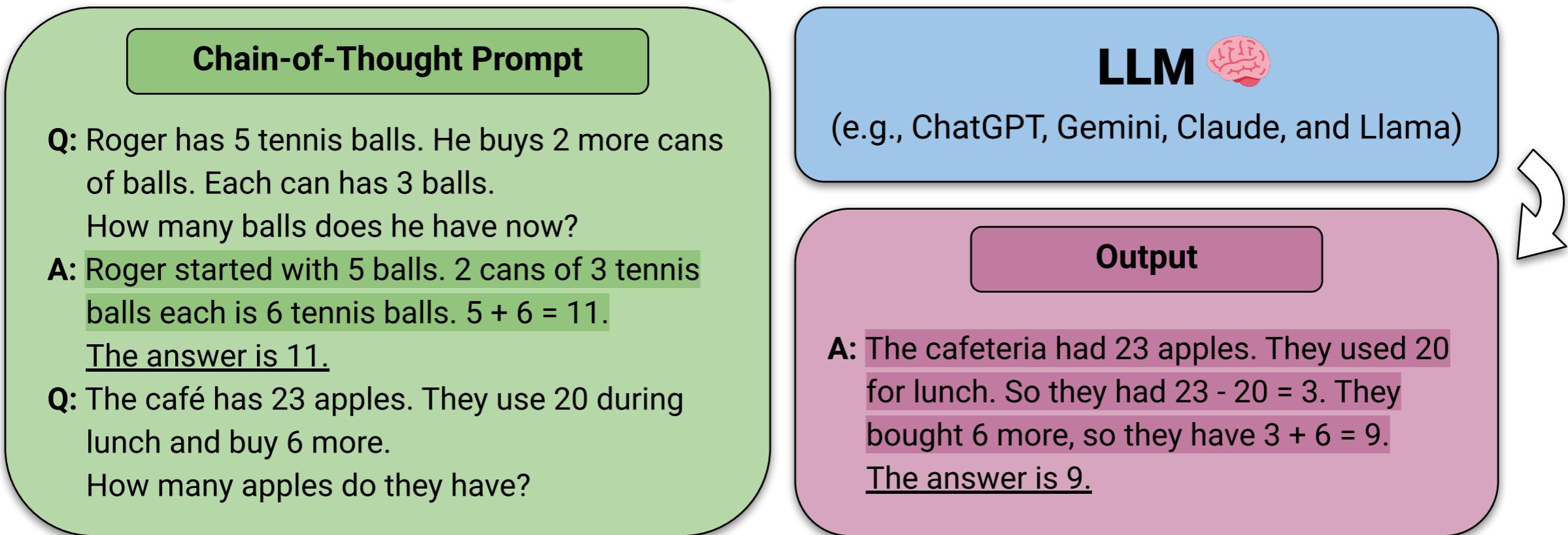
A: Let's think step by step.

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

<https://arxiv.org/abs/2201.11903>



# CoT to LLM Reasoning



<https://github.com/atfortes/Awesome-LLM-Reasoning>

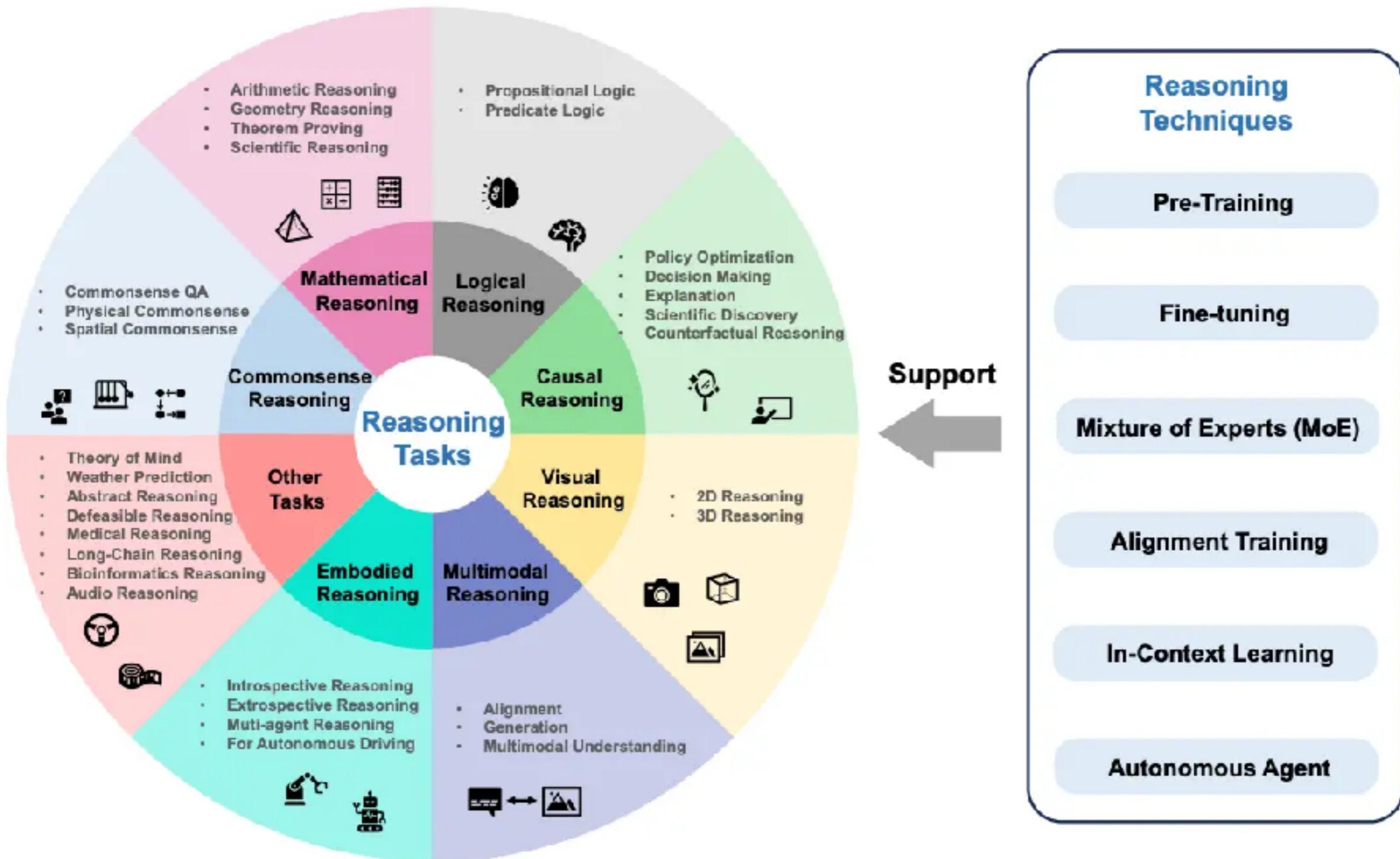


# LLM Reasoning models



Deepseek R1

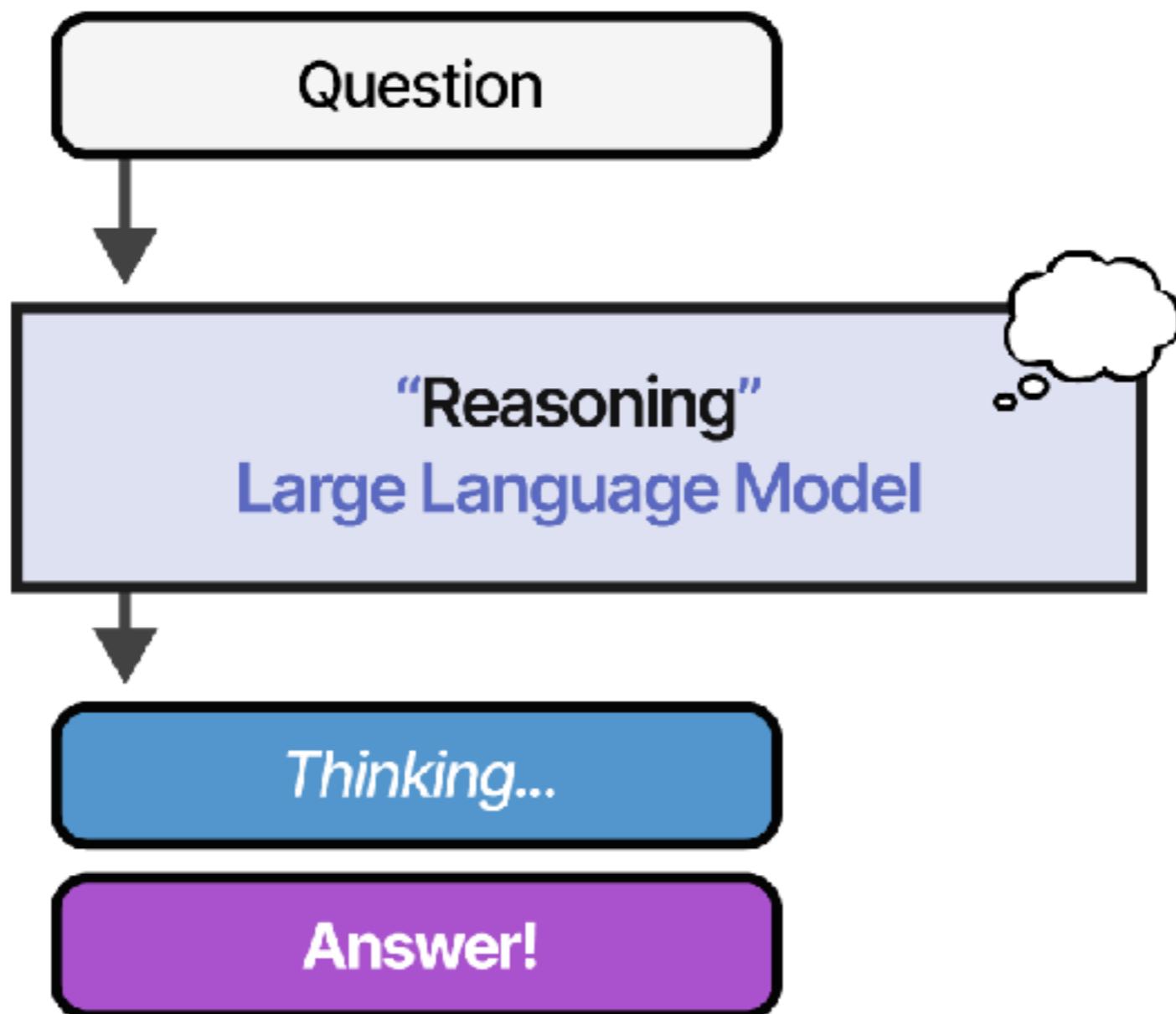




<https://www.promptingguide.ai/research/llm-reasoning>



# LLM Reasoning



<https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-reasoning-langs>



# LLM Reasoning

Question

I have **10** apples. I gave **2** apples away. I ate **1**. How many do I have?

“Reasoning”

Large Language Model

You have **10** apples

You gave **2** away and have **8** left

You ate **1** and have **7** left

You have **7** apples

reason steps

(typically Chain-of-Thought)

final answer

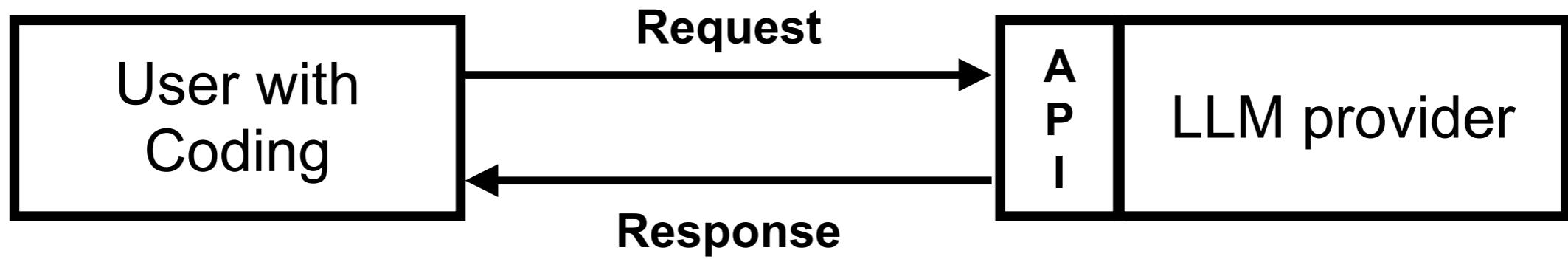
<https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-reasoning-langs>



# Working with APIs



# Working with APIs



<https://platform.openai.com/docs/overview>



# Problems ?



# Problems ?

Hallucinations

Limit of context window size

Lack of access to non-public data

Limit knowledge of test data of models



# Hallucinations ?

Hallucinations in LLMs refer to the generation of content that is irrelevant, made-up, or inconsistent with the input data

Incorrect  
information

Lower trust in  
model



# Causes and Types of LLMs Hallucination



## Types of LLMs Hallucination

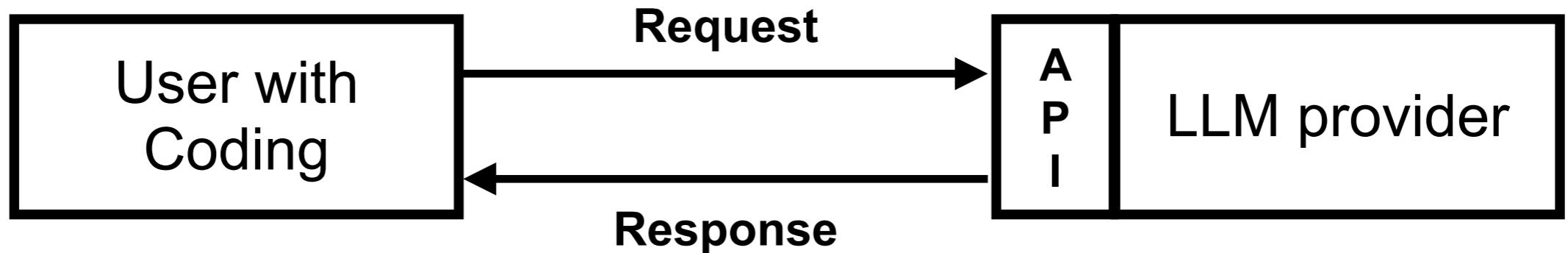
- ← Sentence Contradiction
- ↖↗ Prompt Contradiction
- ✗ Factual Contradiction
- ←➡ Nonsensical Output
- ⟳ Irrelevant or Random Hallucinations



# Context window size ?

Amount of text a model can process and remember at once time

Measure with **token**



# Example of GPT 4.1

The screenshot shows the GPT-4.1 model page. At the top, there's a blue rounded rectangle with "GPT-4.1" in white. To its right is the text "Default" with a dropdown arrow, and a small square icon. Below this is the text "Flagship GPT model for complex tasks". To the right are two buttons: "Compare" and "Try in Playground".

Below this is a horizontal bar with five sections: "INTELLIGENCE" (four filled circles, labeled "Higher"), "SPEED" (three filled arrows pointing right, labeled "Medium"), "PRICE" (\$2 - \$8, "Input + Output"), "INPUT" (text and image icons), and "OUTPUT" (text and image icons).

The main content area starts with a paragraph: "GPT-4.1 is our flagship model for complex tasks. It is well suited for problem solving across domains." To the right of this paragraph is a red dashed box containing three bullet points: "1,047,576 context window", "32,768 max output tokens", and "Jun 01, 2024 knowledge cutoff".

Below this is a "Pricing" section. It says "Pricing is based on the number of tokens used. For tool-specific models, like search and computer use, there's a fee per tool call. See details in the [pricing page](#)". There are three boxes: "Text tokens" (Input: \$2.00, Cached input: \$0.50, Output: \$8.00), "Per 1M tokens + Batch API price" (a toggle switch), and a "Batch API price" section.

<https://platform.openai.com/docs/models/gpt-4.1>



# Example of Llama 4

Llama 4:  
Leading Multimodal Intelligence

Newest model suite offering unrivaled speed and efficiency

**Llama 4 Behemoth**

288B active parameters, 16 experts  
2T total parameters

The most intelligent teacher model for distillation

[Preview](#)

**Llama 4 Maverick**

17B active parameters, 128 experts  
400B total parameters

Native multimodal with 1M context length

[Available](#)

**Llama 4 Scout**

17B active parameters, 16 experts  
109B total parameters

Industry leading 10M context length  
Optimized inference

[Available](#)

<https://ai.meta.com/blog/llama-4-multimodal-intelligence/>



# **Large context window size is new trend !!**



# Context window size of models

Different models may have different window size

Model name	Context window size
GPT 4.1	1M
Gemini 1.5, 2.5	1M
Llama 4	1M
Claude Sonnet 3.7	200,000
o4	200,000



# Large context window size is new trend

**But come with Cost !!**

Money

Resources

Response time



# Example of GPT 4.1

The screenshot shows the GPT-4.1 model page. At the top, there's a blue rounded rectangle containing the text "GPT-4.1" and "Default". Below it is a subtext: "Flagship GPT model for complex tasks". To the right are two buttons: "Compare" and "Try in Playground".

Below this, there's a horizontal bar with five sections: "INTELLIGENCE" (four filled circles, labeled "Higher"), "SPEED" (three lightning bolt icons, labeled "Medium"), "PRICE" (\$2 - \$8, Input - Output), "INPUT" (document, image, code icons, labeled "Text, Image"), and "OUTPUT" (document, image, code icons, labeled "Text").

The main content area starts with a paragraph: "GPT-4.1 is our flagship model for complex tasks. It is well suited for problem solving across domains." To its right are three icons with text: a diamond icon for "1,047,576 context window", a right-pointing arrow for "32,768 max output tokens", and a square icon for "Jun 01, 2024 knowledge cutoff".

A red dashed-line box encloses the "Pricing" section. Inside, it says "Pricing is based on the number of tokens used. For tool-specific models, like search and computer use, there's a fee per tool call. See details in the [pricing page](#)". It also includes "Text tokens" and a toggle switch for "Per 1M tokens + Batch API price".

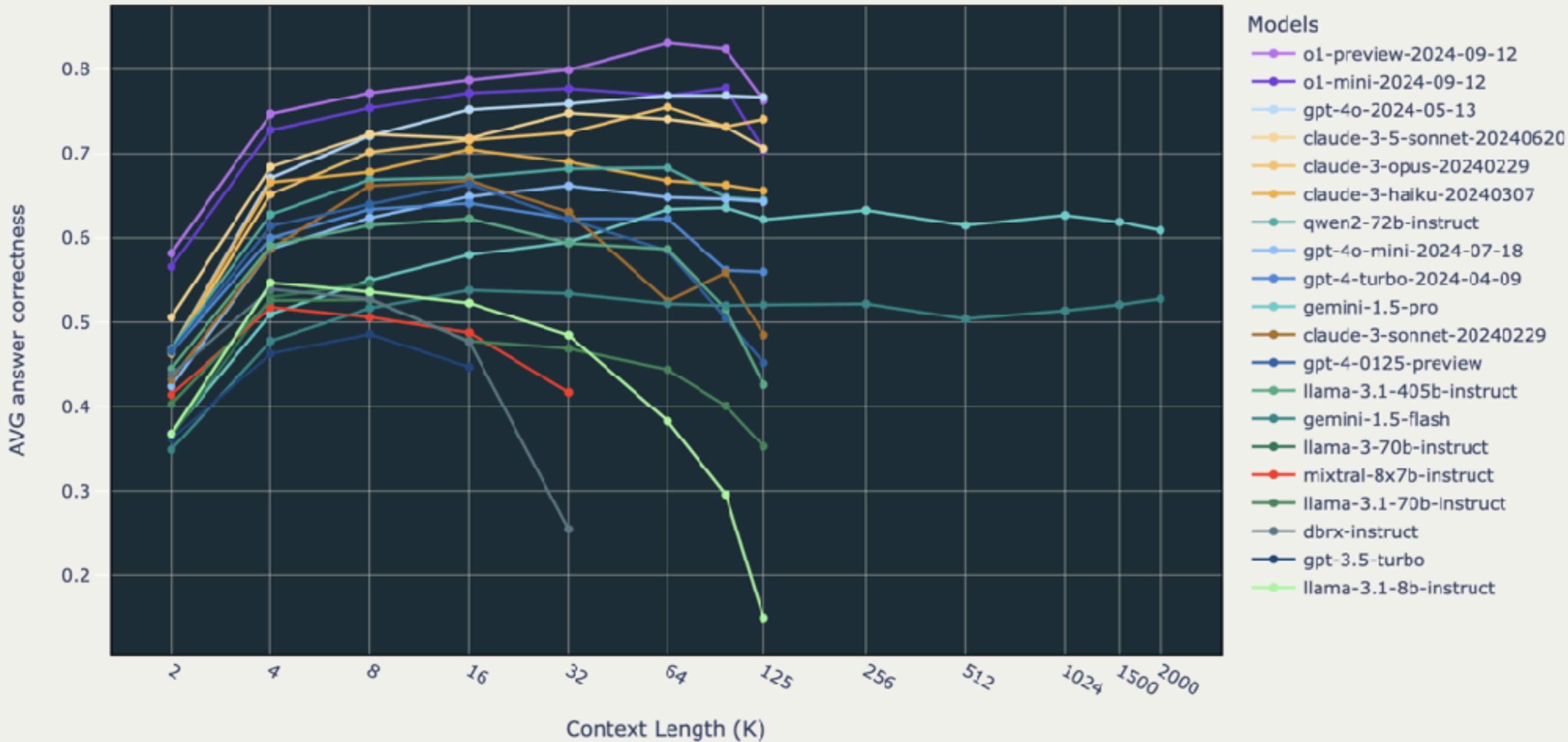
Below this are three boxes: "Input" (\$2.00), "Cached input" (\$0.50), and "Output" (\$8.00).

<https://platform.openai.com/docs/models/gpt-4.1>



# Long context LLM is better !!

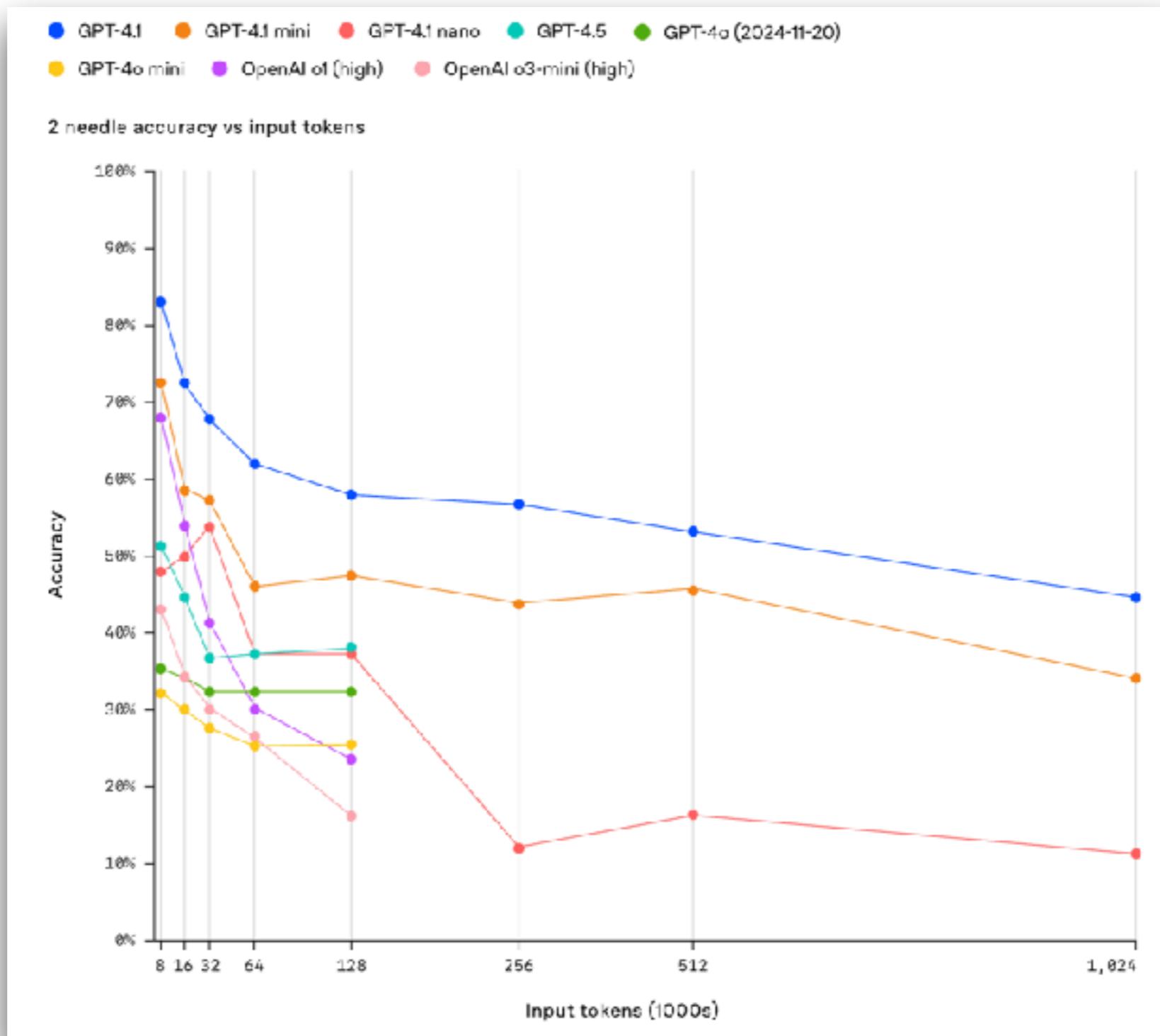
Long Context RAG Performance of LLMs



<https://arxiv.org/abs/2411.03538>



# Long context LLM is better !!



<https://openai.com/index/gpt-4-1/>



# Outdate knowledge of models



# Outdate knowledge of models

The screenshot shows the GPT-4.1 model page. At the top, there's a blue rounded rectangle with "GPT-4.1" and a "Default" dropdown. Below it, the text "Flagship GPT model for complex tasks". To the right are "Compare" and "Try in Playground" buttons. The main section has five categories: "INTELLIGENCE" (4 stars, "Higher"), "SPEED" (3 stars, "Medium"), "PRICE" (\$2 - \$8, "Input + Output"), "INPUT" (Text, Image), and "OUTPUT" (Text). Below this, a paragraph says "GPT-4.1 is our flagship model for complex tasks. It is well suited for problem solving across domains." To the right are three icons: a star with "1,047,576 context window", a right arrow with "32,768 max output tokens", and a square with "Jun 01, 2024 knowledge cutoff". A horizontal line separates this from a "Pricing" section. In "Pricing", it says "Pricing is based on the number of tokens used. For tool-specific models, like search and computer use, there's a fee per tool call. See details in the [pricing page](#)". Below are three boxes: "Text tokens" with "Input \$2.00", "Cached input \$0.50", and "Output \$8.00". A switch next to "Output" is set to "Per 1M tokens + Batch API price".

<https://platform.openai.com/docs/models/gpt-4.1>



# Knowledge cut-off date

Model name	Cut-off date
GPT 4.1, o4-mini	2024/06
Gemini 2.5	2025/01
Llama 4	2024/08
Claude Sonnet 3.7	2024/10



# Solutions ?

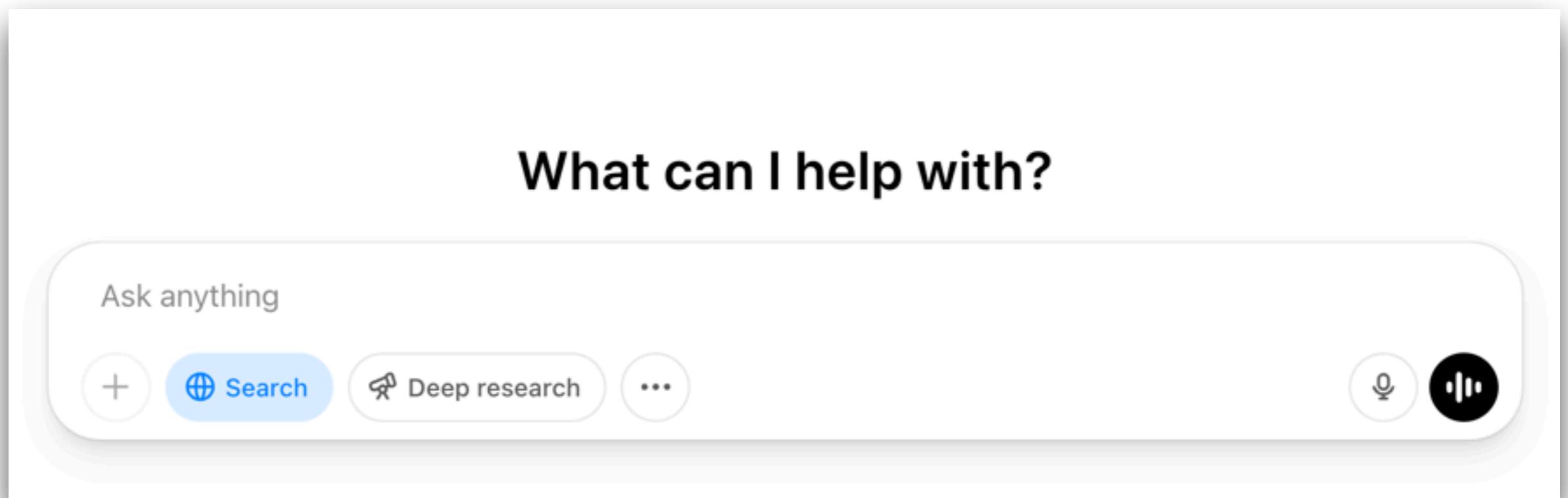


# Solution ?

- Prompt engineering
- Increase context window size
- Context caching or prompt caching
- Add search features
- Fine-tuning with high-quality domain dataset
- RAG (Retrieval-Augmented Generation)



# Search feature ?

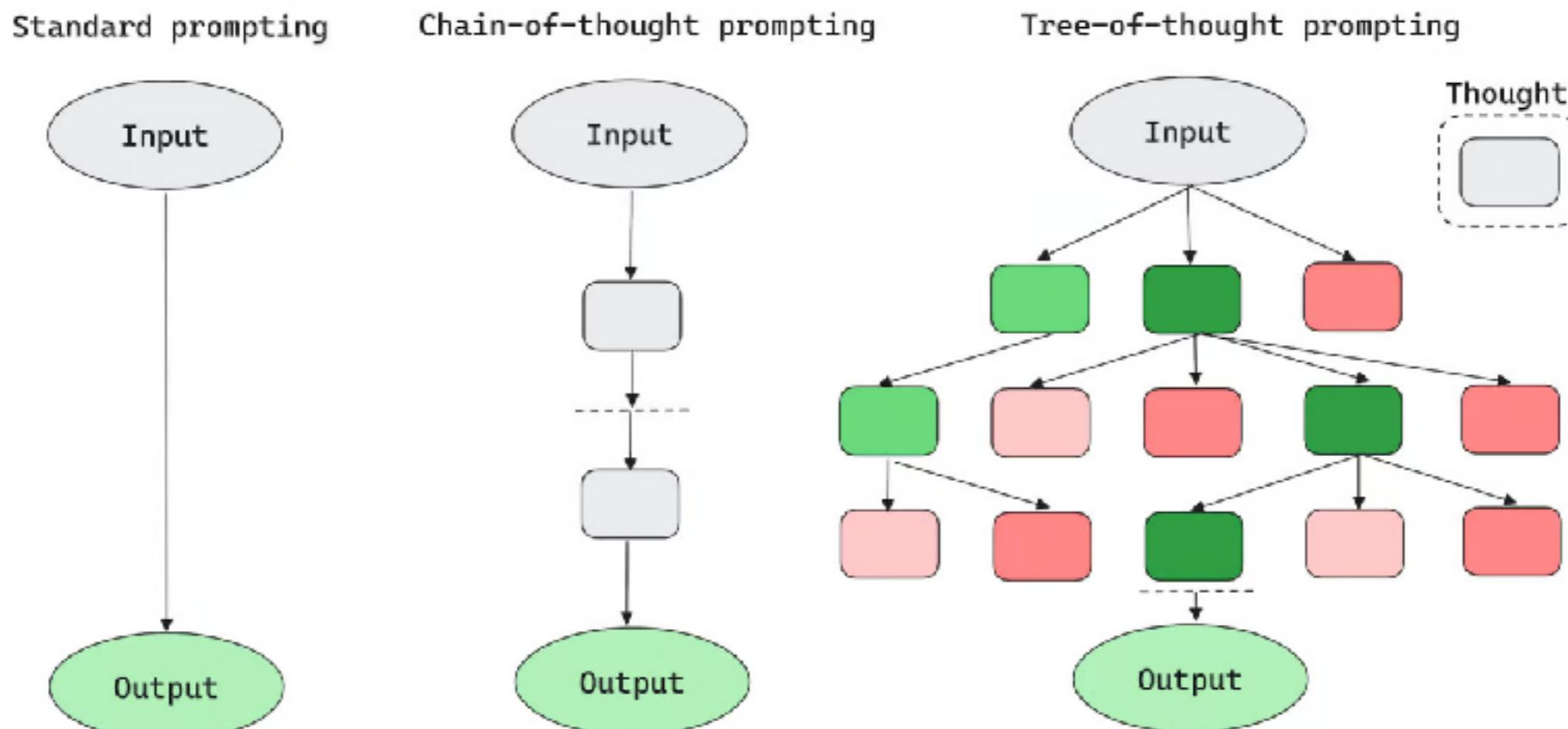


<https://chatgpt.com/>



# Prompt Engineering

A Guide to Prompting LLMs!



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



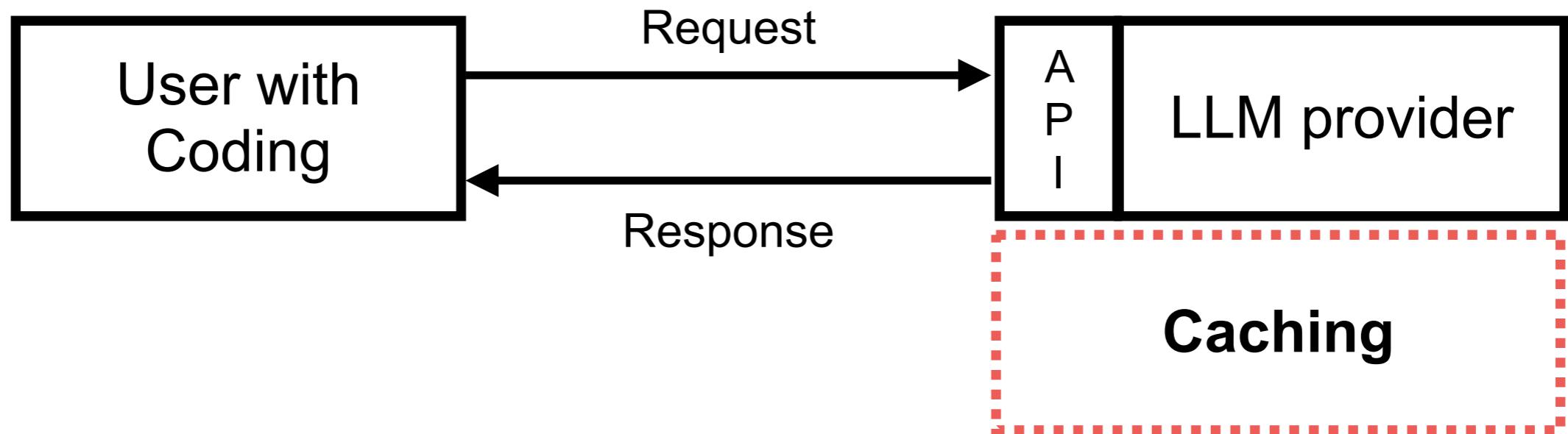
# Context or Prompt Caching

Optimize the processing of large context window

Reduce computation time and resources usages

Cost saving

**Reuse previous context**



# **Response caching ?**

## **semantic caching**



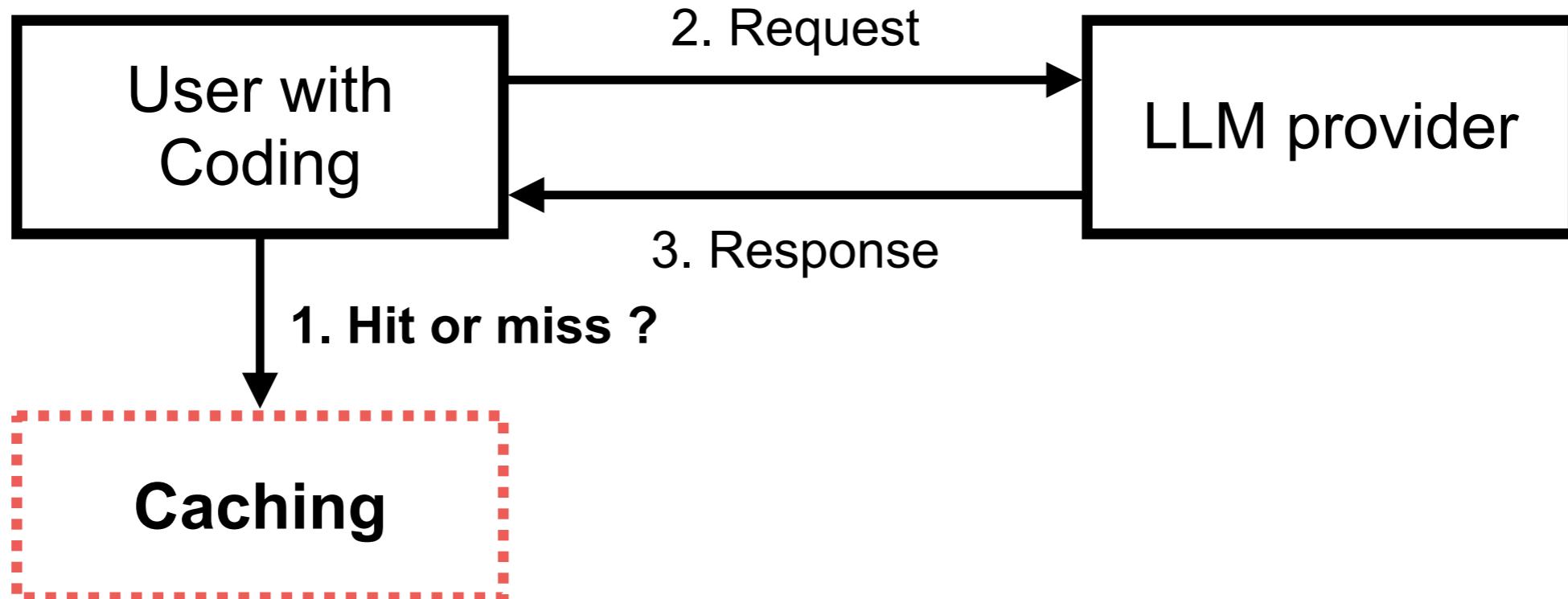
# Semantic Caching

Caching response from LLM

Reduce no. of query to LLM provider

Reduce cost

Improve scalability of system



# Function Calling ?

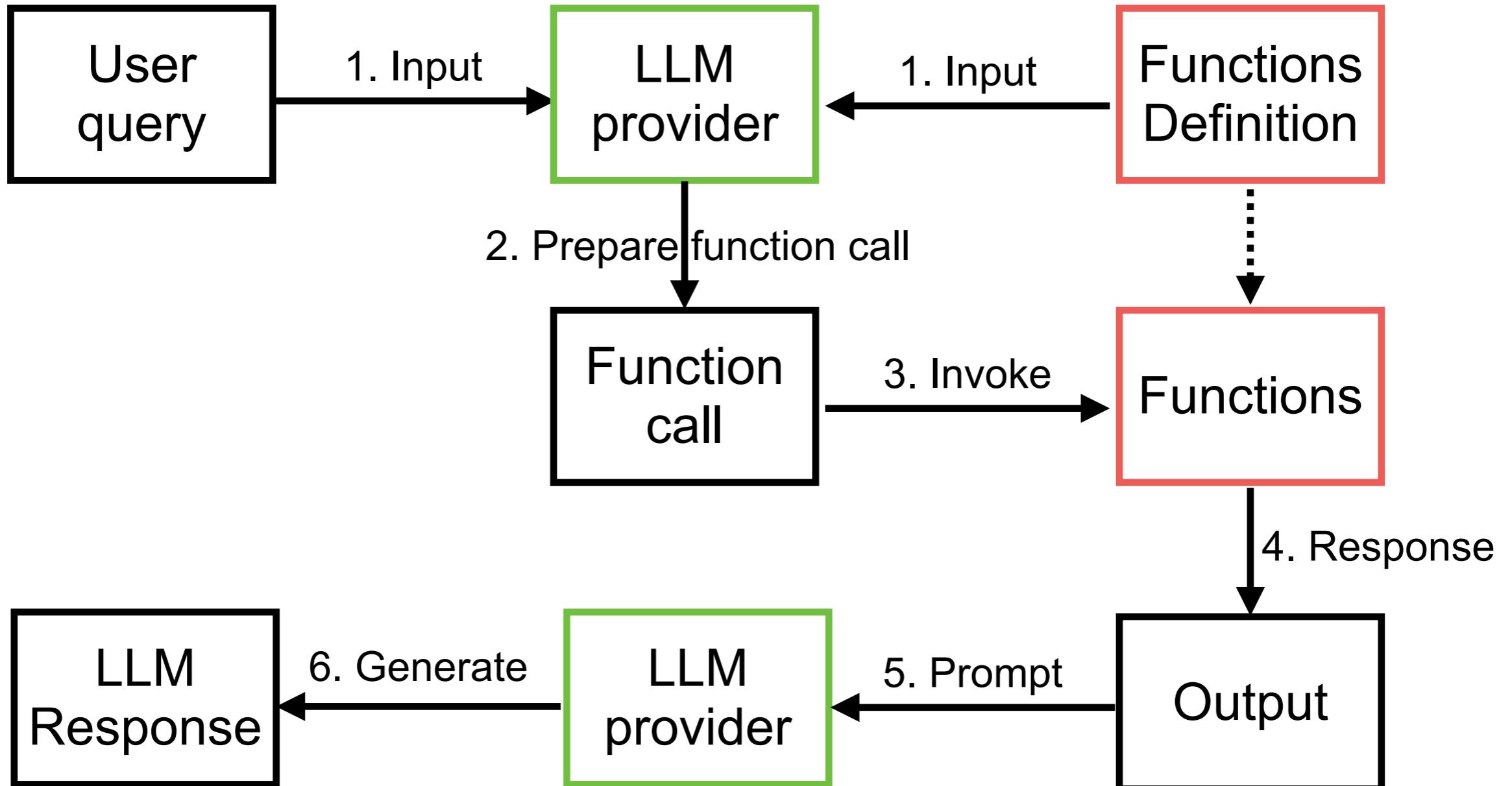


# Function Calling ?

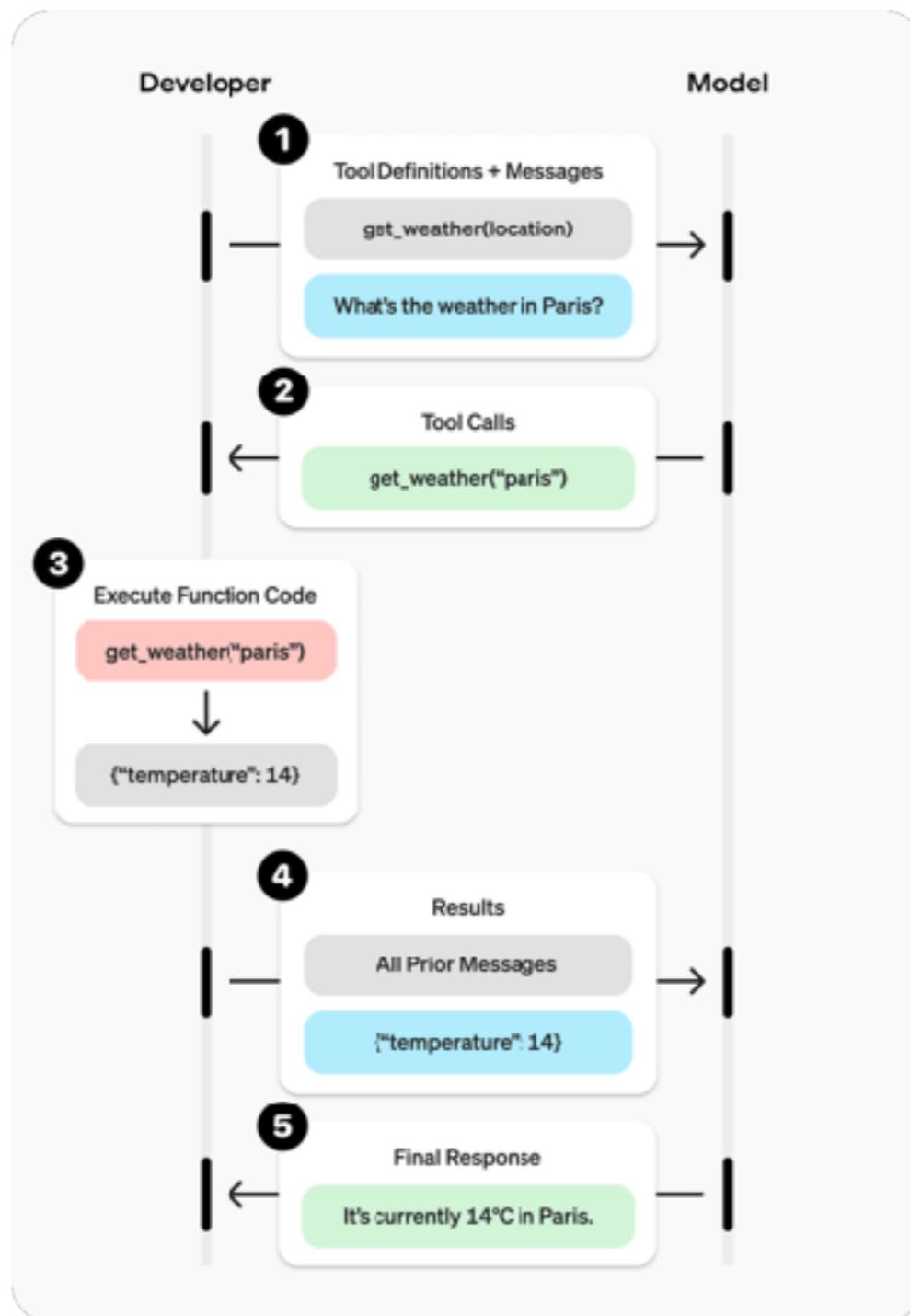
Allow LLM to recognize what tool it need based on user's input and when to invoke it.



# Function Calling



# Function calling ?



<https://platform.openai.com/docs/guides/function-calling?api-mode=responses>



# Support function calling

Provider name	Model name
OpenAI	GPT 4
Anthropic	Claude 3 (Sonnet, Haiku, Opus)
Google	Gemini



# Function calling !!

**APIs: Every tool needs its own key**

Traditional APIs require different authentication and integration for each service,  
like needing different keys for different locks

APIs

M

31

Google Sheets

Google Maps

Google Photos

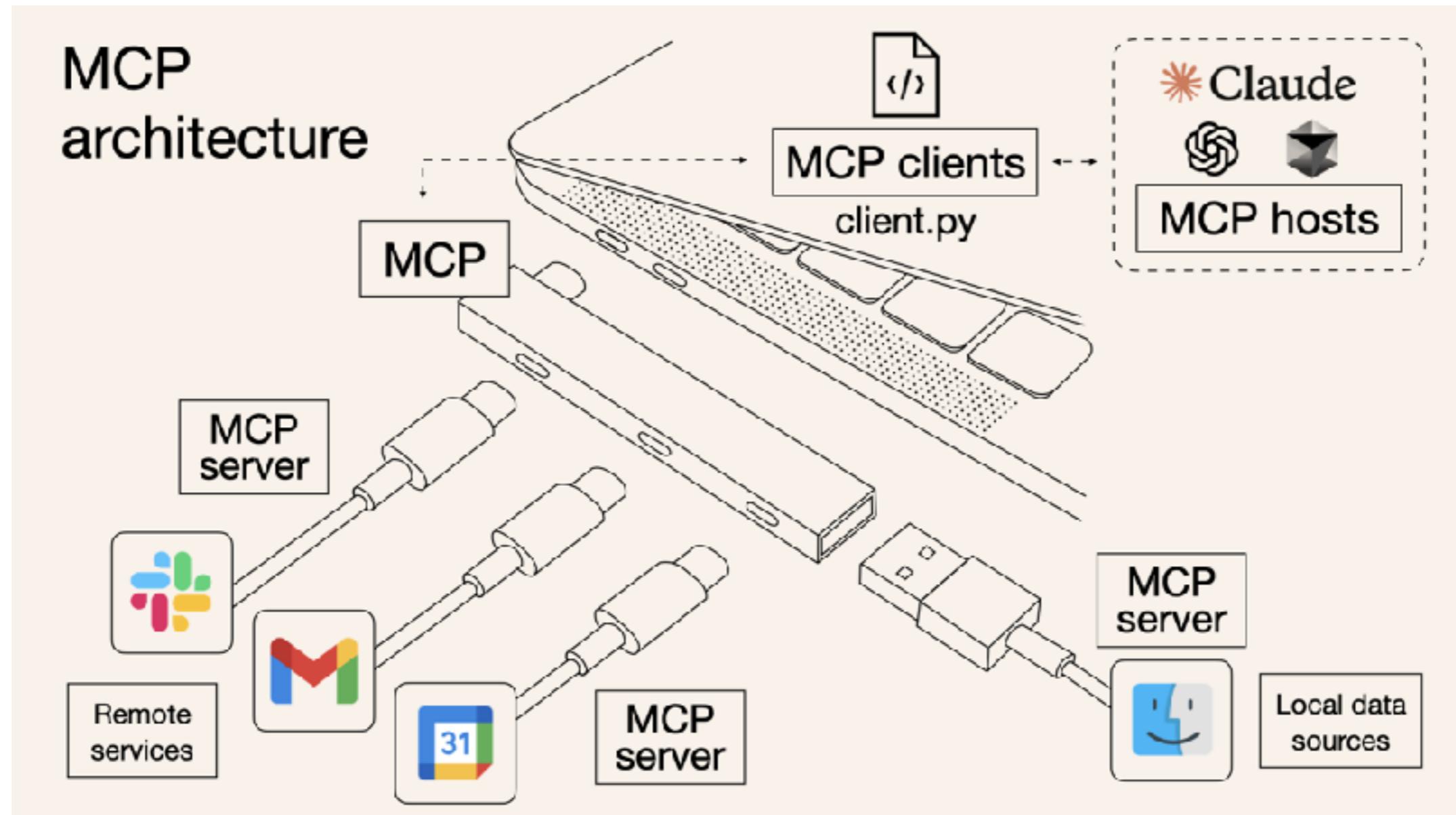
Google Drive

WhatsApp

<https://norahsakal.com/blog/mcp-vs-api-model-context-protocol-explained/>



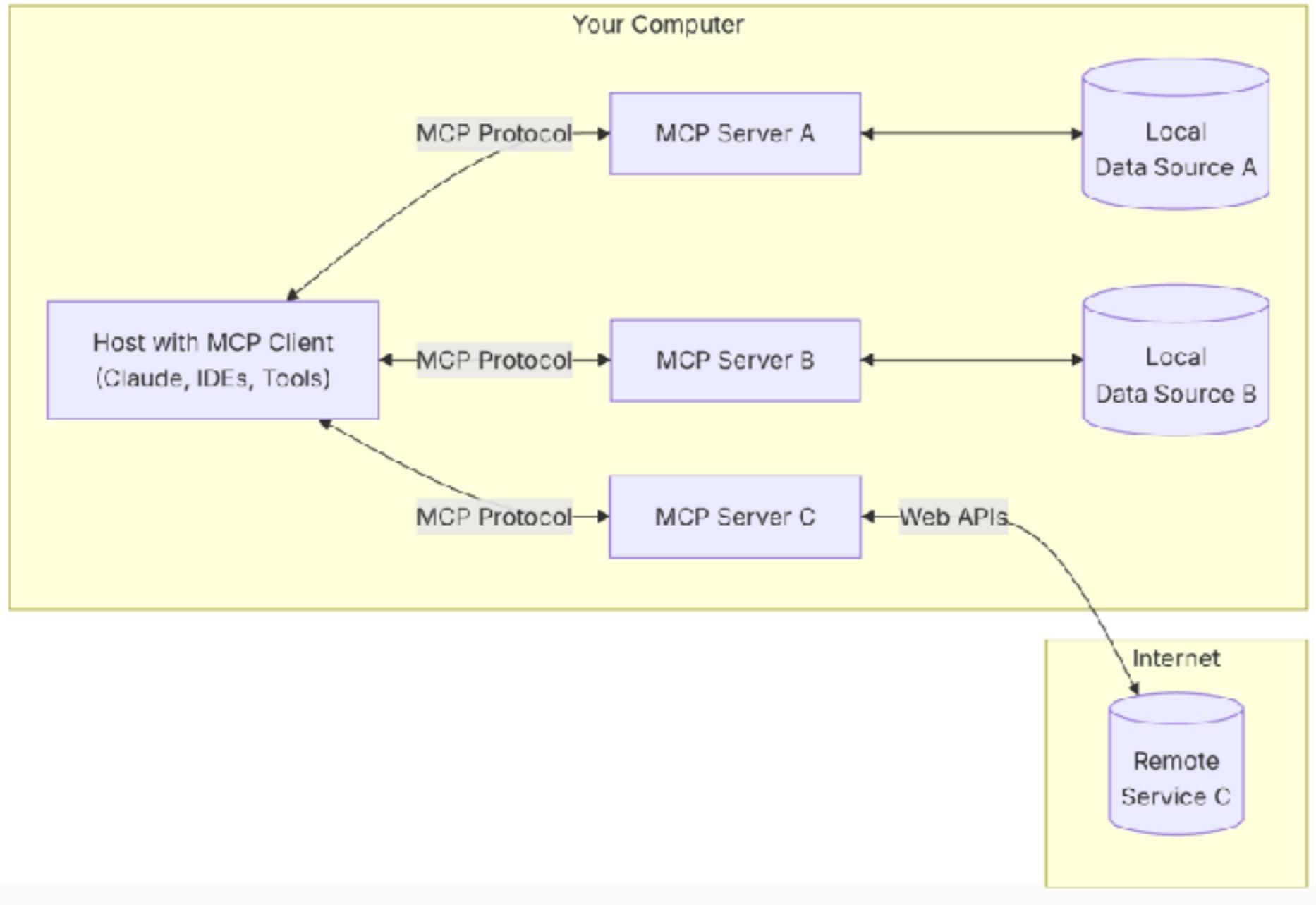
# Standardized function calling !!



<https://norahsakal.com/blog/mcp-vs-api-model-context-protocol-explained/>



# Model Context Protocol



<https://modelcontextprotocol.io/>

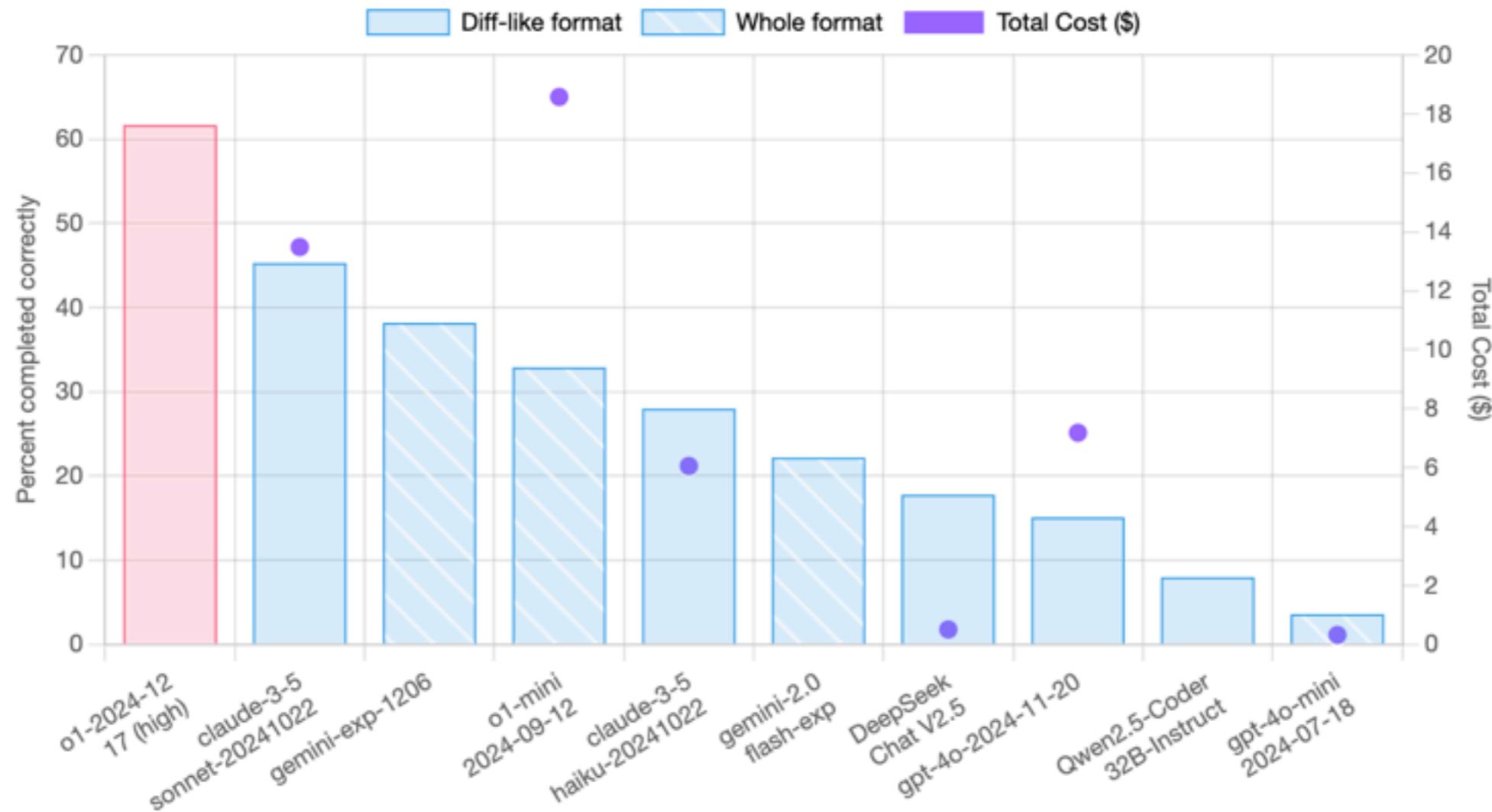


# Choosing the right AI model for your task

<https://docs.github.com/en/copilot/using-github-copilot/ai-models/choosing-the-right-ai-model-for-your-task>



# Aider LLM Leaderboards



<https://aider.chat/docs/leaderboards/>



# TRACKING AI

Monitoring Artificial Intelligence

By: Maxim Lott



[Go to Political Compass](#)

[About](#)

[AI Model Info](#)

[See every AI answer](#)

[Searchable Database](#)

[Explore Tests](#)

This site quizzes 20 Verbal & 6 Vision AIs every week | Last Updated: 06:03PM EDT on April 17, 2025

[About Offline Test](#)  [About Mensa Norway](#)

## IQ Test Results

Score reflects average of last 7 tests given

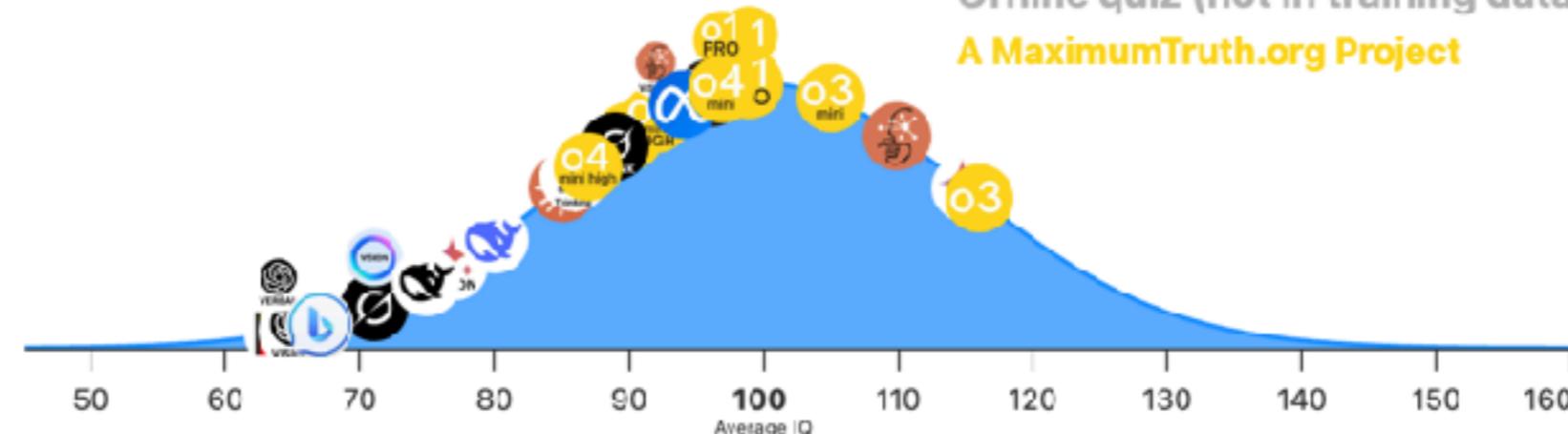
[Reset](#)

[Show Offline Test](#)

[Show Mensa Norway](#)



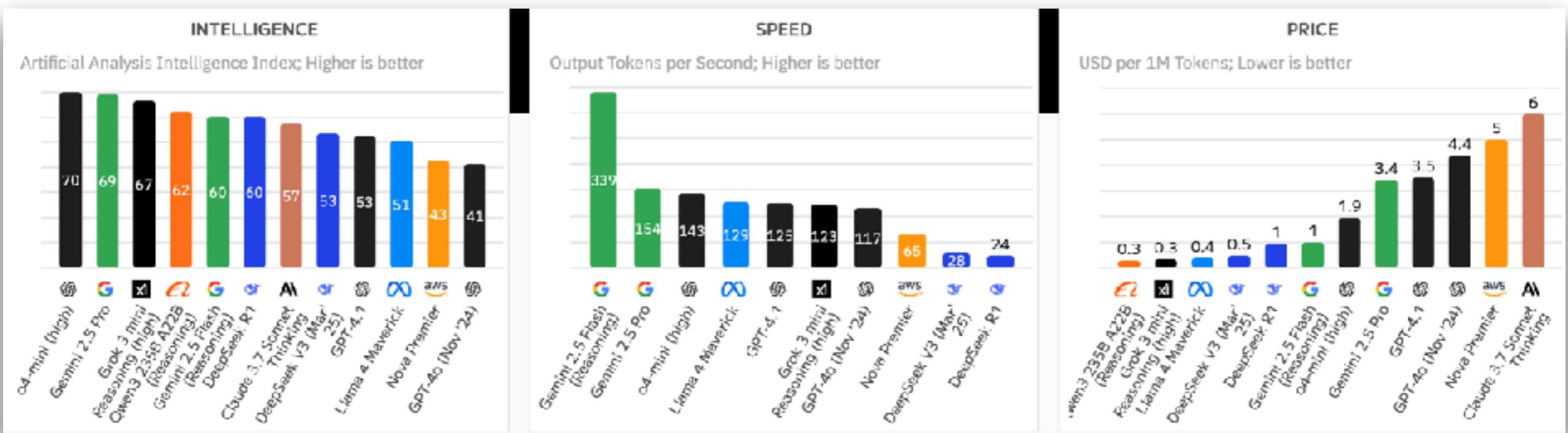
**TrackingAI.org**  
Offline quiz (not in training data)  
A MaximumTruth.org Project



<https://trackingai.org/>



# Artificial Analysis



<https://artificialanalysis.ai/>



# **RAG**

# **(Retrieval Augmented Generation)**



# 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

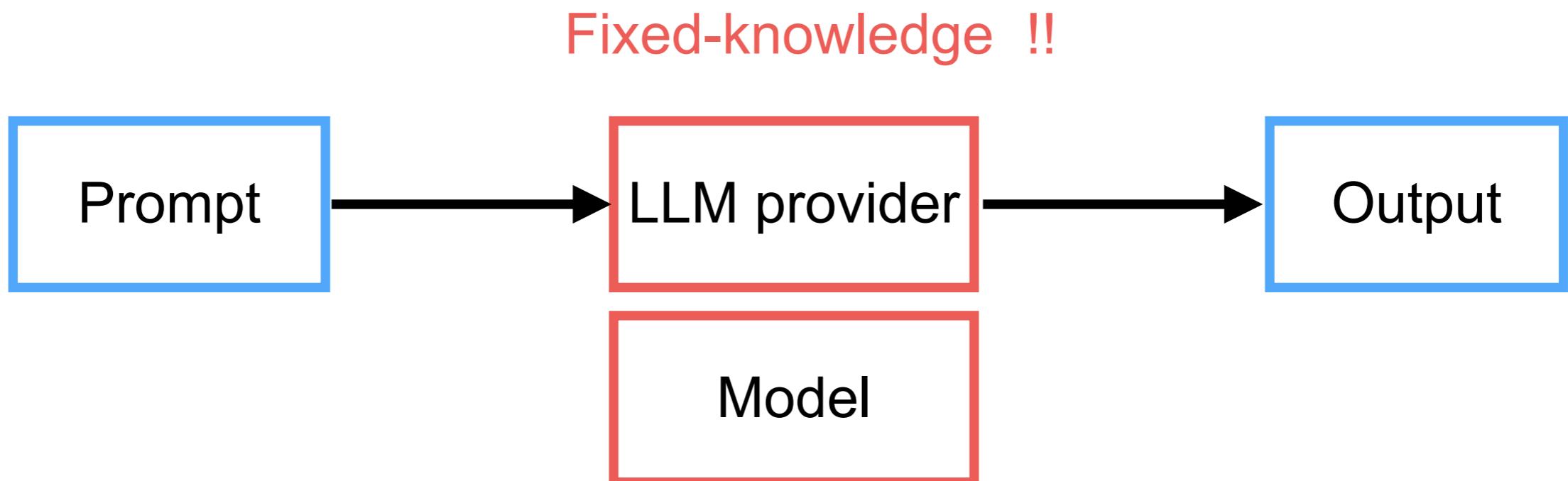
Uses the retrieved information to generate a response

## Enhancements

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

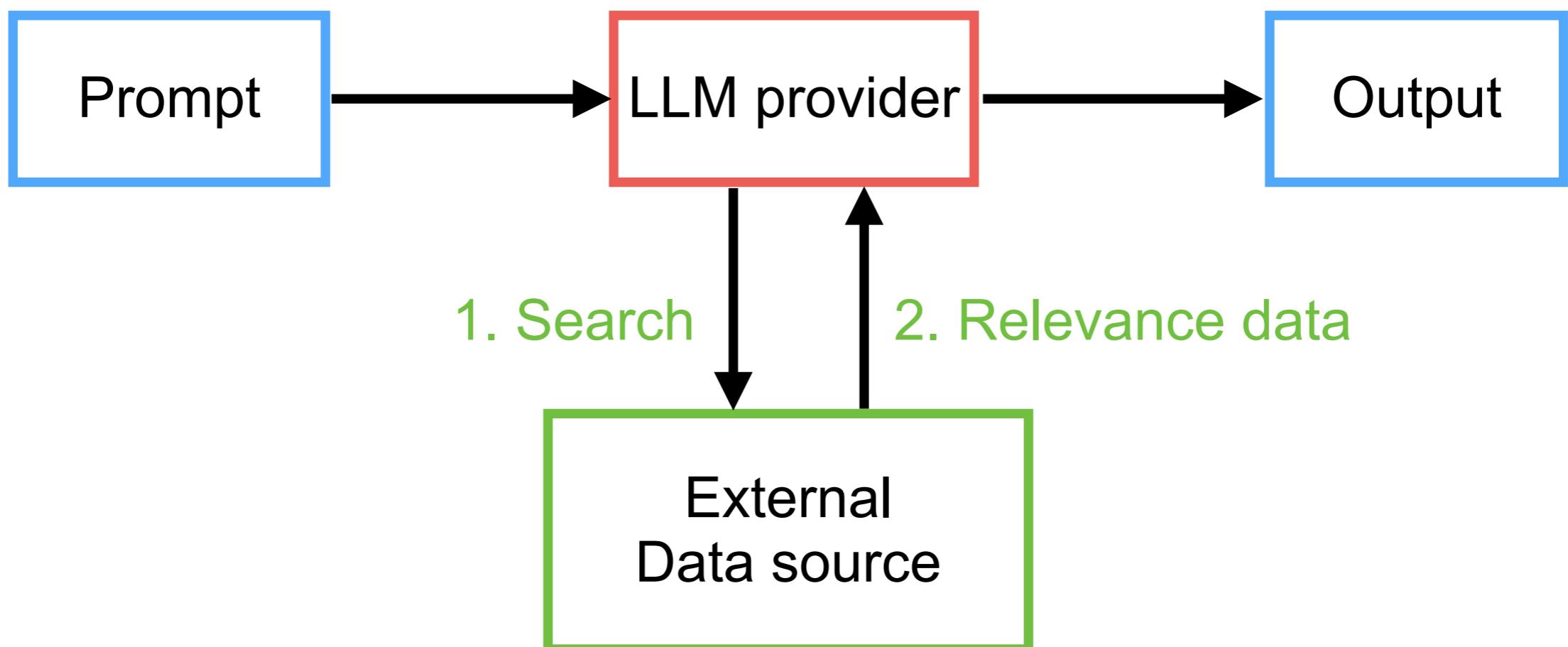


# Limitation of LLM



# RAG ?

Fixed-knowledge !!



# How to search/retrieve data ?

Data source

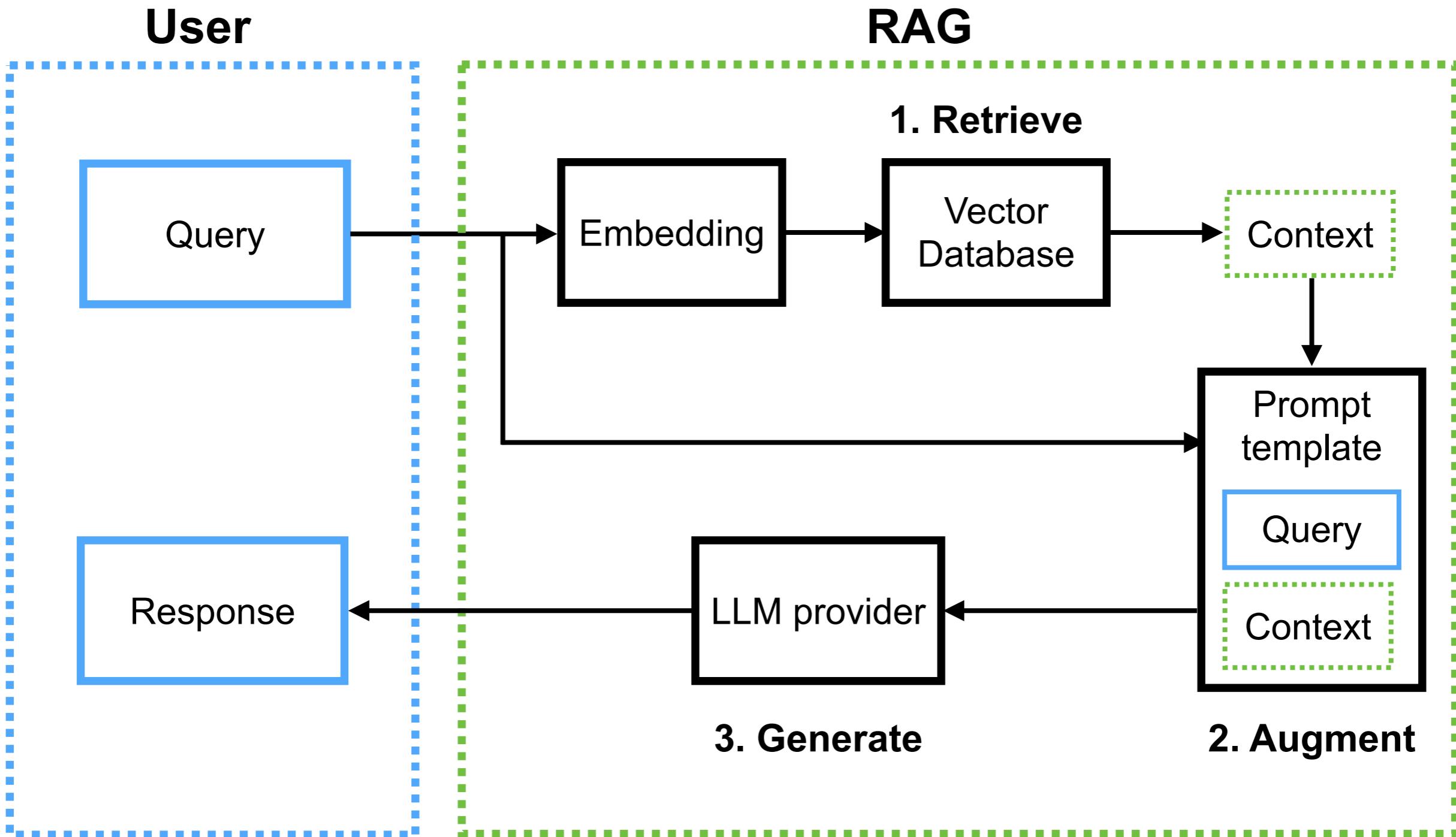
Data preparation

Search

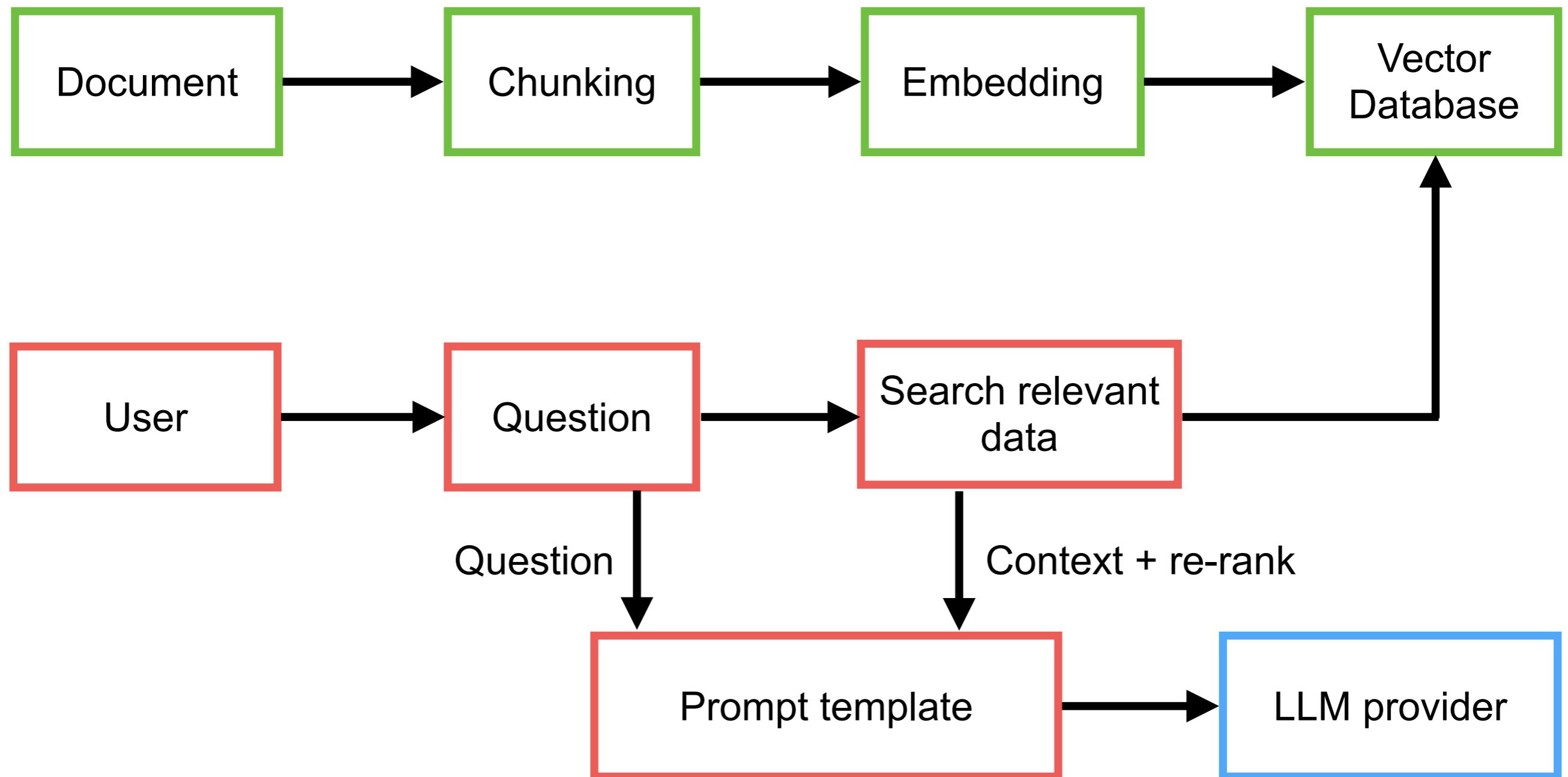
Search result



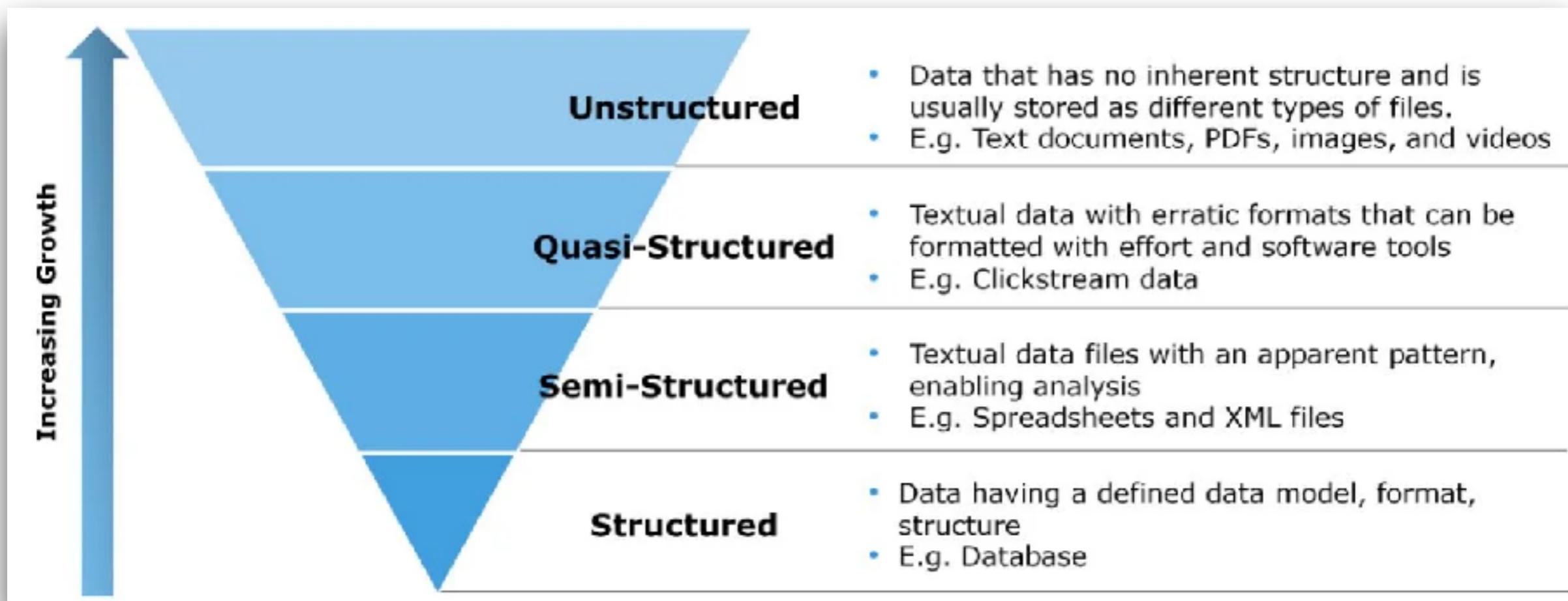
# Basic RAG Architecture



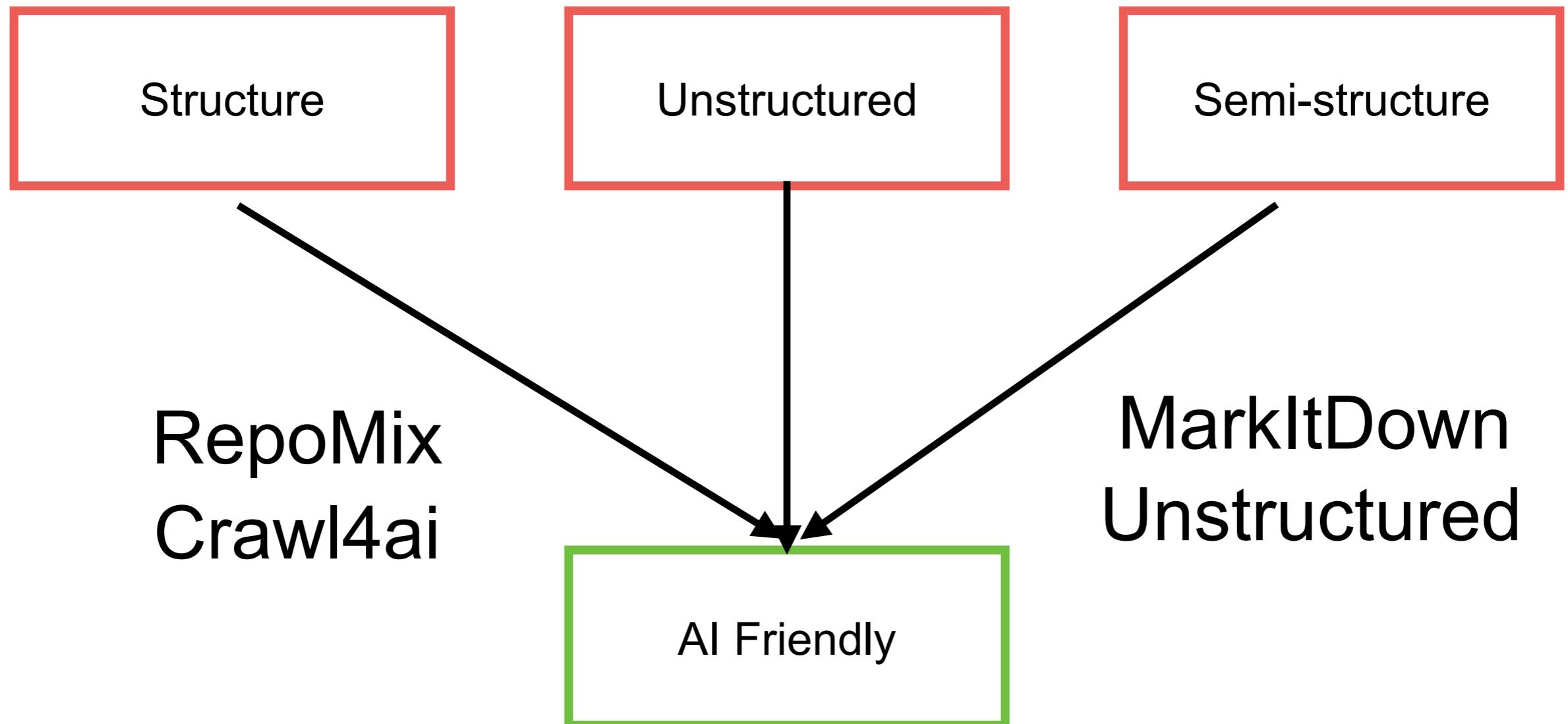
# RAG process



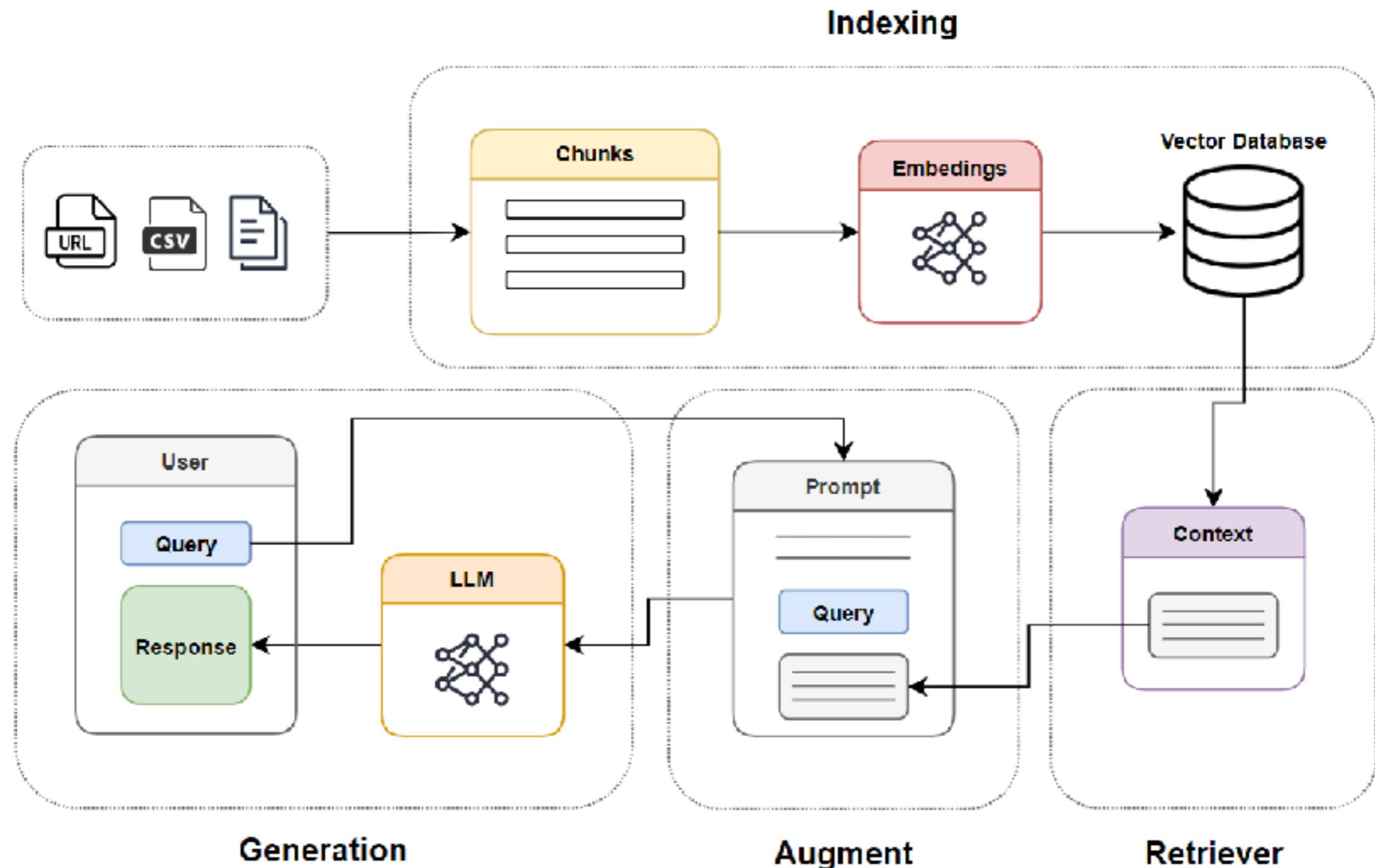
# Structures of Data ?



# Friendly Data for LLM/AI



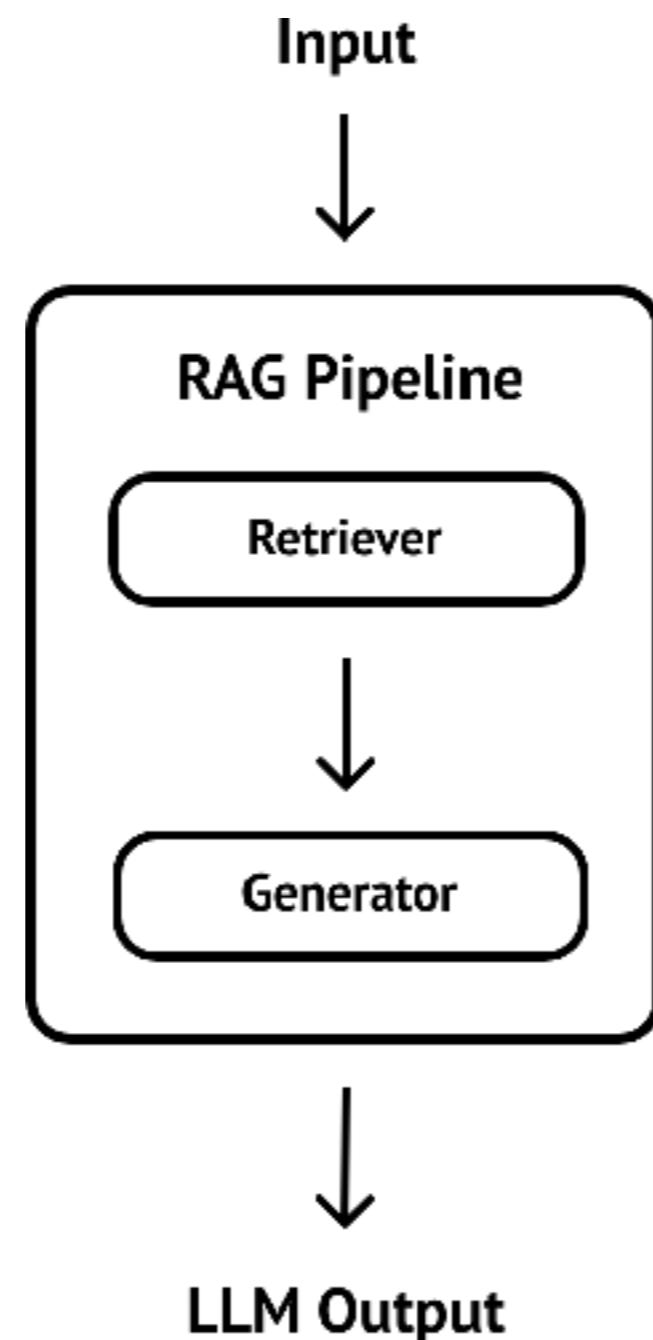
# RAG Cookbook (techniques)



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



# RAG Evaluation !!



#### Retriever Metrics:

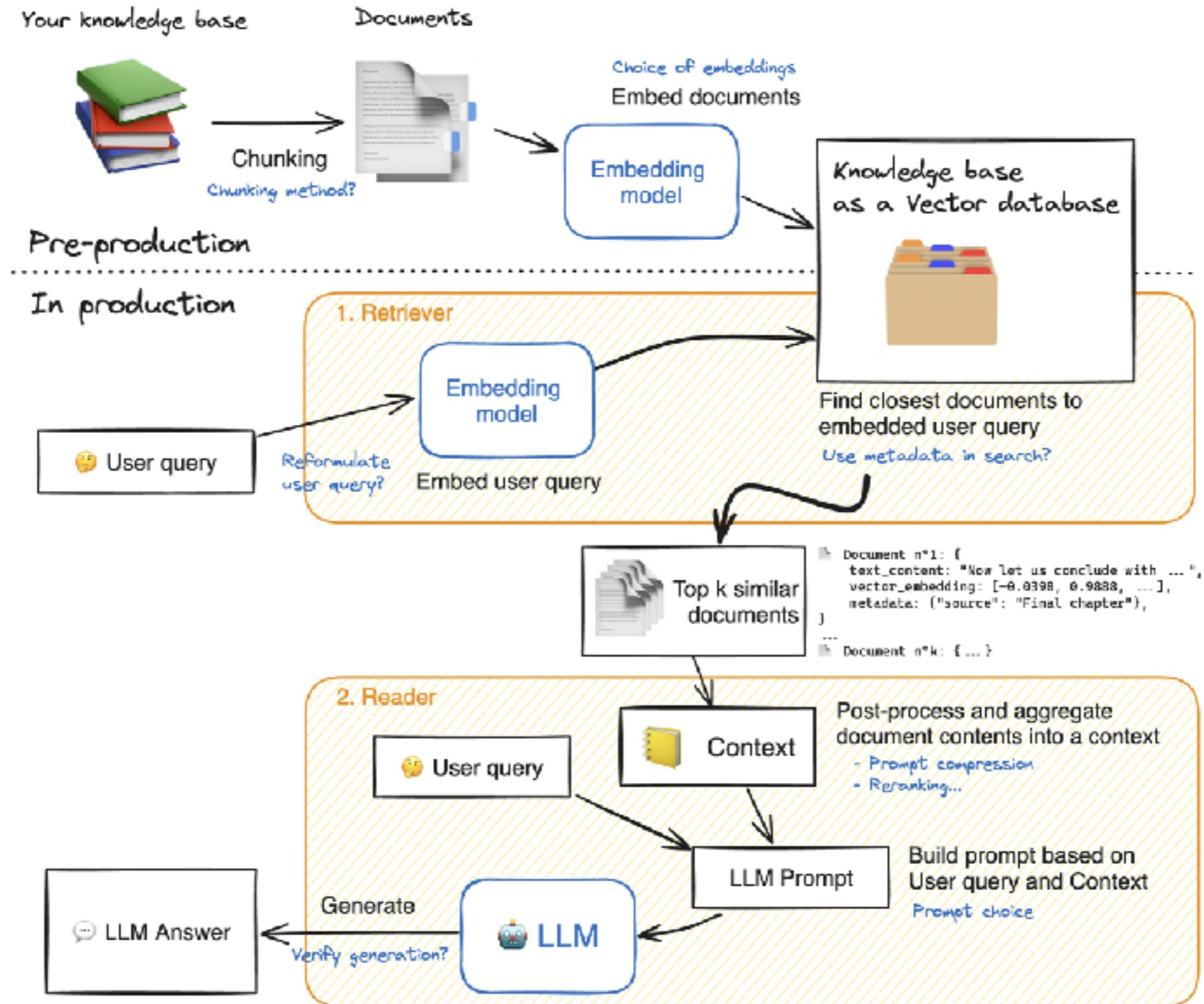
- Contextual Recall
- Contextual Precision
- Contextual Relevancy

#### Generator Metrics:

- Answer Relevancy
- Faithfulness

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

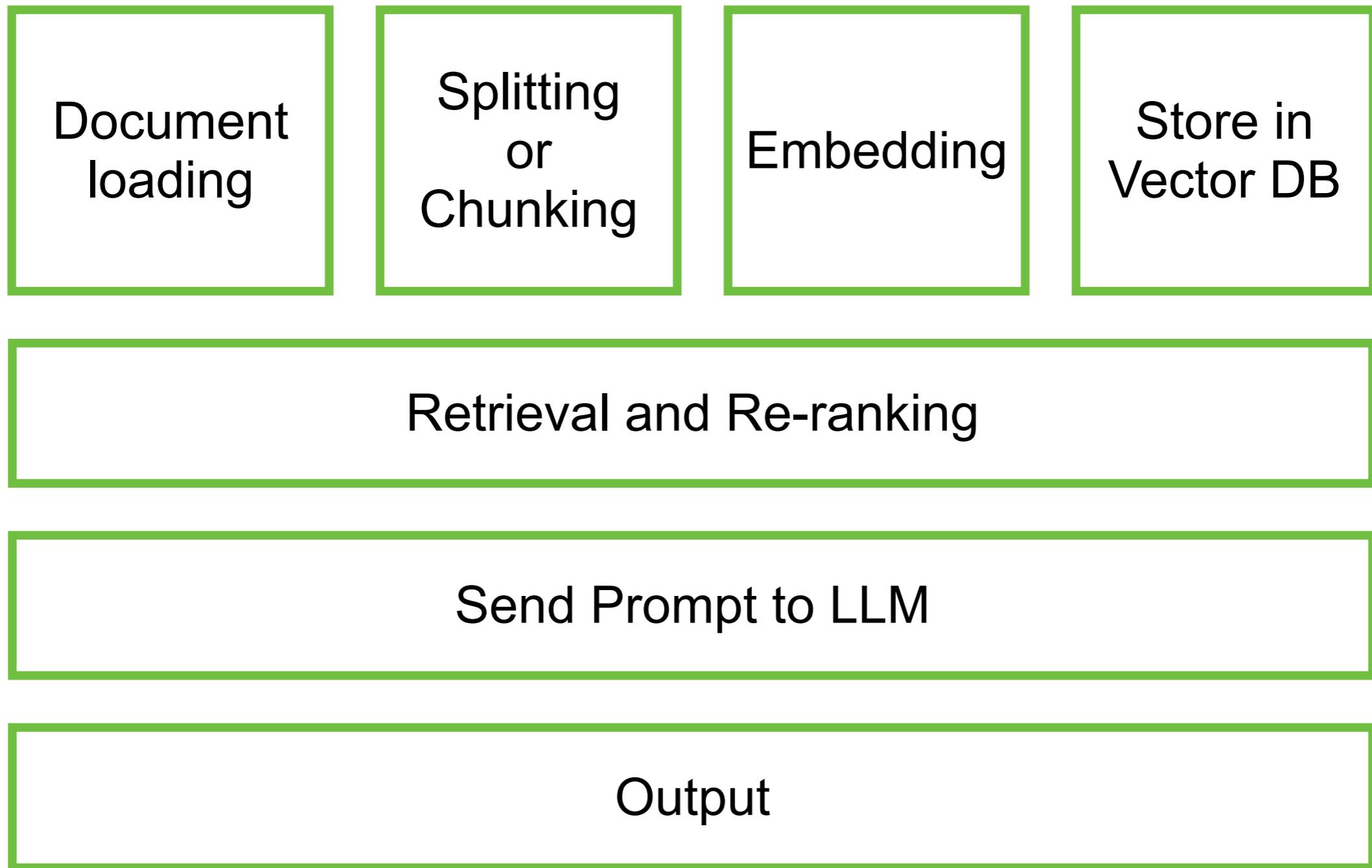




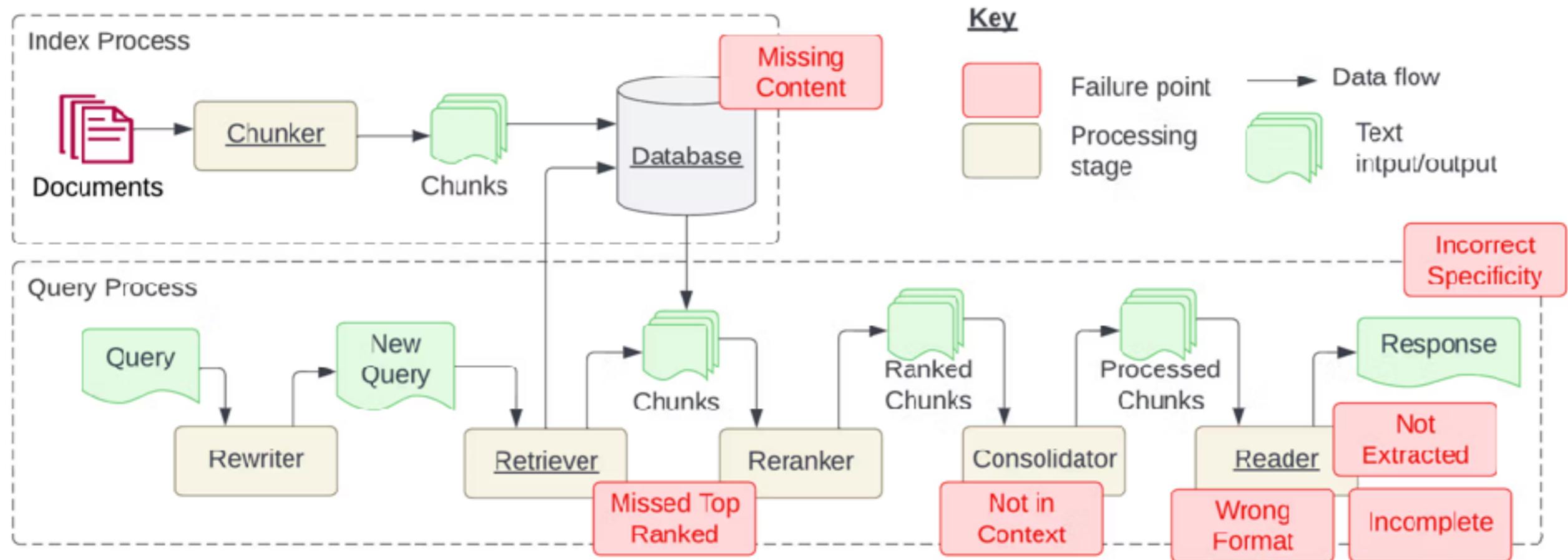
[https://huggingface.co/learn/cookbook/en/rag\\_evaluation](https://huggingface.co/learn/cookbook/en/rag_evaluation)



# 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 ?



# 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 ?



# Pre-Retrieval optimization ?

Preprocessing and cleansing data

Chunking strategies

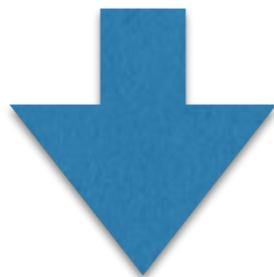
Add metadata

Embedding model selection

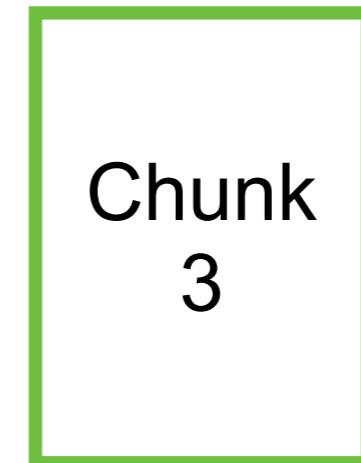
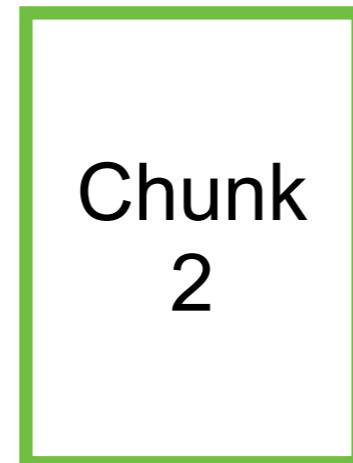
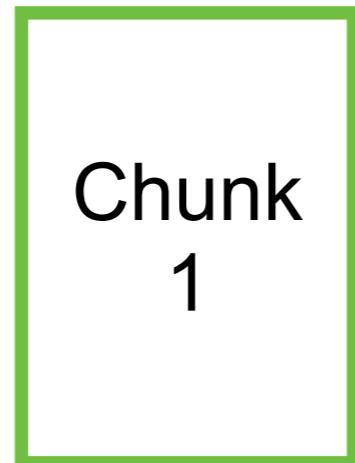


# Chunking strategies

Size of Document  
**> context window size**



Chunking



# 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 ?



# Retrieval optimization ?

Re-ranking

Hybrid search

Query  
transformation

Multi-vector  
embedding

Contextual  
retrieval

Vector database  
Selection



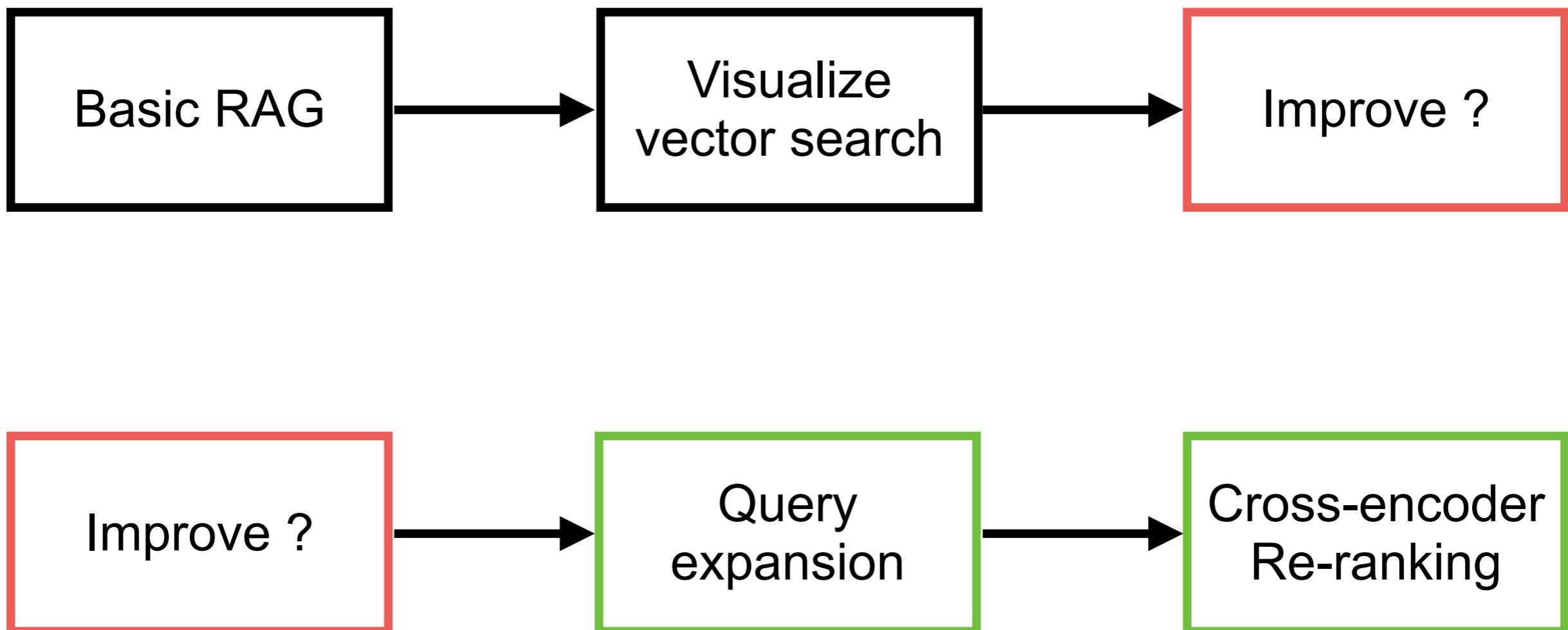


# RAG Workshop

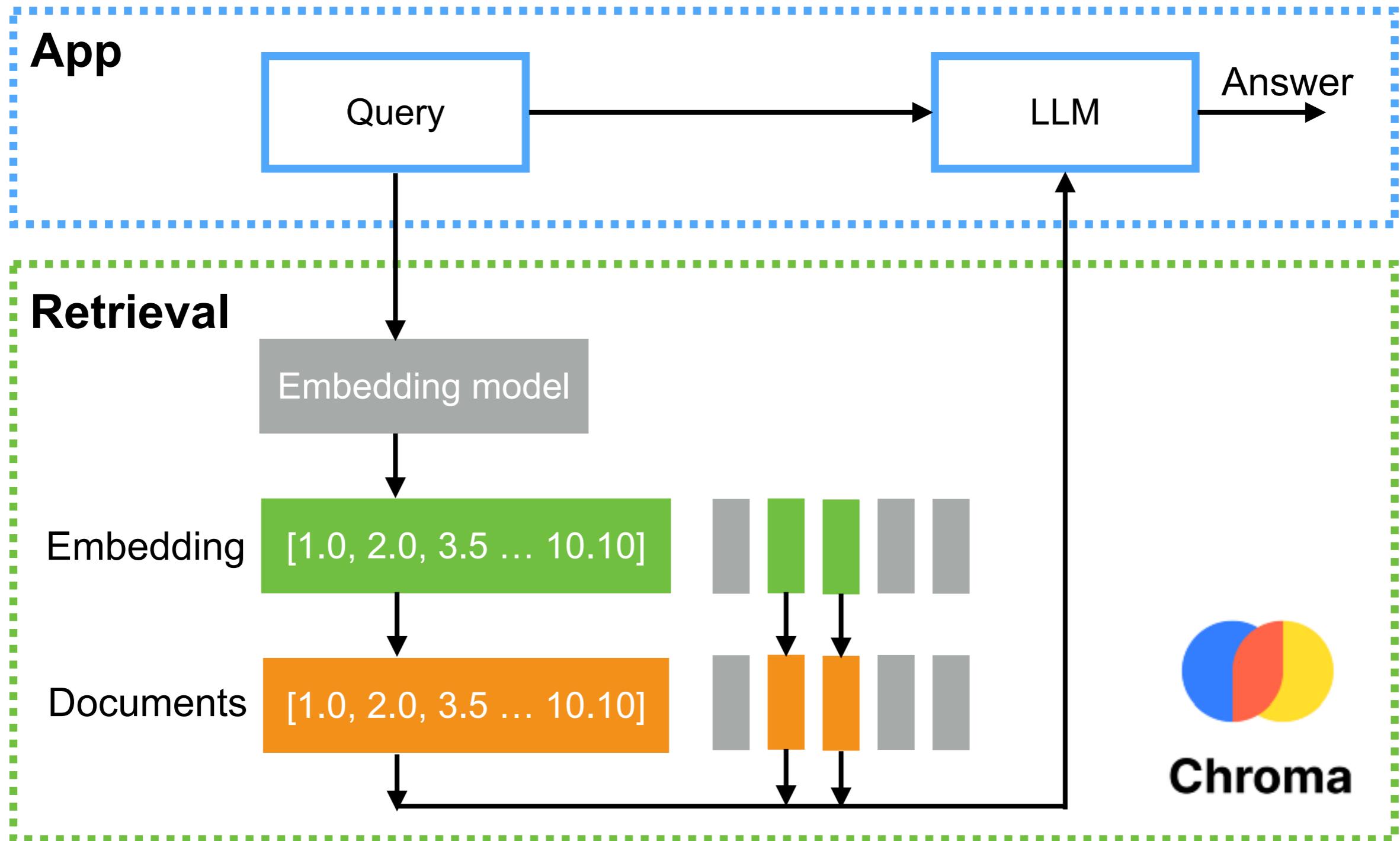
<https://github.com/up1/workshop-basic-llm/tree/main/workshop/basic-rag>



# RAG workshop



# Basic RAG



# Query Expansion

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

Ambiguity

Vocabulary  
mismatch

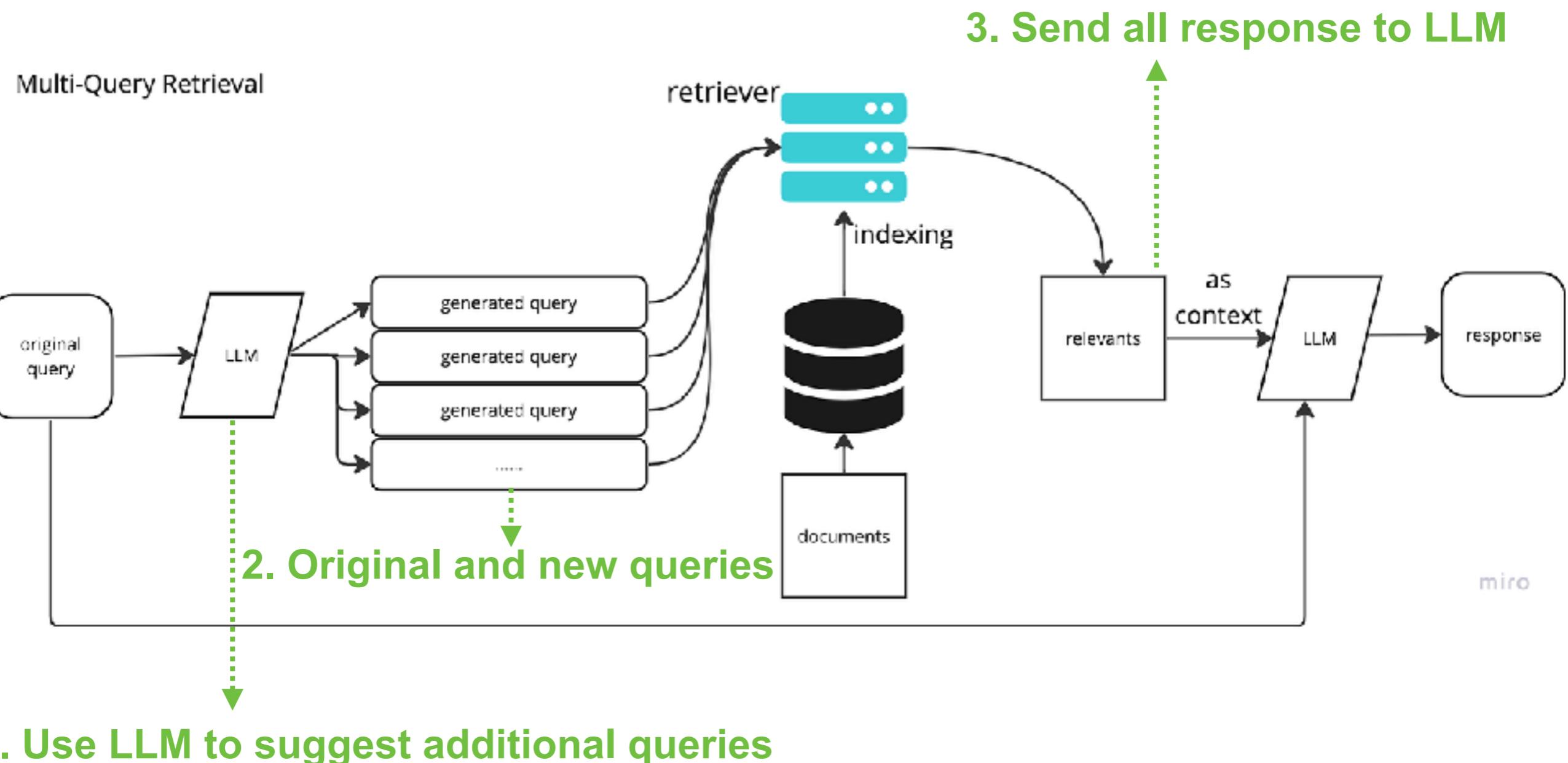
Lack of context

“Add synonyms, related context and contextual terms”

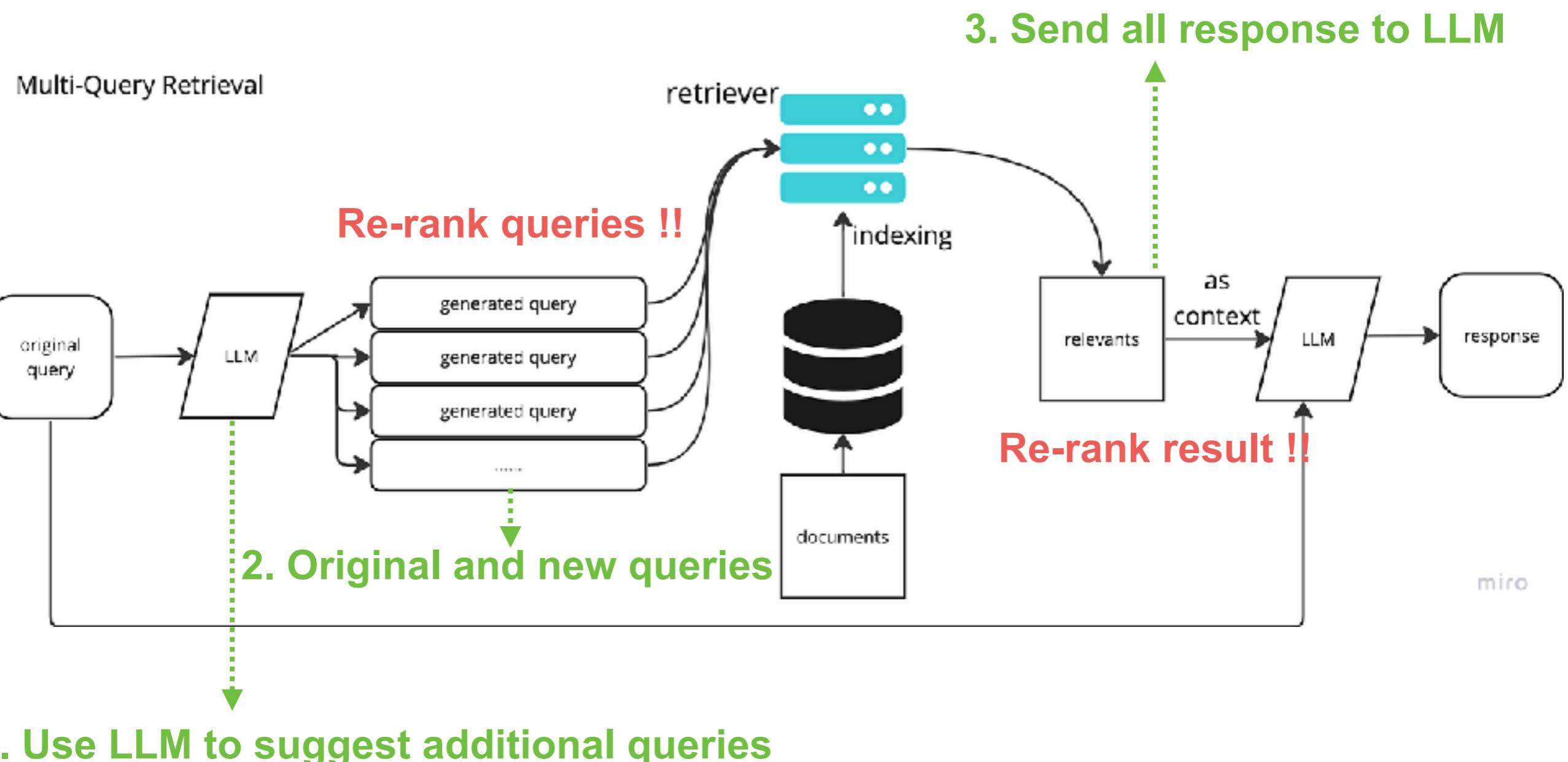
<https://arxiv.org/abs/2305.03653>



# Query Expansion

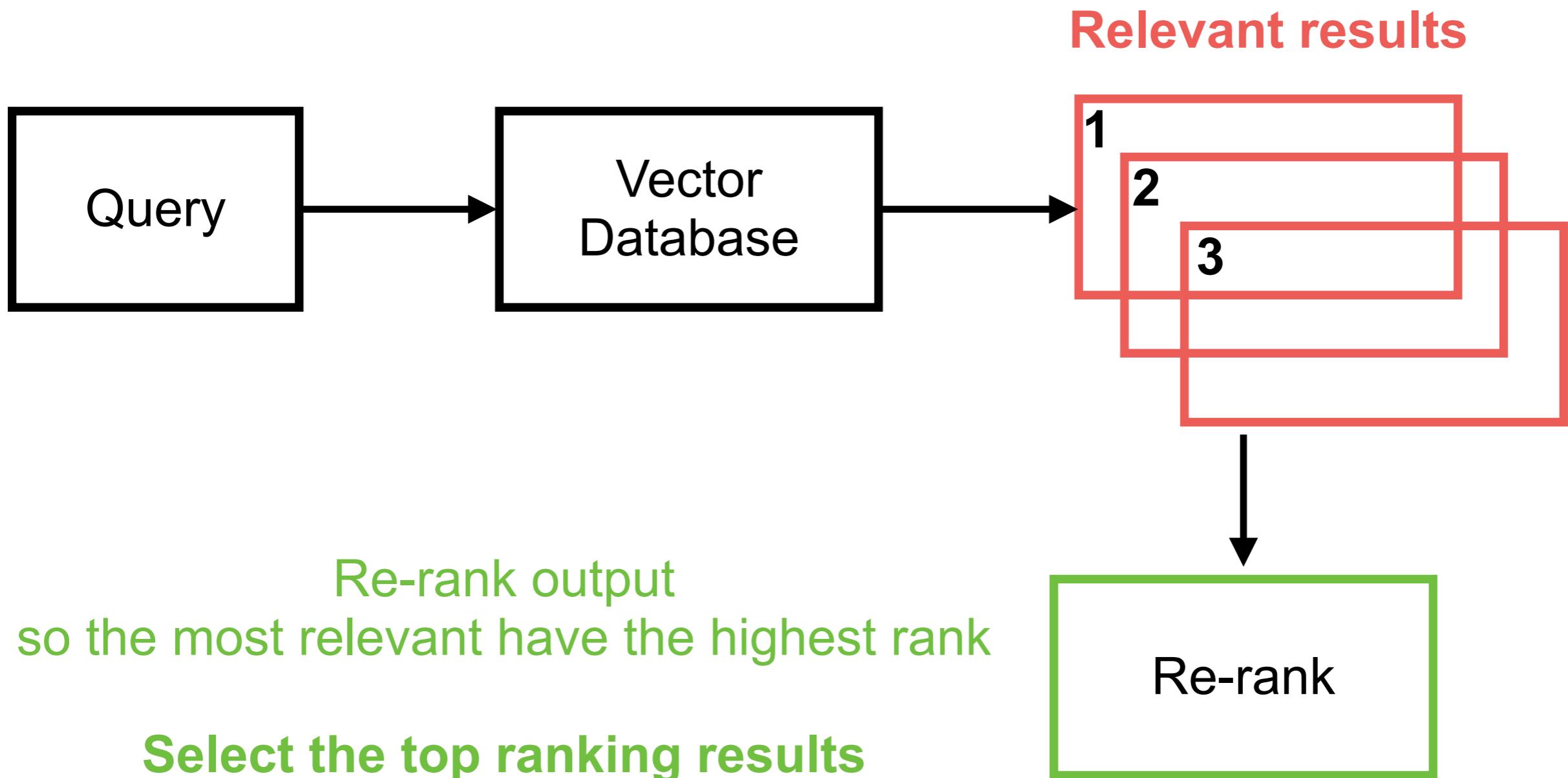


# Re-rank !!

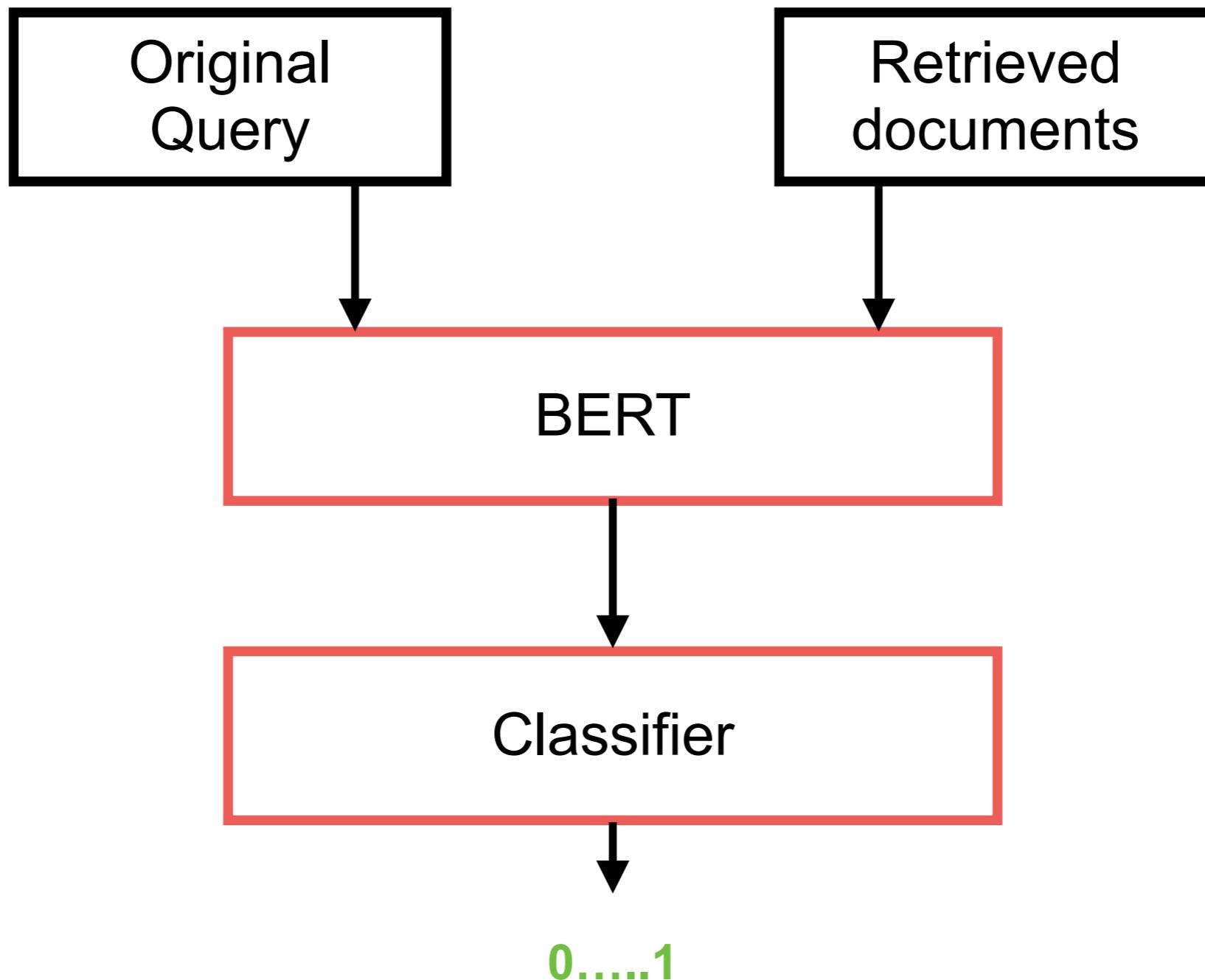


# Cross-encoder Re-ranking

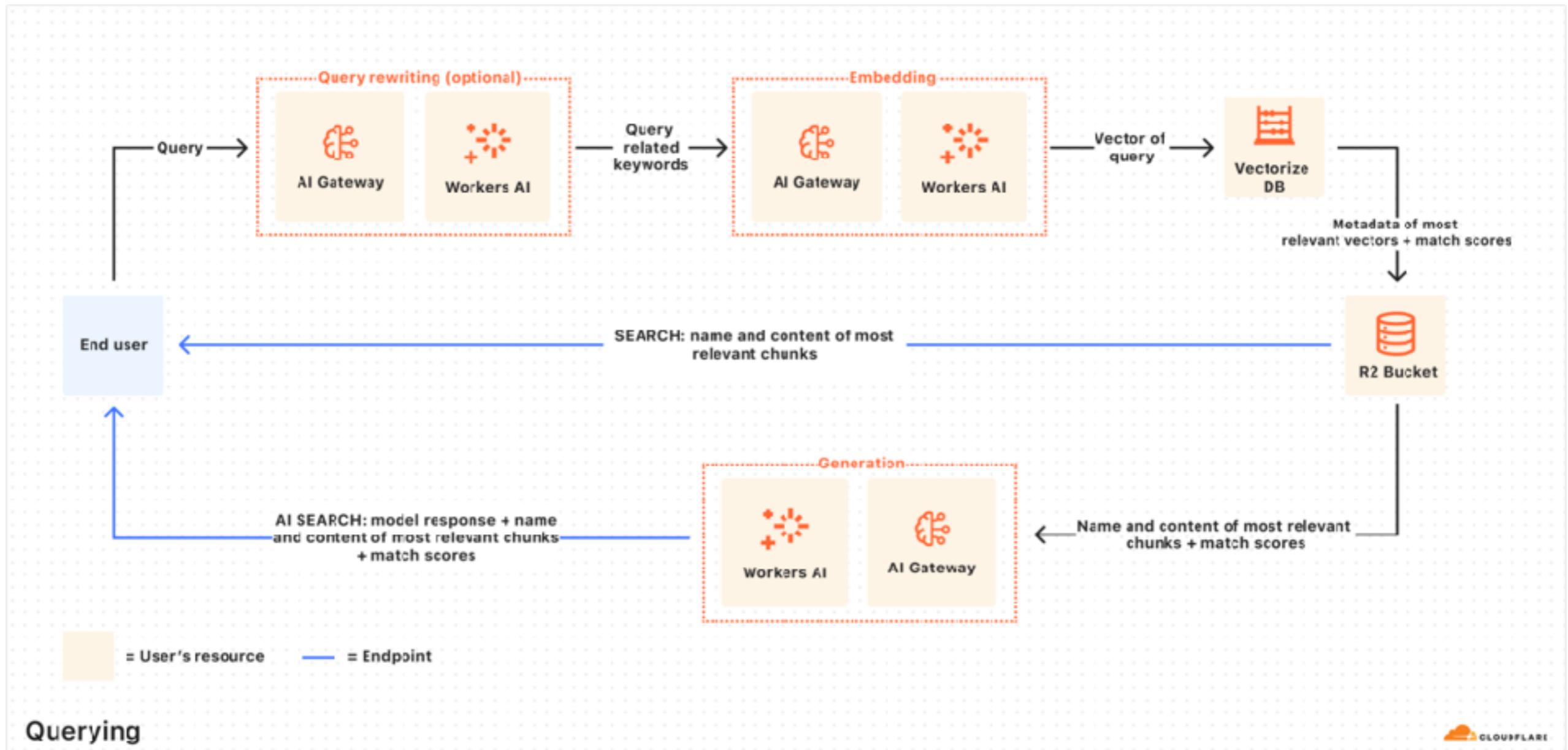
How to ordering relevant results !!



# Cross-Encoder



# Cloudflare AutoRAG



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



# Guardrails



<https://github.com/guardrails-ai/guardrails>



# Guardrails

Help to build reliable AI applications

Guard for  
input and output

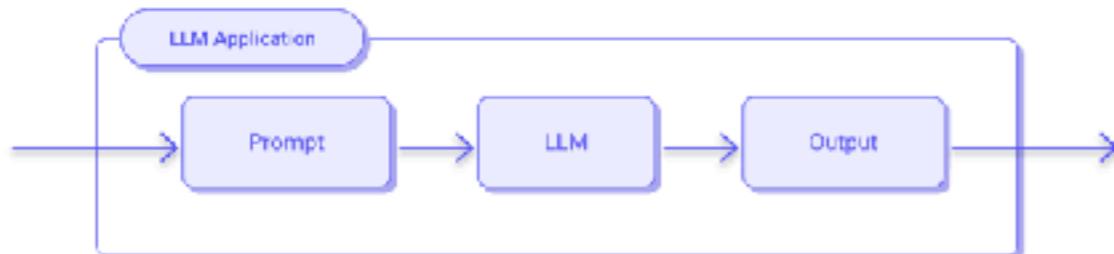
Help to generate  
Structured output

<https://github.com/guardrails-ai/guardrails>

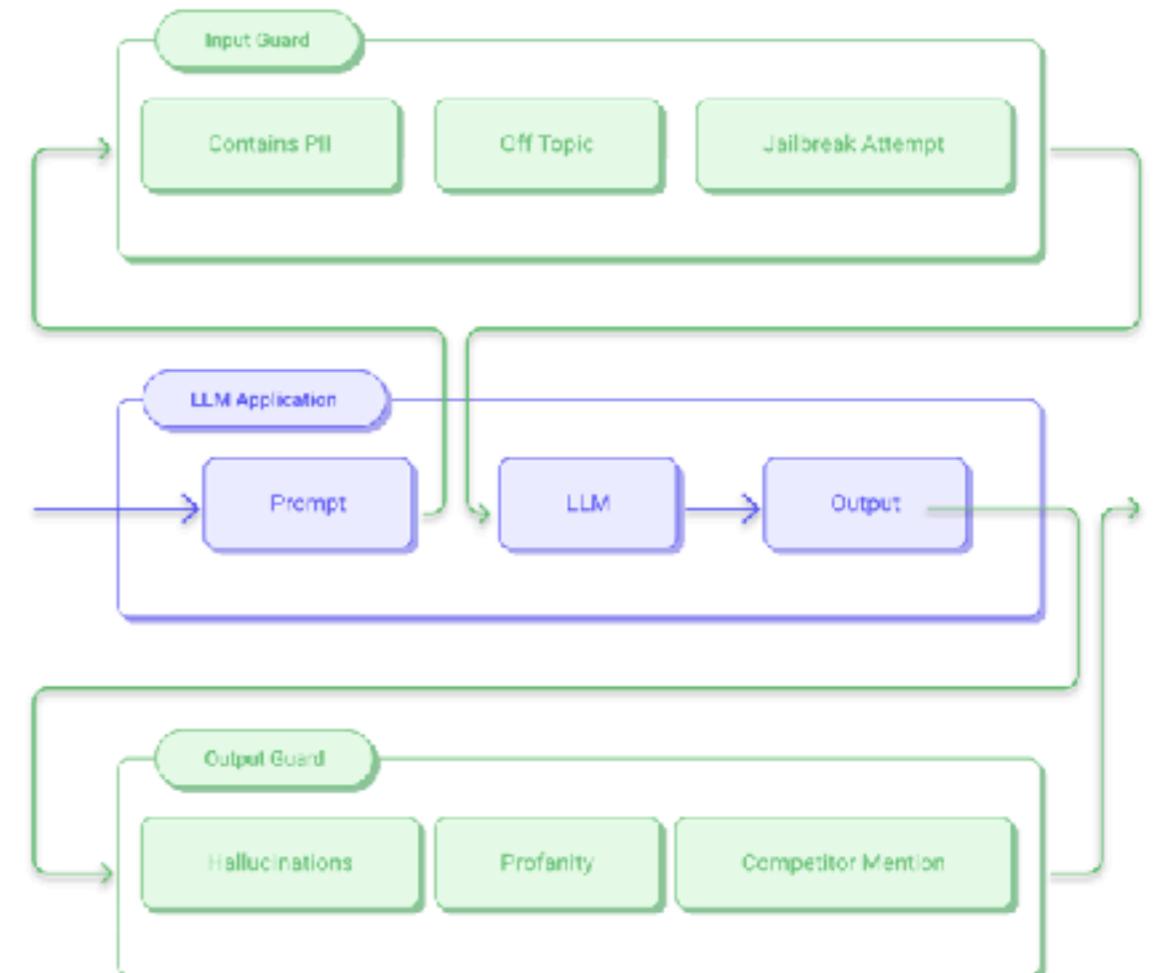


# Guardrails

*Without Guardrails*



*With Guardrails*



# Guardrails Hub

The screenshot shows the 'Validators' section of the Guardrails AI website. On the left, there are several filter categories: 'USE CASES' (Chatbots, Customer Support, Structured Data, RAG, Summarization, Codegen, Text2SQL), 'RISK CATEGORY' (Etiquette, Brand Risk, Factuality, Formatting, Invalid Code, Jailbreaking, Code Exploits, Data Leakage), 'INFRASTRUCTURE REQUIREMENTS' (ML, LLM, NA, Rule), 'CONTENT TYPE' (String, Object, List, Integer, Float, SQL, Code, CSV, Python), 'CERTIFICATION' (Guardrails Certified), and 'LANGUAGE' (EN). The main area displays a grid of 10 validators, each with a title, description, last updated date, input type (String, Brand Risk, Factuality, ML), and a 'Select' button:

- Arize Dataset Embeddings**: Validates that user-generated input does not match the dataset of jailbreak... Last updated 8 months ago. Select button.
- Ban List**: Validates that the output does not contain banned words, using fuzzy search. Last updated 8 months ago. Select button.
- Bespoke MiniCheck**: Validates that the LLM-generated text is supported by the provided context using... Last updated 7 months ago. Select button.
- Bias Check**: Validates that the text is free from biases related to age, gender, sex, ethnicity,... Last updated 1 week ago. Select button.
- Competitor Check**: Flags mentions of competitors. Fixes responses by filtering out competitor names. Last updated 5 months ago. Select button.
- Correct Language**: Validate that an LLM-generated text is in the expected language. If the text is not ... Last updated 11 months ago. Select button.
- Cucumber Expression Match**: Validates that the input string matches a specified cucumber expression. Last updated 6 months ago. Select button.
- Detect PII**: Detects personally identifiable information (PII) in text, using Microsoft Presidio. Last updated 6 months ago. Select button.

<https://hub.guardrailsai.com/>



# Guardrails Index

## AI Guardrails Index

Created by Guardrails AI

[Download PDF](#)

[Register for Webinar](#)

## AI Guardrails Categories

We broke AI safety down into 6 categories and curated datasets and models that demonstrate the state of AI guardrails using LLMs and other open source models.

<https://index.guardrailsai.com/>



# **CAG**

# **(Cache Augmented Generation)**

<https://arxiv.org/abs/2412.15605v1>



# CAG

When you don't need for real-time retrieval (**static**)  
Preload all relevant knowledge into model's **context**  
Precompute key-value(KV) cache to store and reuse  
**Larger context window size !! (1M)**

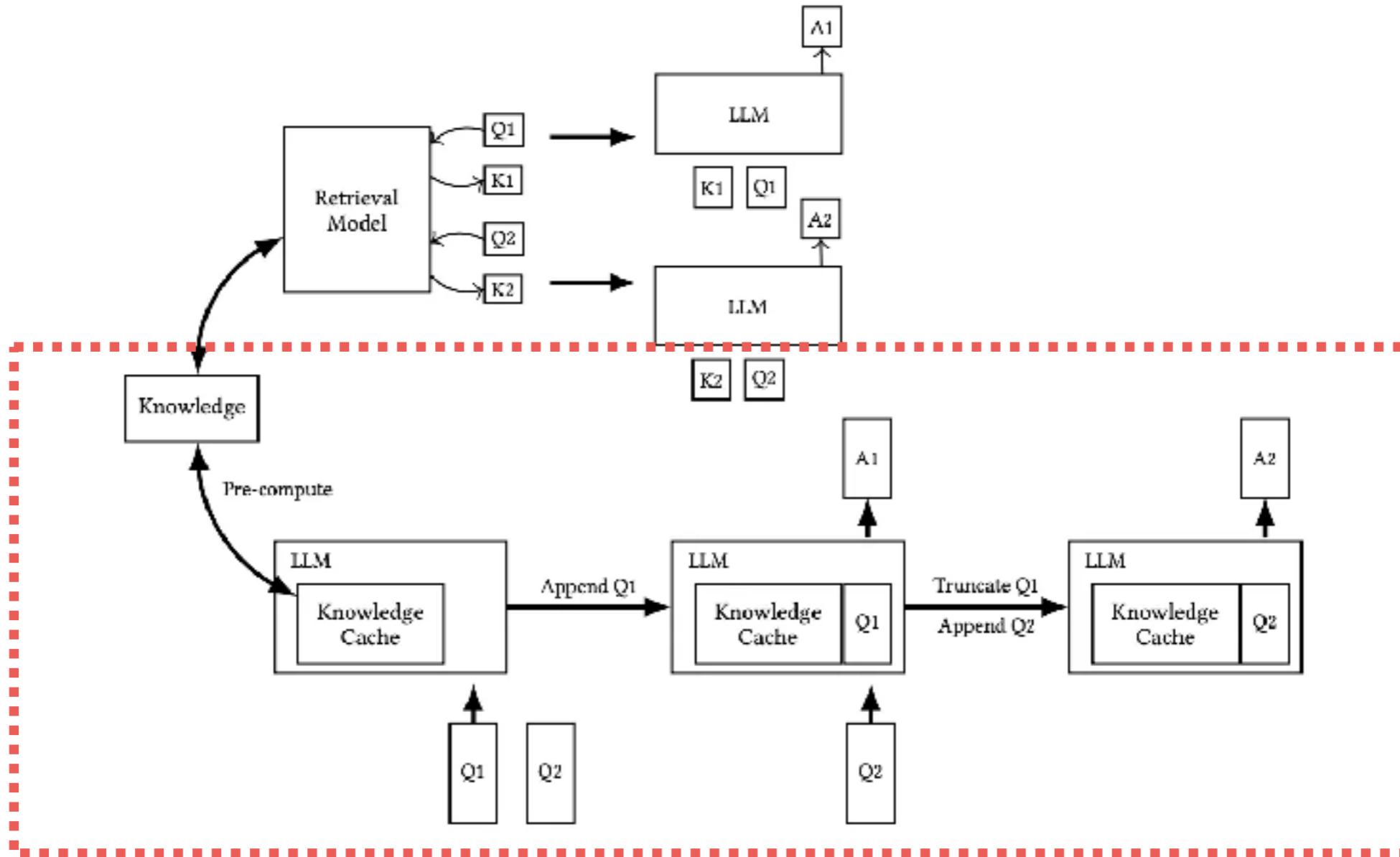
Retrieval-free  
operation

Improve  
efficiency

Simplify  
architecture

<https://arxiv.org/abs/2412.15605v1>

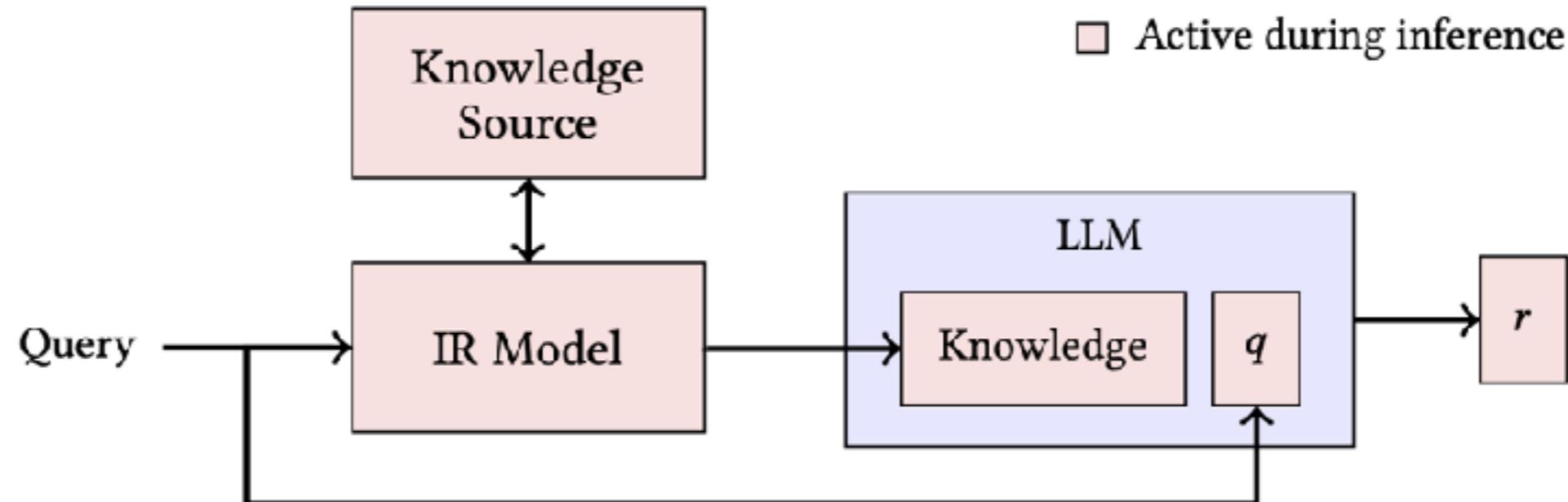




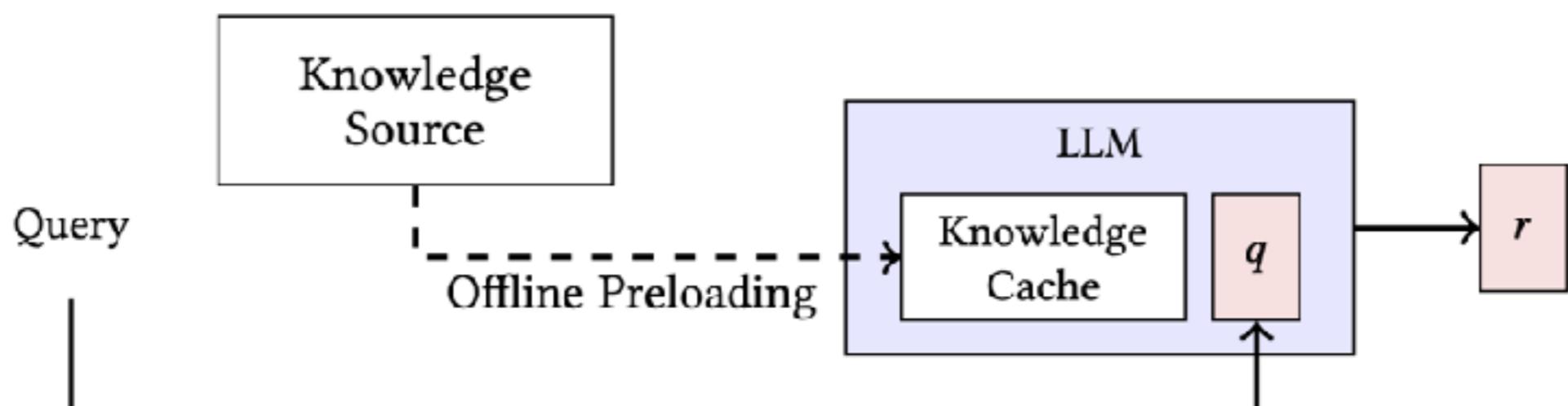
**Figure 1: Comparison of Traditional RAG and our CAG Workflows:** The upper section illustrates the RAG pipeline, including real-time retrieval and reference text input during inference, while the lower section depicts our CAG approach, which preloads the KV-cache, eliminating the retrieval step and reference text input at inference.

<https://arxiv.org/abs/2412.15605v1>





Retrieval-Augmented Generation



Cache-Augmented Generation

<https://github.com/hhuang/CAG>



# RAG + CAG is better

Cached  
context

Your  
question

Retrieved  
context

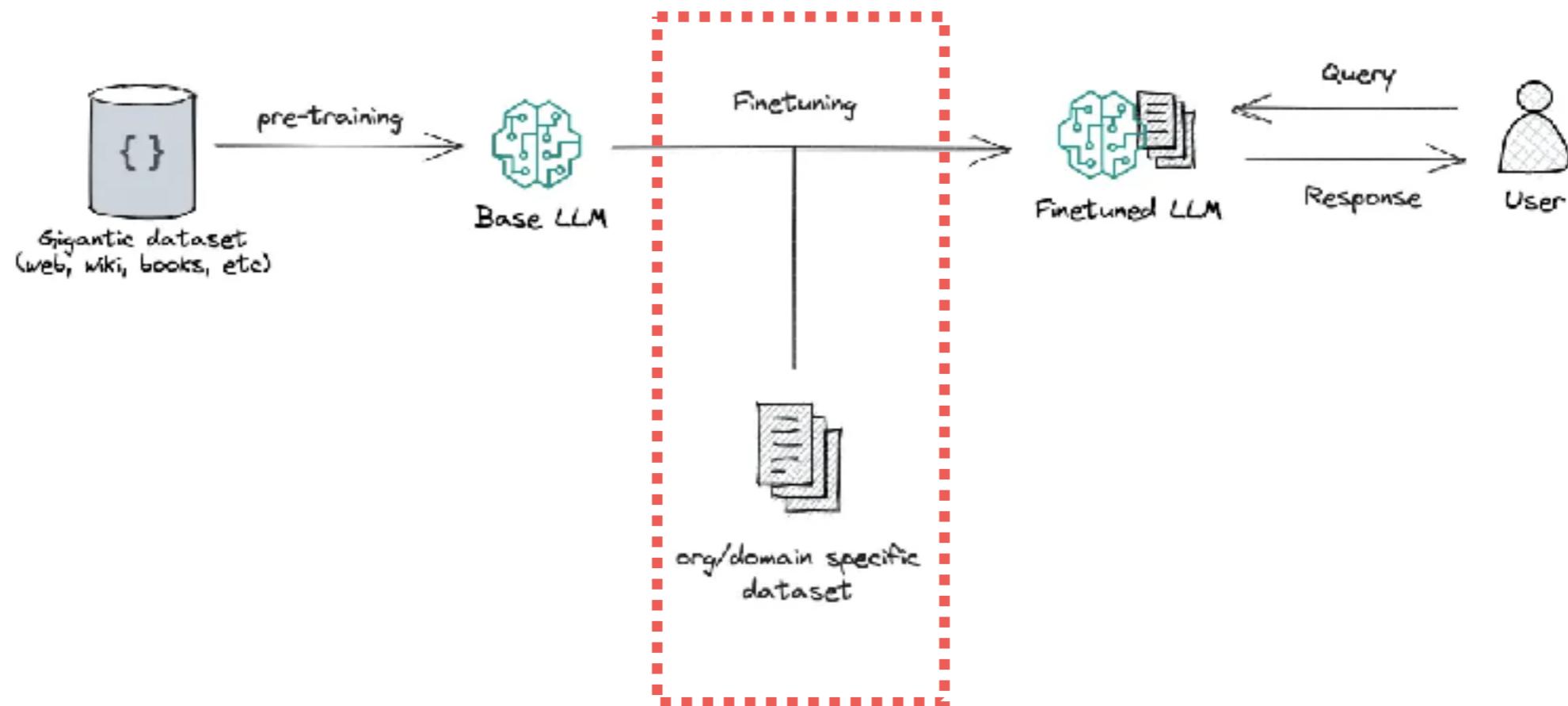


# Fine Tuning



# Fine Tuning

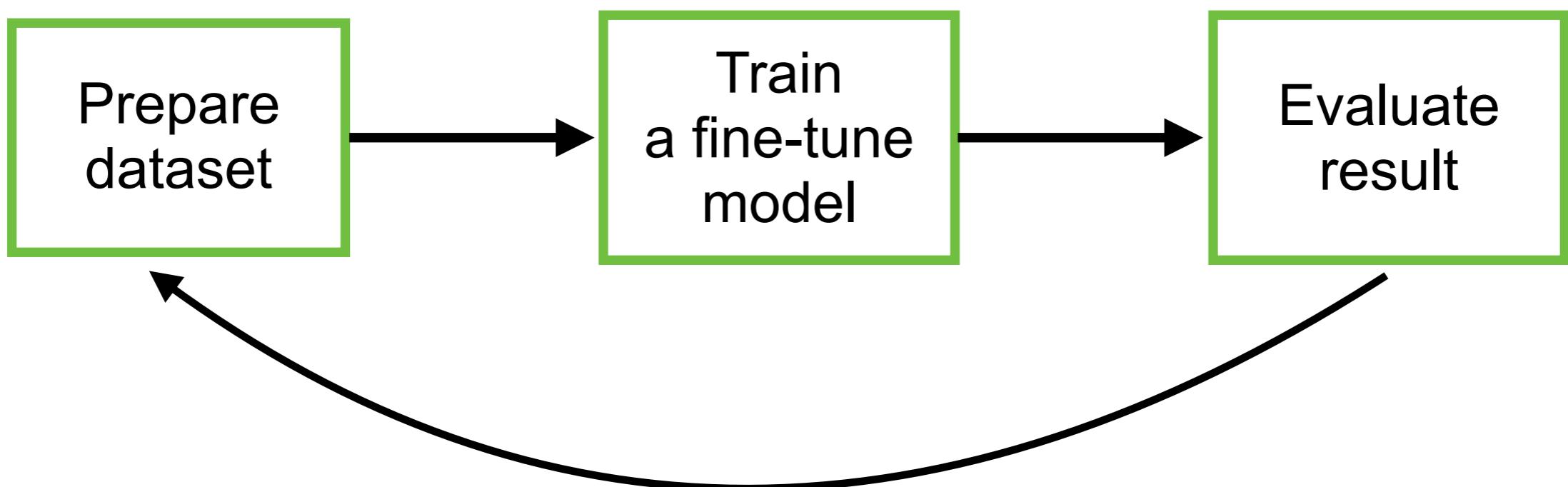
Training with small and specific dataset  
Adjust the model's weight based on our data



<https://towardsdatascience.com/rag-vs-finetuning-which-is-the-best-tool-to-boost-your-lm-application-94654b1eaba7>



# Fine Tuning Process



<https://platform.openai.com/docs/guides/fine-tuning>



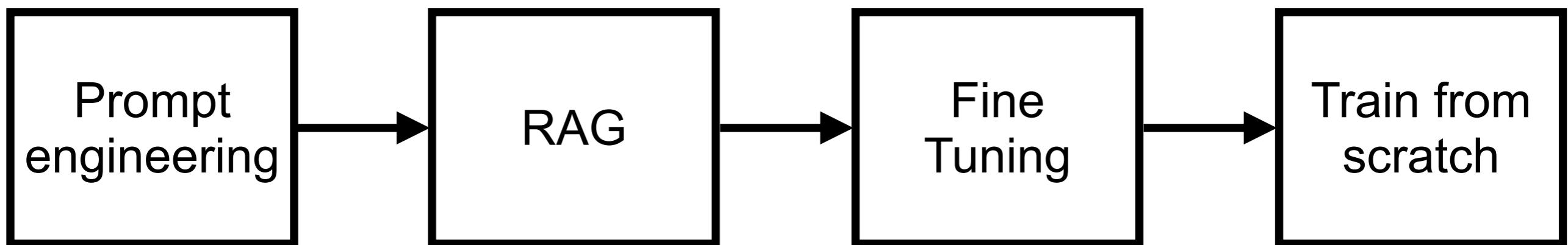
# RAG vs Fine-tuning

Scenario	RAG preferred	Fine tune preferred
Skillset requirements	Strong in RAG engineering	Strong in deep learning model and fine tuning
Data freshness	Realtime or frequently updated	Static, domain-specific data
Domain complexity	Multiple domains or high data diversity	Specialized and heavy domain (medical)
Resources usage	Lower computation	Higher computation



# Steps with LLM

Balance with complexity, cost and quality



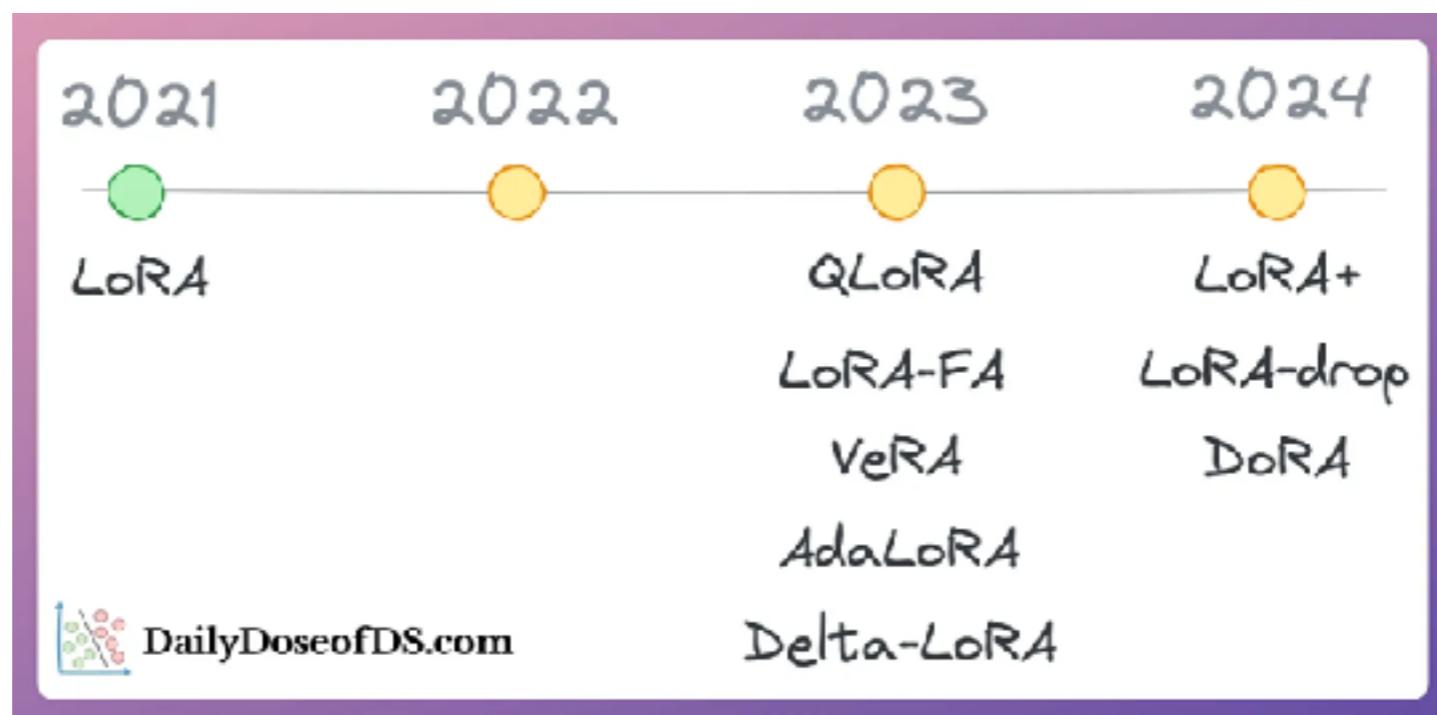
# Fine Tuning Techniques !!

LoRA (Low-Rank Adaptation)

LoRA-FA (Frozen-A)

VeRA (Vector-based Random Matrix Adaptation)

Delta-LoRA

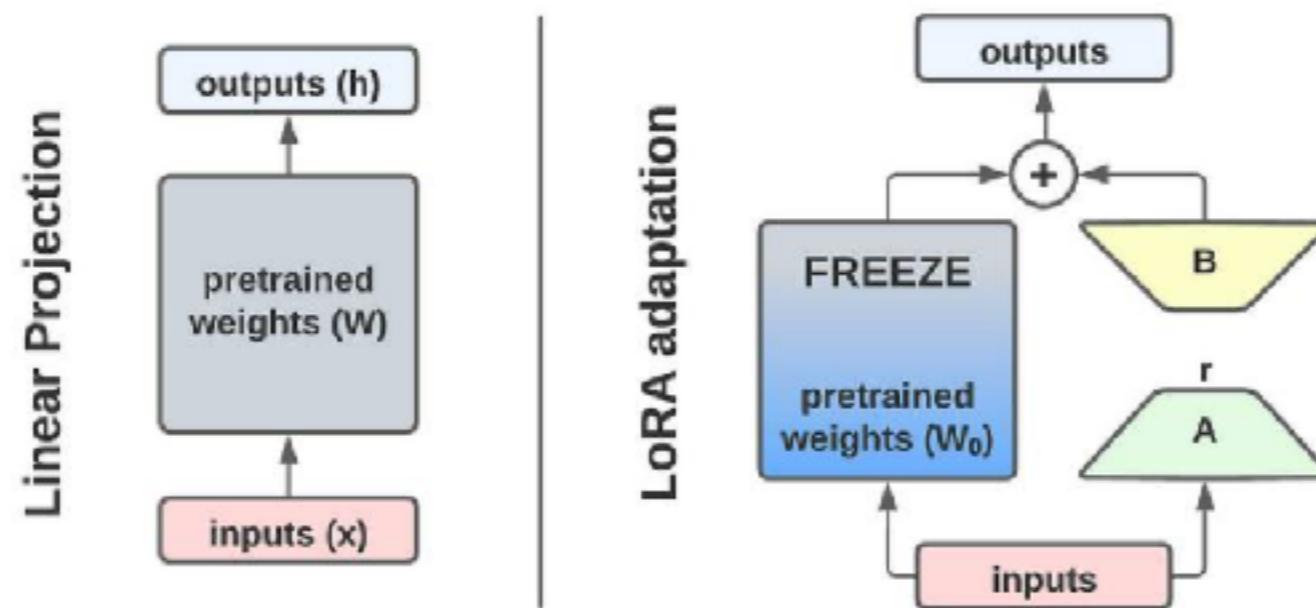


<https://blog.dailydoseofds.com/p/5-llm-fine-tuning-techniques-explained>



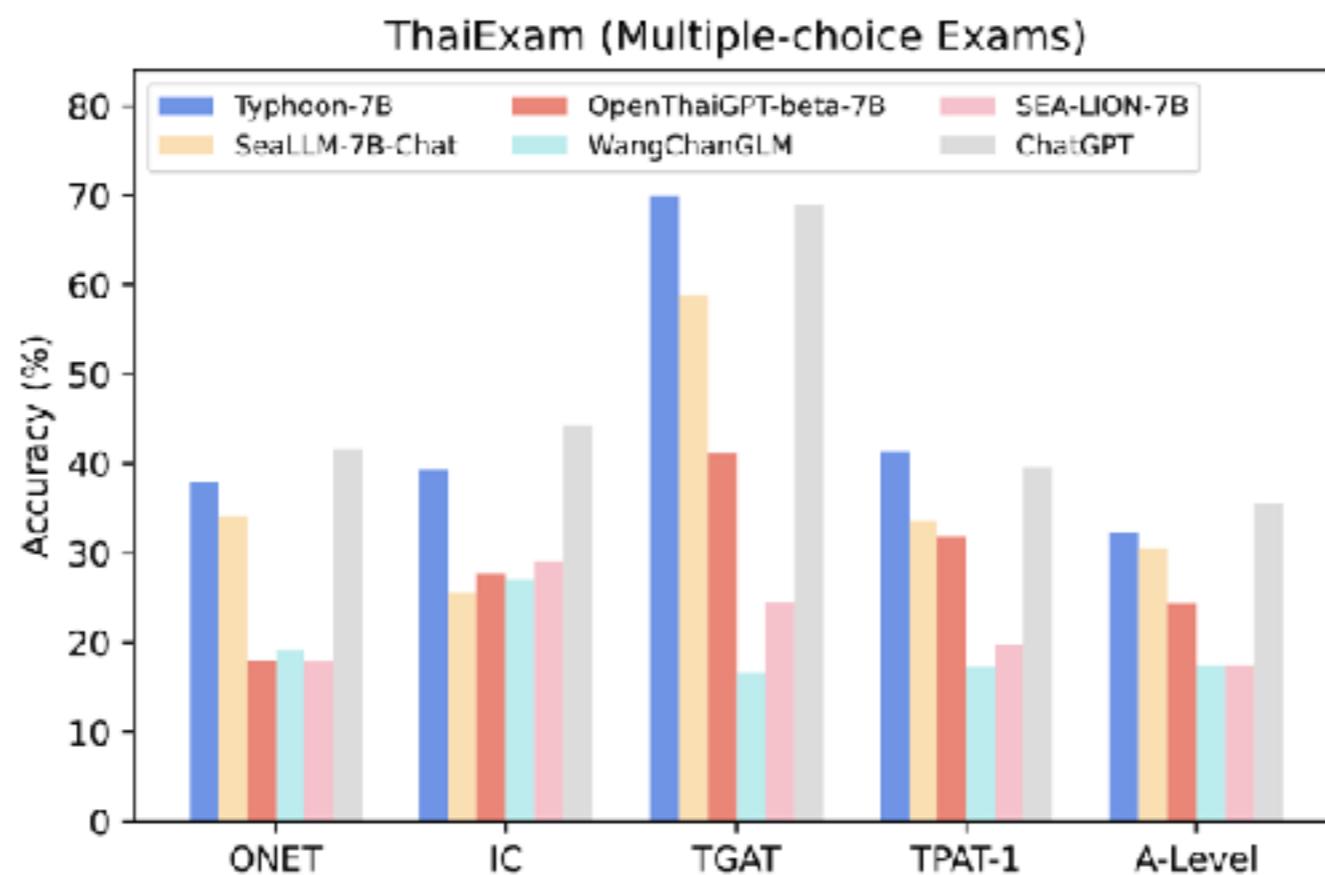
# LoRA (Low-Rank Adaptation)

Reduces the number of trainable parameters  
Making fine-tuning faster  
More memory-efficient



# Typhoon-7B Thai LLM

Based-on Mistral-7B  
Thai words 5,000+

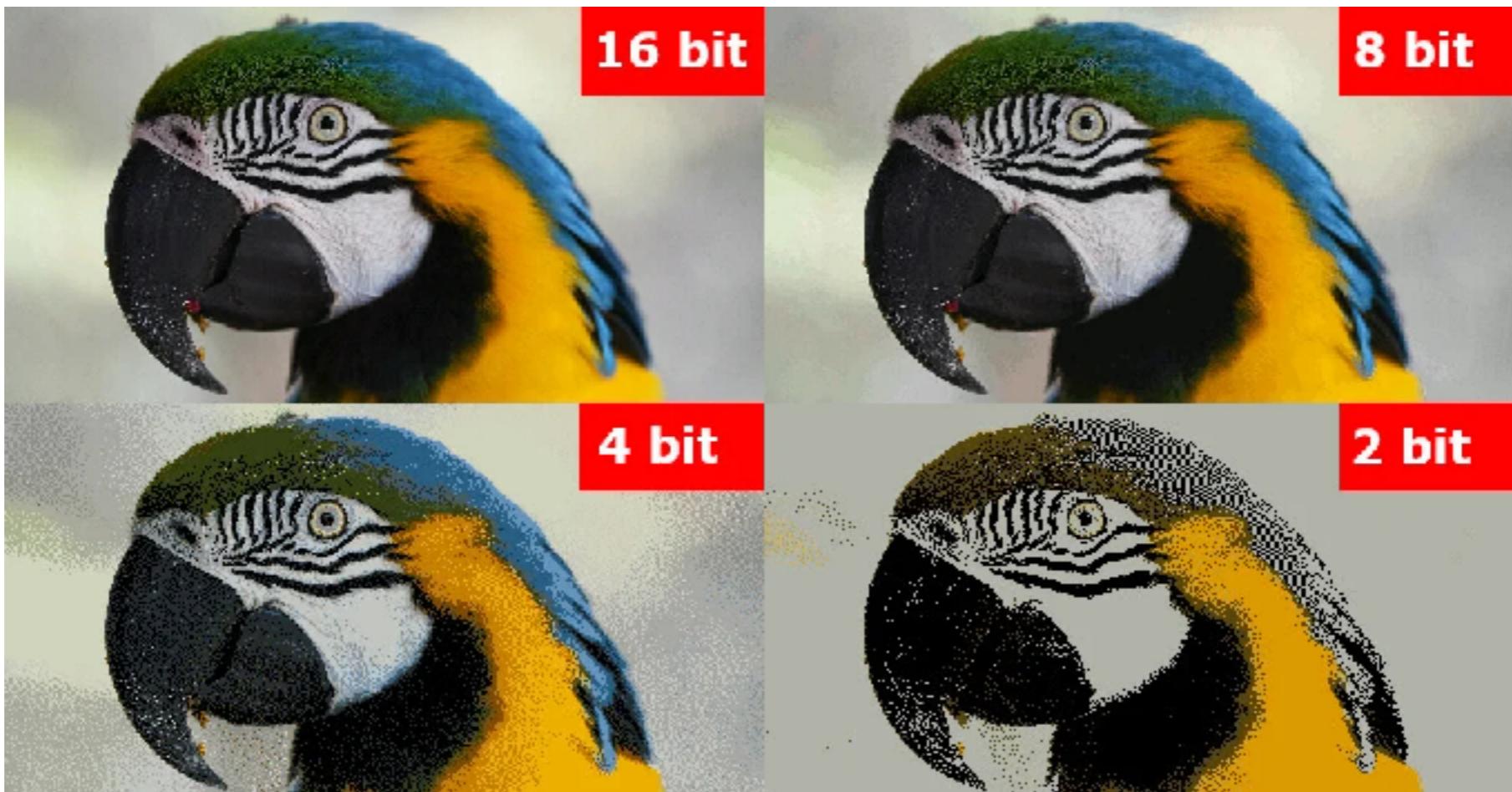


<https://huggingface.co/scb10x/typhoon-7b>



# Reduce size of Model ?

## Quantization



# Quantization in LLM model

Reduce size of model  
Reduce resource usage  
Faster

Model	Original Size (FP32)	Quantized Size (INT4)
LLaMA3.1-8B	38.4 GB	4.8 GB
LLaMA3.1-70B	336 GB	42 GB
LLaMA3.1-405B	1,994 GB	243 GB

<https://medium.com/@lmpo/understanding-model-quantization-for-langs-1573490d44ad>



# Local LLM



# Local LLM

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

Reduce cost

Data privacy

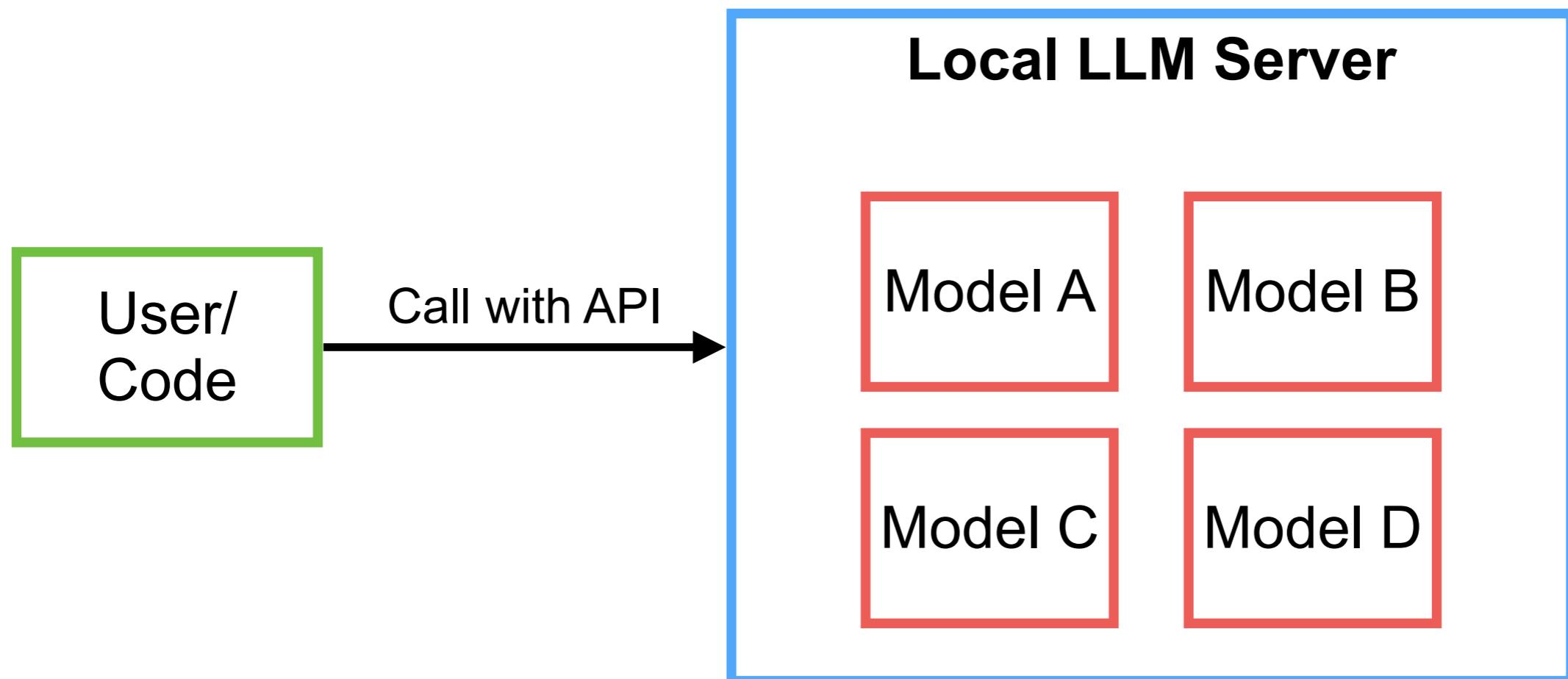
Responsive

Offline mode



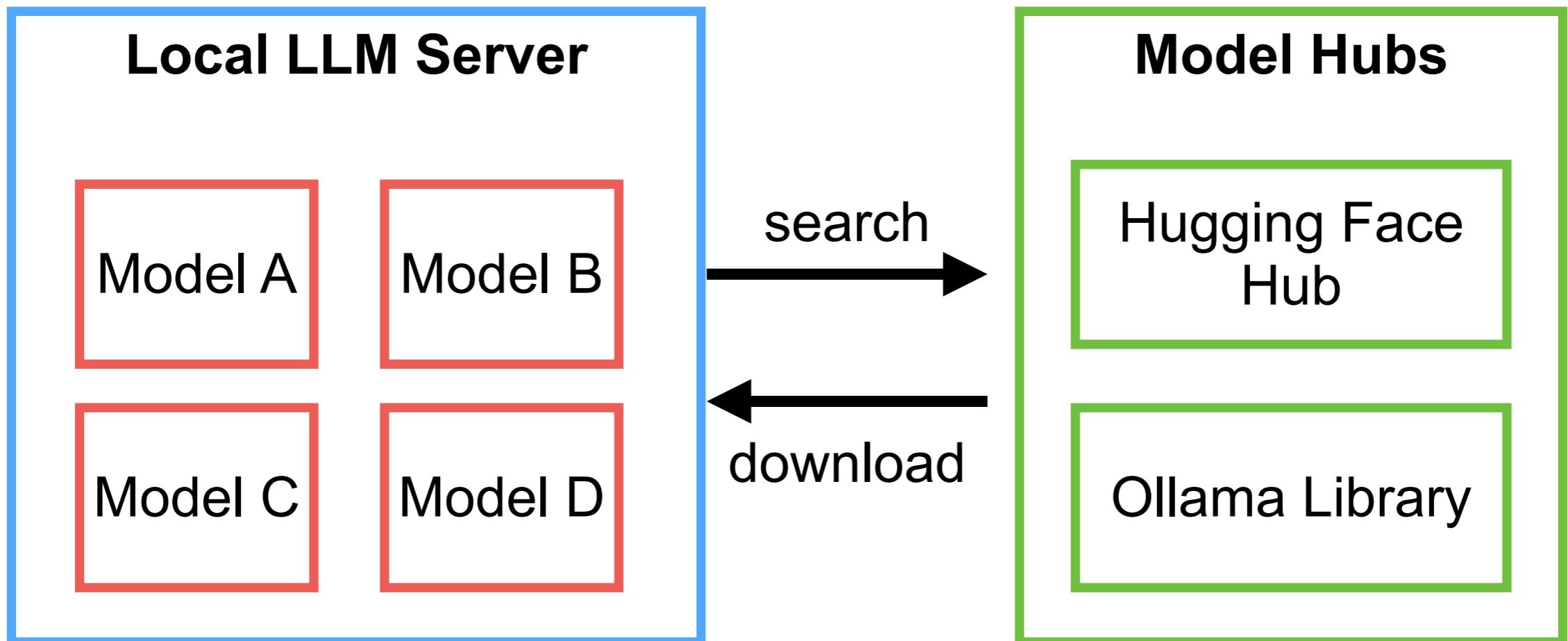
# Local LLM

Improve your LLM models, more accurate answer



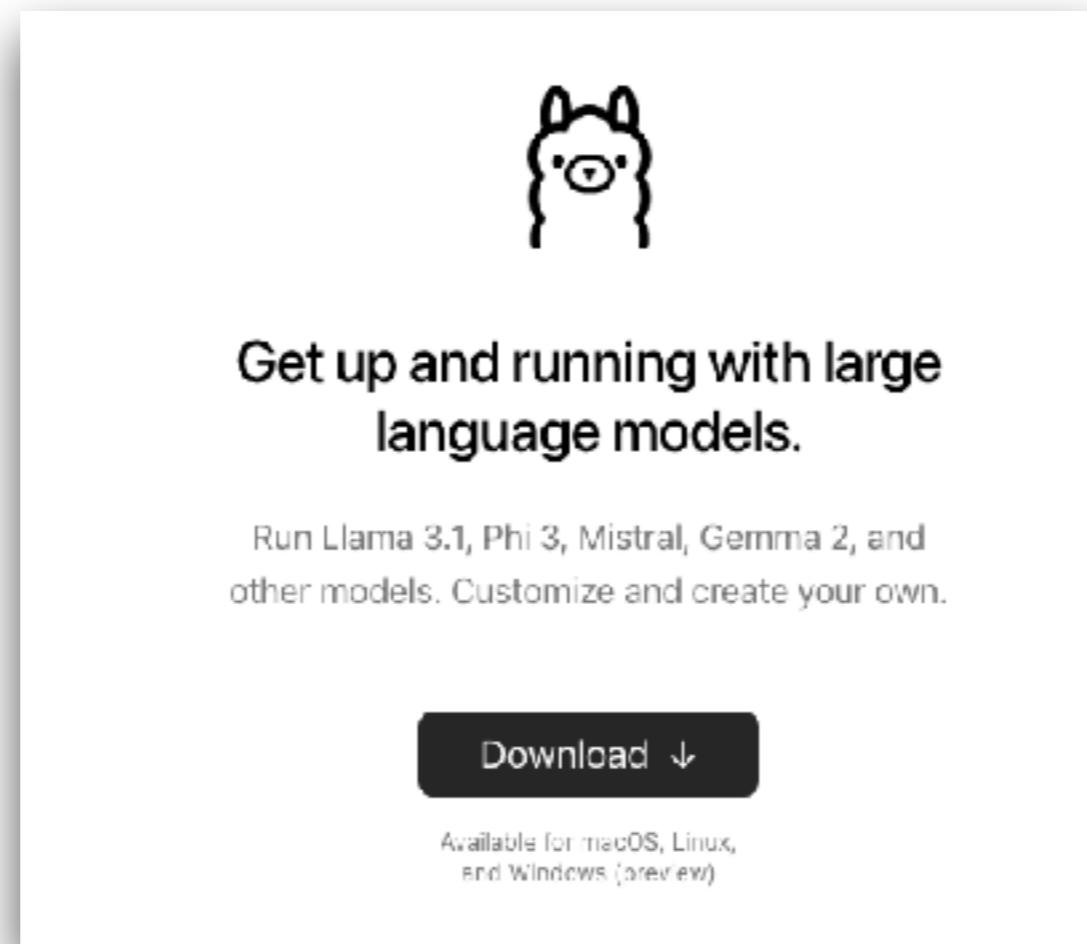
# Models ?

How to download models ?



# Local LLM with Ollama

\$ollama run **llama3.1**



<https://ollama.com/>



# Model in Ollama

**gemma3**  
The current, most capable model that runs on a single GPU.

[vision](#) [1b](#) [4b](#) [12b](#) [27b](#)

3.8M Pulls 21 Tags Updated 7 days ago

---

**qwo**  
QwQ is the reasoning model of the Qwen series.

[tools](#) [32b](#)

1.4M Pulls 8 Tags Updated 6 weeks ago

---

**deepseek-r1**  
DeepSeek's first-generation of reasoning models with comparable performance to OpenAI-o1, including six dense models distilled from DeepSeek-R1 based on Llama and Qwen.

[1.5b](#) [7b](#) [8b](#) [14b](#) [32b](#) [70b](#) [671b](#)

39.6M Pulls 29 Tags Updated 2 months ago

<https://ollama.com/library>



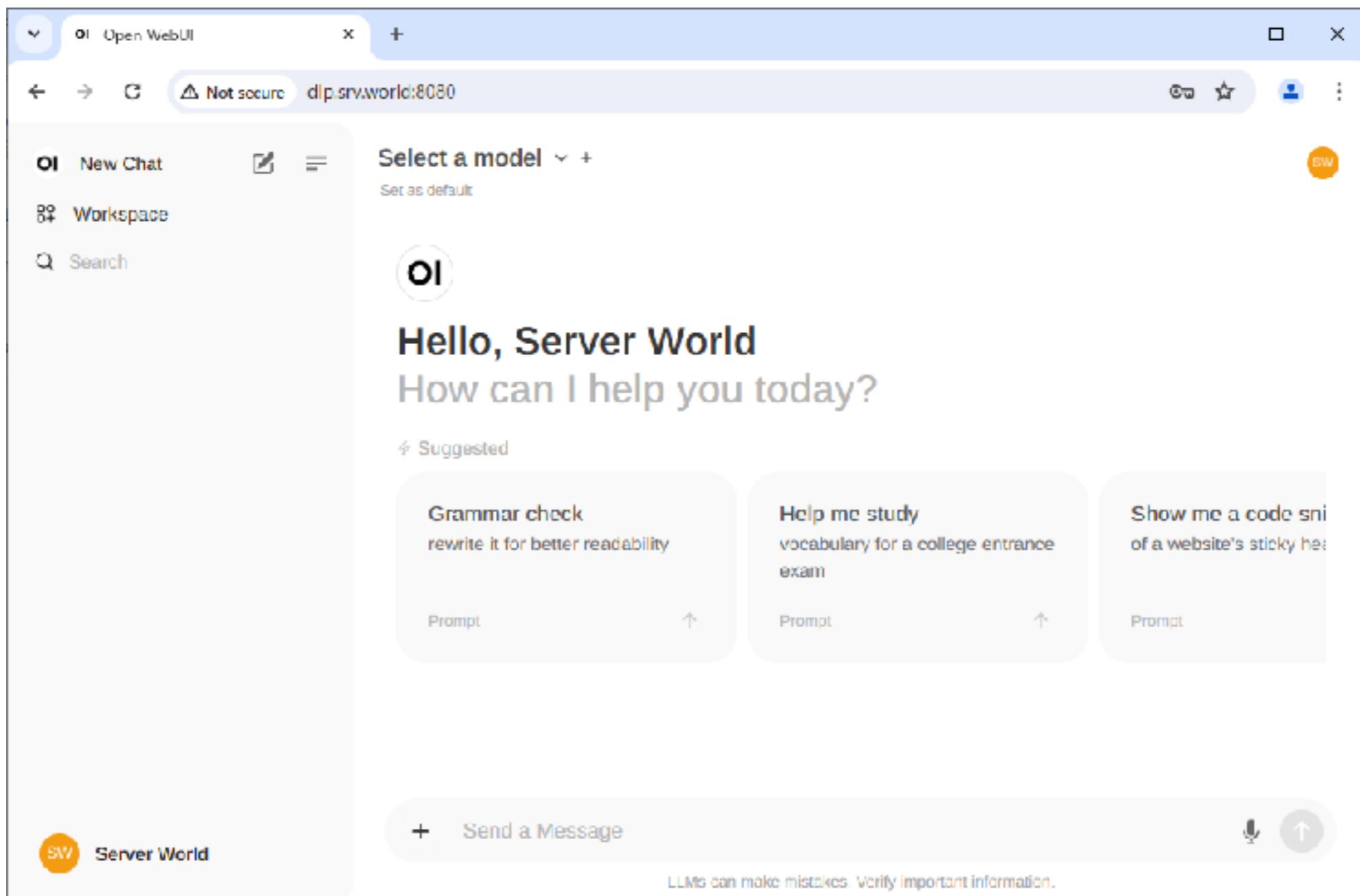
# LiteLLM



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



# OpenWebUI



<https://openwebui.com/>



# Working with No-Code



# Prompt Flow



# Prompt flow

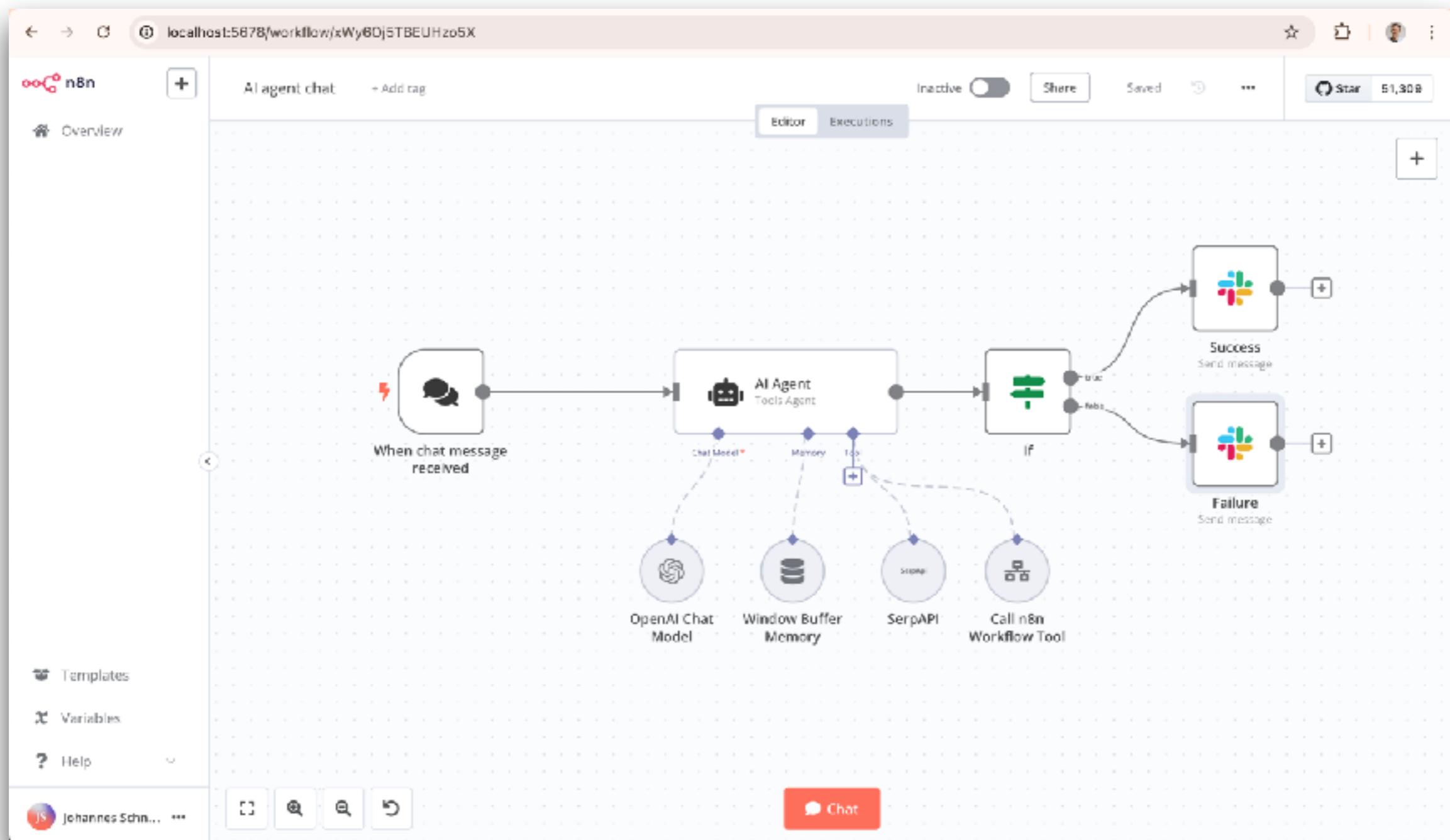
Build high-quality LLM apps - from prototyping, testing to production deployment and monitoring



<https://microsoft.github.io/promptflow/>



# N8N



<https://n8n.io/>



# Let's go !!

