LLM

Jiqiong QIU

Objectif du cours

- Le but de ce cours est l'introduction aux modèles LLM (large language model)
- Le cours se focalise essentiellement sur les aspects suivants :
  - *Introduction à LLM*
  - *Prompt engineering*
  - *Semantic search*
  - *Longchain/ LllamaIndex*
  - *RAG*
  - *Function calling*

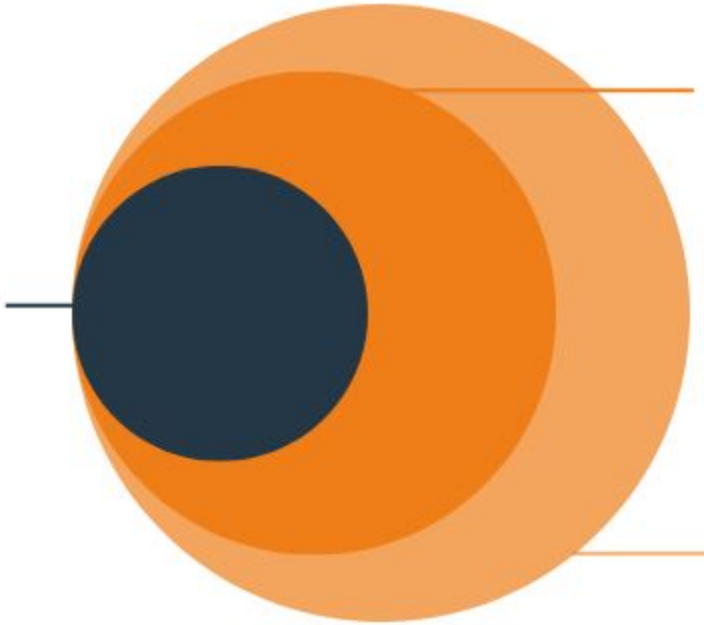
LLM

GNOMON DIGITAL

Modèle

Le Deep learning (apprentissage profond)

Les systèmes apprennent par les réseaux neuronaux sans intervention de l'homme.



Moyenne

Le machine learning (apprentissage automatique)

Les systèmes sont programmés pour apprendre à réaliser des tâches en étudiant un ensemble de vastes données

Objective

L'Intelligence Artificielle (IA) ou (AI) pour Artificial Intelligence

Les systèmes reproduisent le fonctionnement de l'intelligence humaine.

AI Générative

- Plus besoin de connaissance ML
- Pas besoin d’apprentissage
- Pas besoin des données
- Just besoin de bien formuler ton demande (prompt)

LLM

GNOMON DIGITAL

AI Builders
LE LEADER DE LA STRATÉGIE DATA ET IA

LLM Providers Landscape

Generative language models

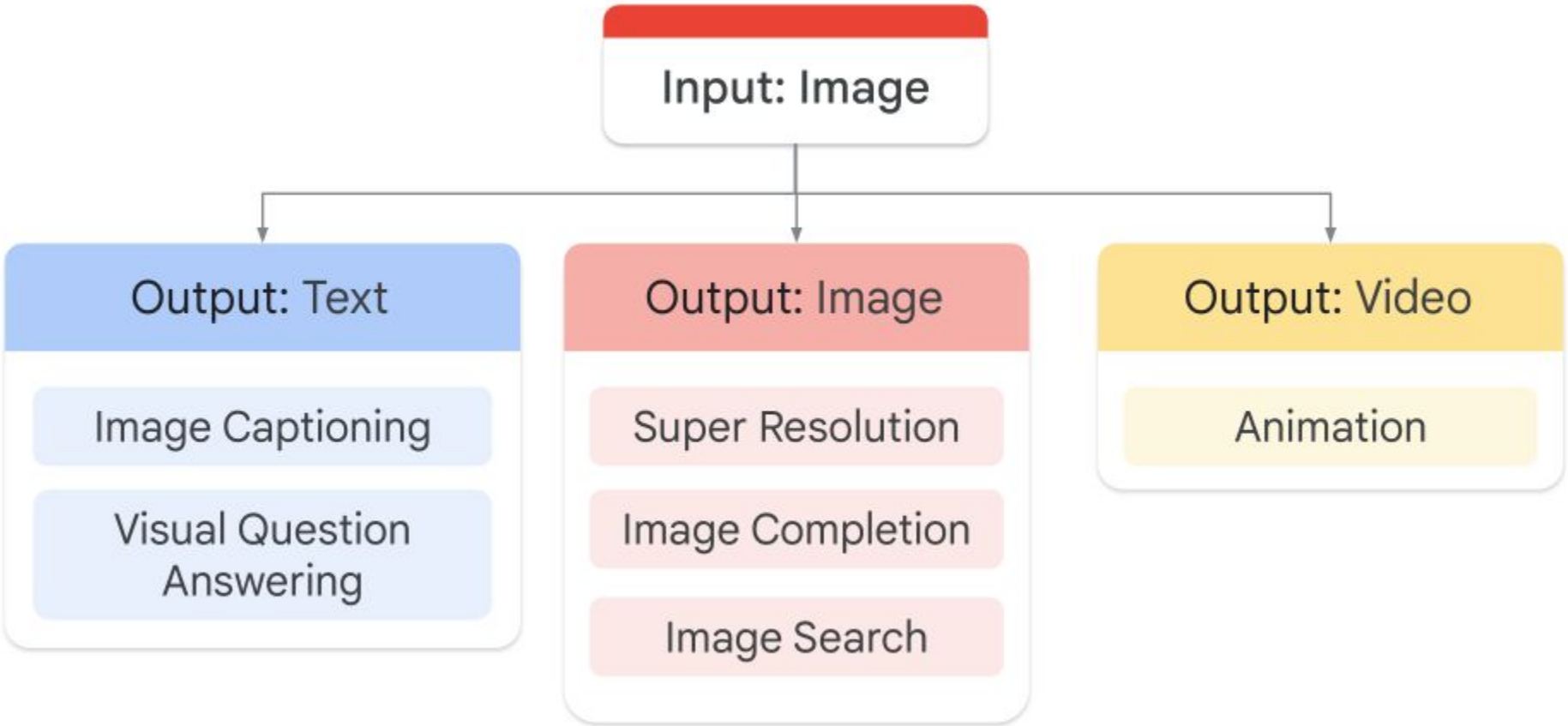
Input: Text
Output: Text

Generative image models

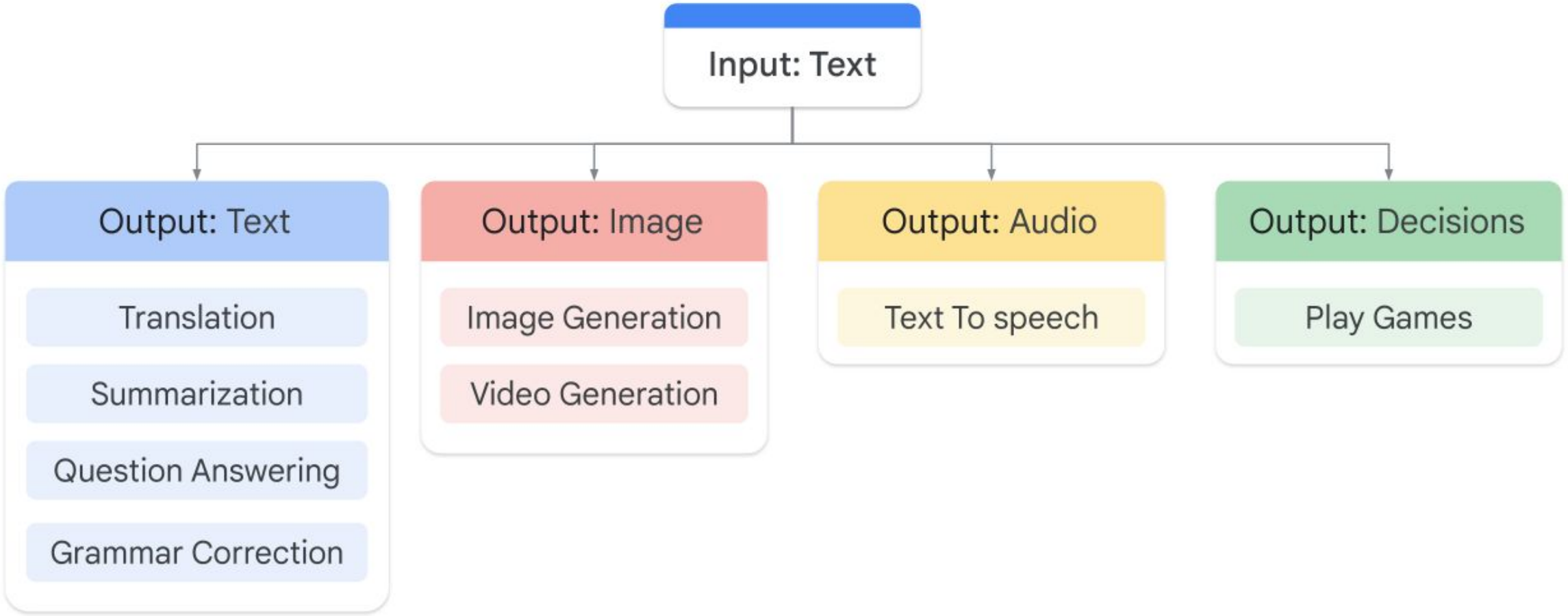
Input: Text
Output: Image

Multimodal:
Input: Text, audio, images, vidéo
Output: Text, audio, images, vidéo

LLM


Input: Image
Output: Text
Image Captioning
Visual Question Answering
Output: Image
Super Resolution
Image Completion
Image Search
Output: Video
Animation

LLM

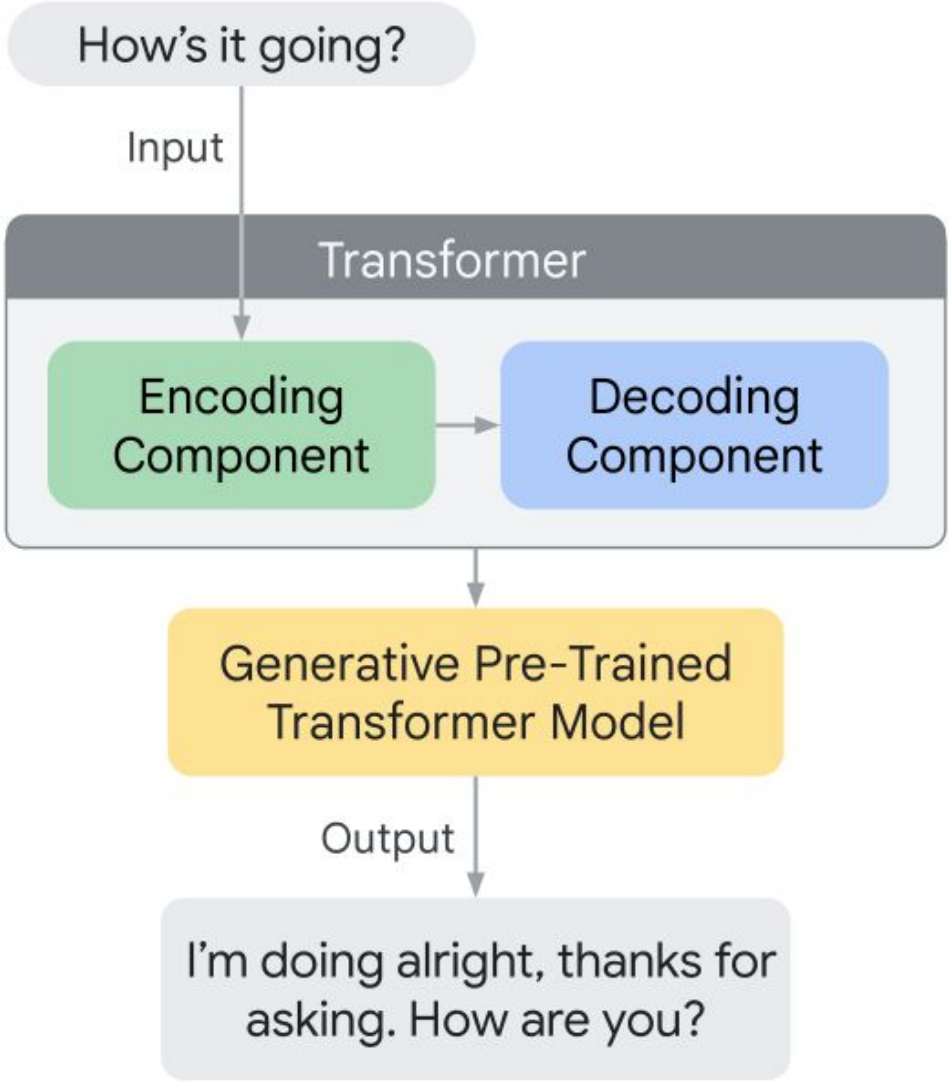

Input: Text
Output: Text
Translation
Summarization
Question Answering
Grammar Correction
Output: Image
Image Generation
Video Generation
Output: Audio
Text To speech
Output: Decisions
Play Games

LLM

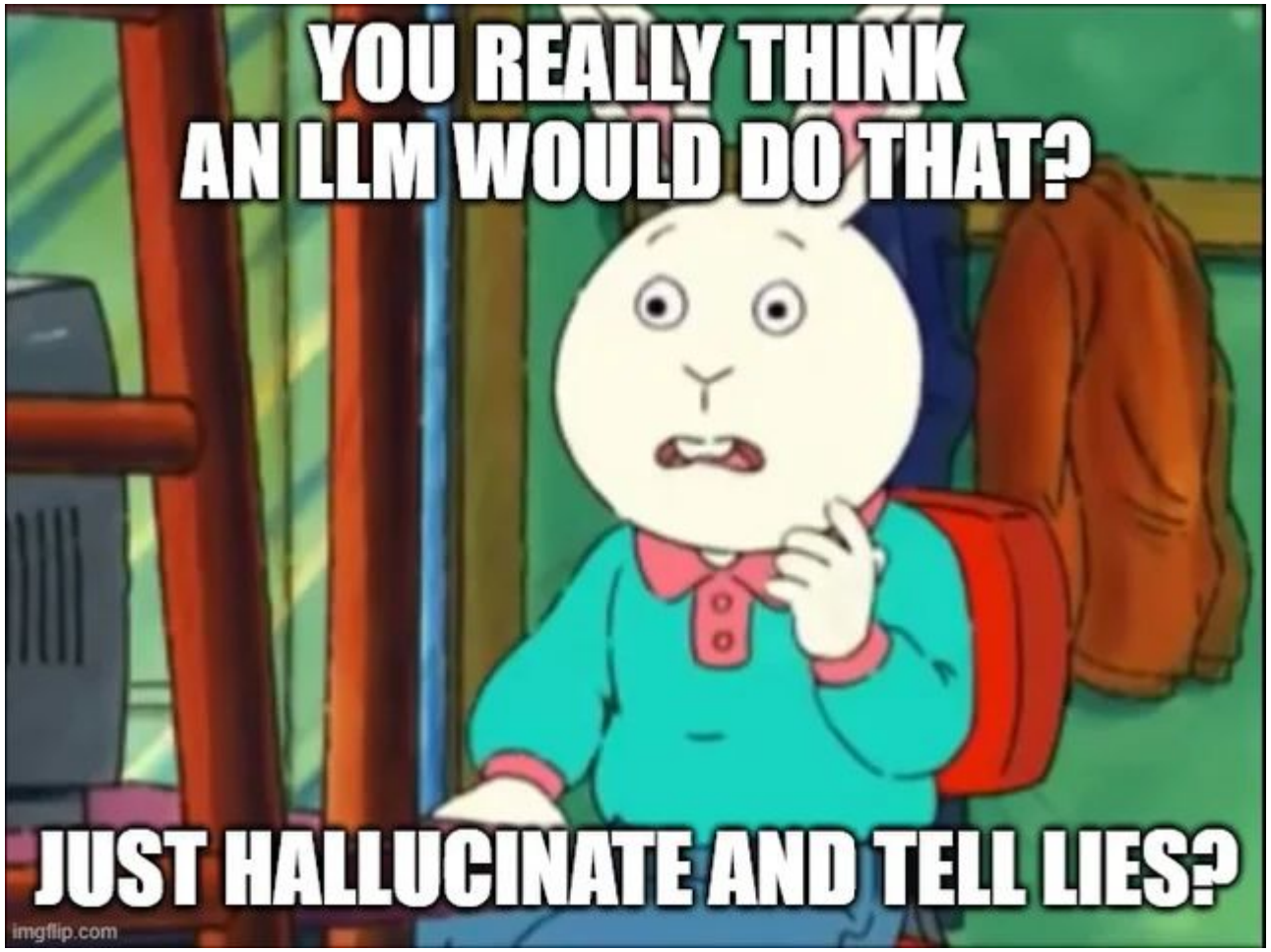
GNOMON DIGITAL

Comment ça marche?

- Grand volume des données
- Billions parameters
- Apprentissage non-supervisé



LLM


YOU REALLY THINK
AN LLM WOULD DO THAT?
JUST HALLUCINATE AND TELL LIES?
imgflip.com

LLM

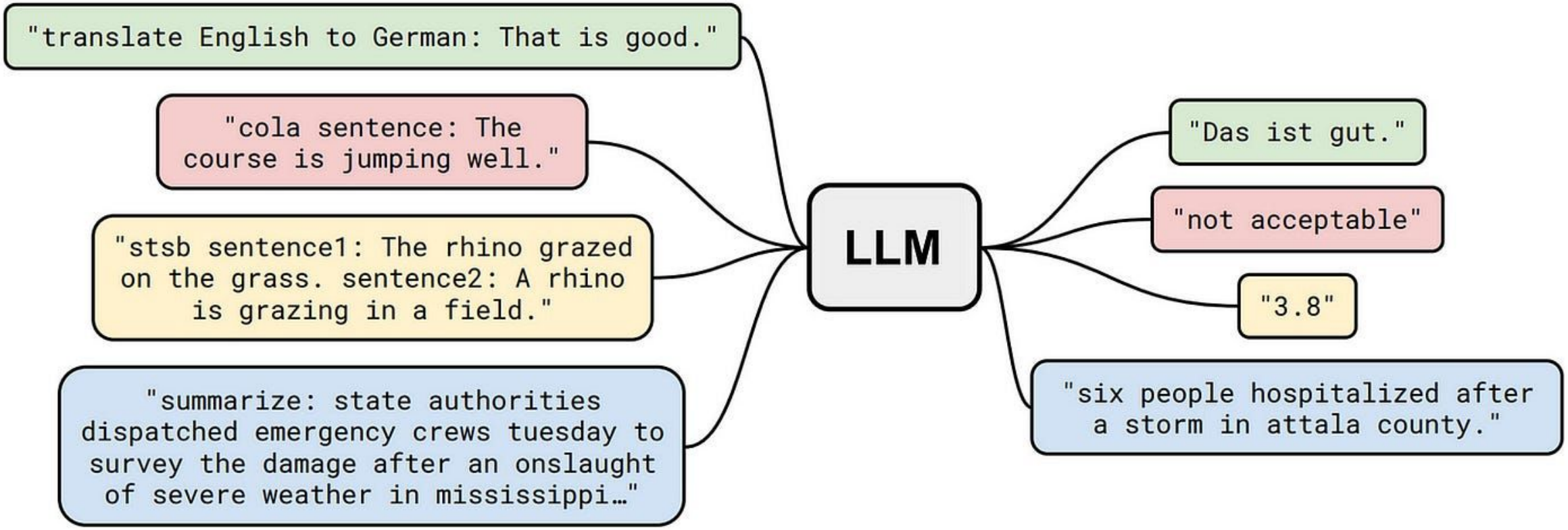
GNOMON DIGITAL

- Hallucinations:
  - Le modèle n’a pas assez des données pour faire l’entrainement
  - Le modèle est entraîné sur des données sales avec des bruits
  - On manque des context
  - Pas assez de contraintes sur le modèle

Prompt Engineering

GNOMON DIGITAL

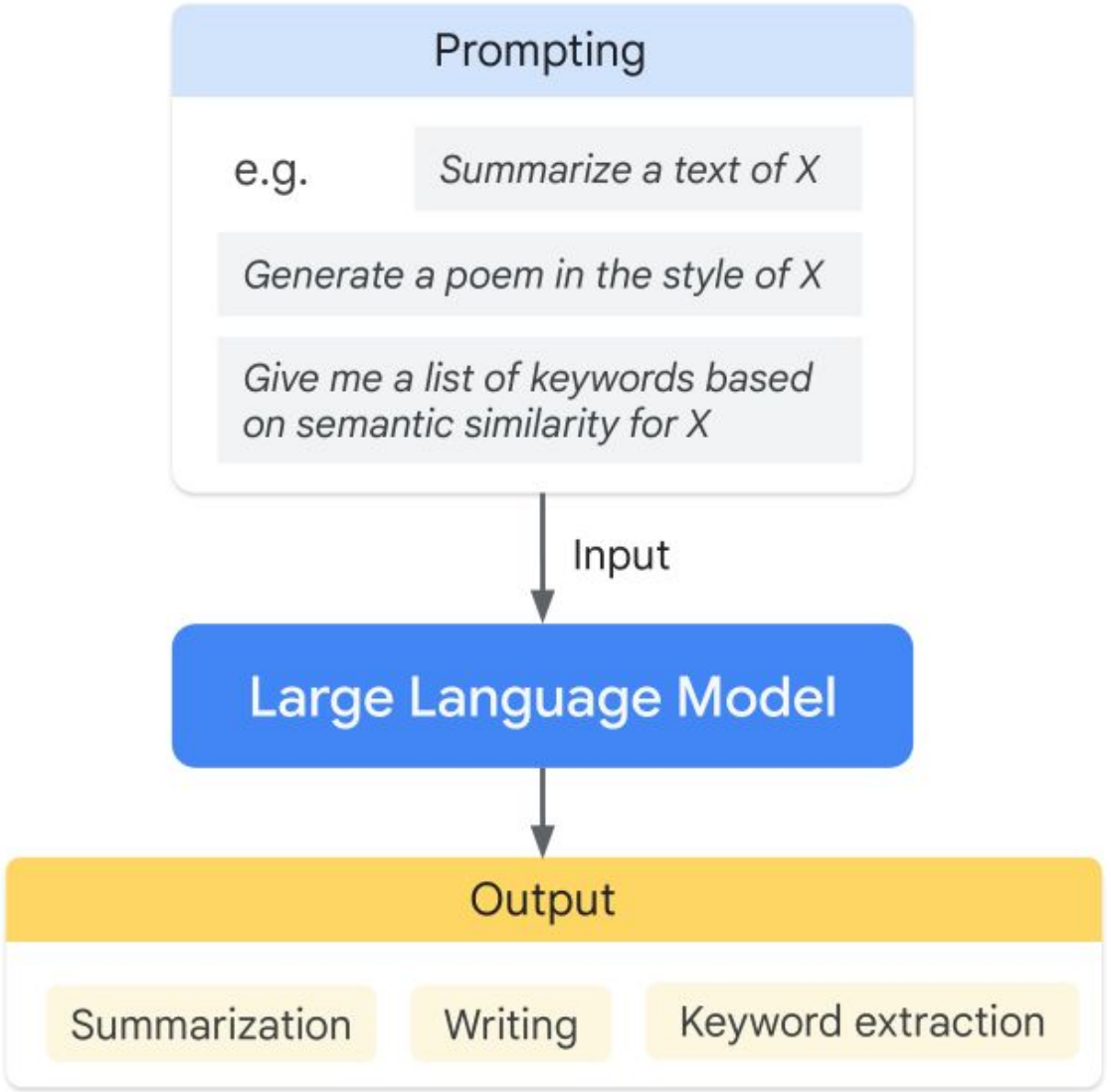
- On donne la context au modèle



Prompt Engineering

GNOMON DIGITAL

- Prompt engineering: la qualité d'entrée détermine la qualité de sortie



LLM

GNOMON DIGITAL

- Comment détermine un LLM

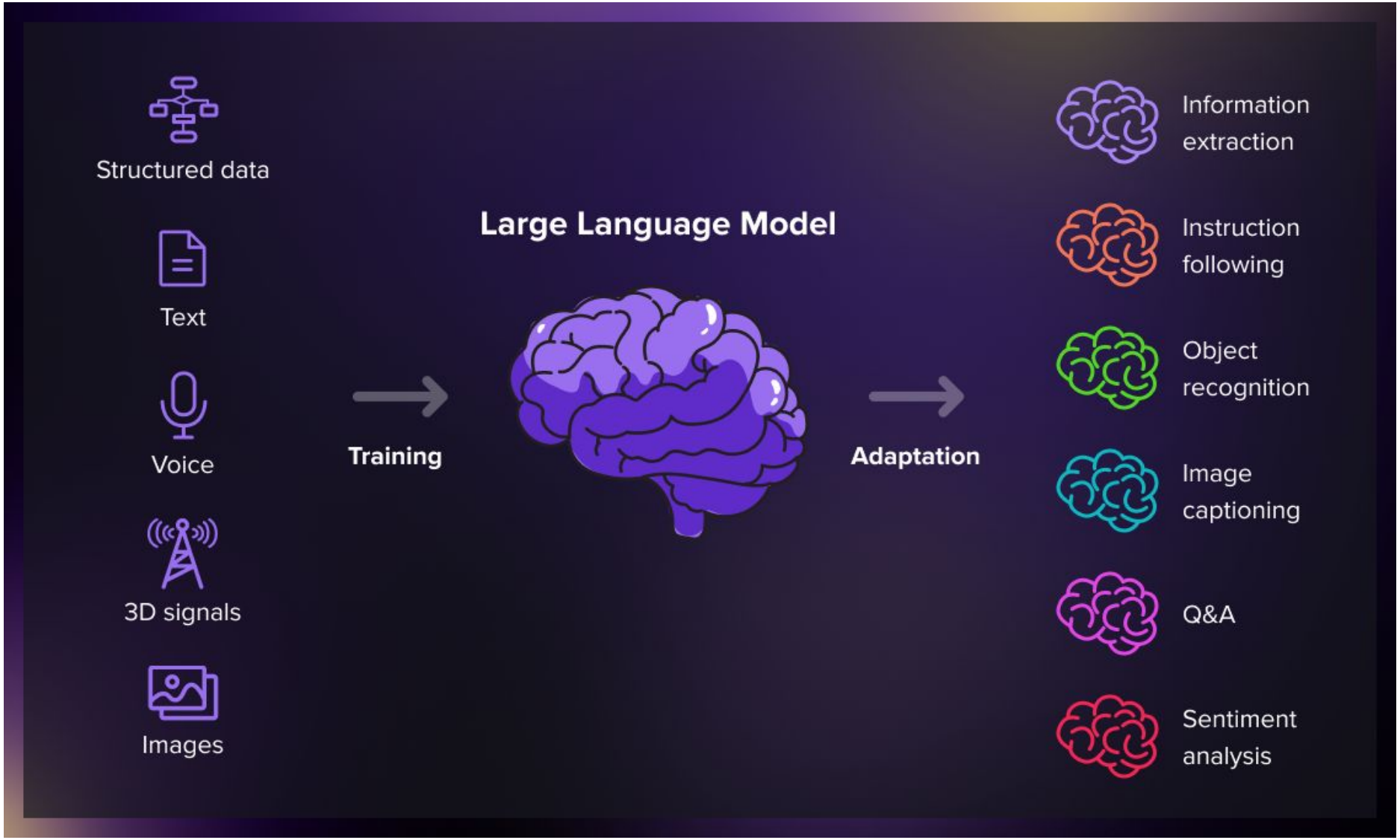
Forming a Database

Inputting a Prompt

Generating content

LLM

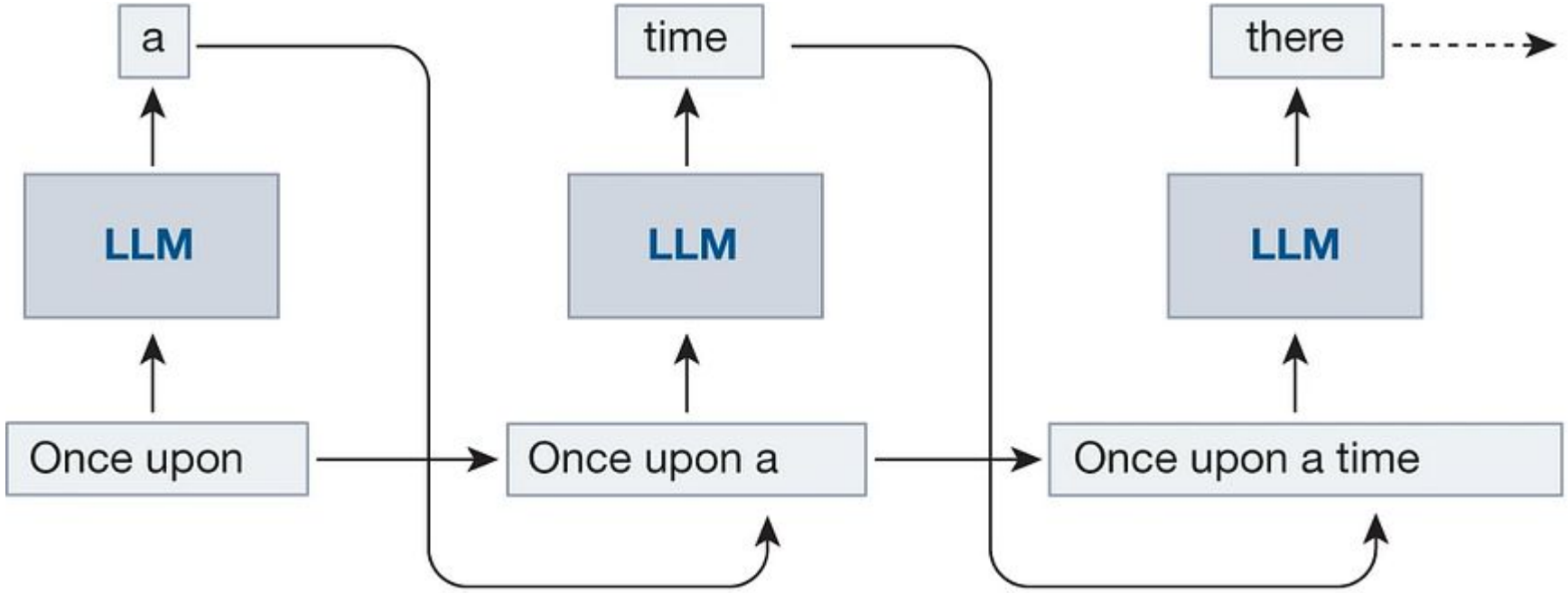
GNOMON DIGITAL



LLM: type de modèles

- Comment détermine un LLM

Text-to-text



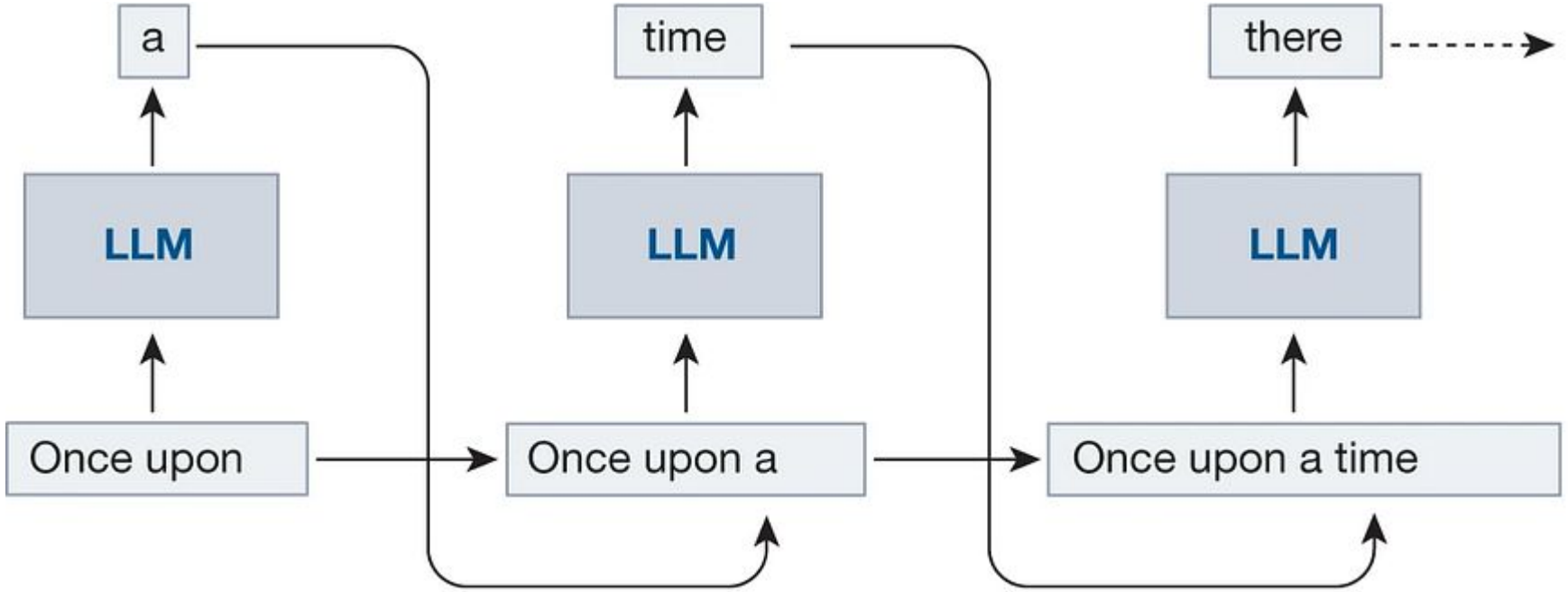
https://chatgpt.com/

LLM: type de modèles

- Comment détermine un LLM

Text-to-text

Domain d'application:
- Génération des text
- Classification
- Résumé
- Traduction
- Reformuler
- Extraction
- Clustering



https://chatgpt.com/

LLM: type de modèles

- Comment détermine un LLM

Text-to-image

Domain d'application:
- Image generation
- Image edition

https://www.midjourney.com/

LLM: type de modèles

- Comment détermine un LLM

Text-to-3D
(video)

Domain d'application:
- Video generation
- Video edition
- Jeux Vidéo

https://modal.com/blog/text-to-video-ai-article

LLM: type de modèles

- Comment détermine un LLM

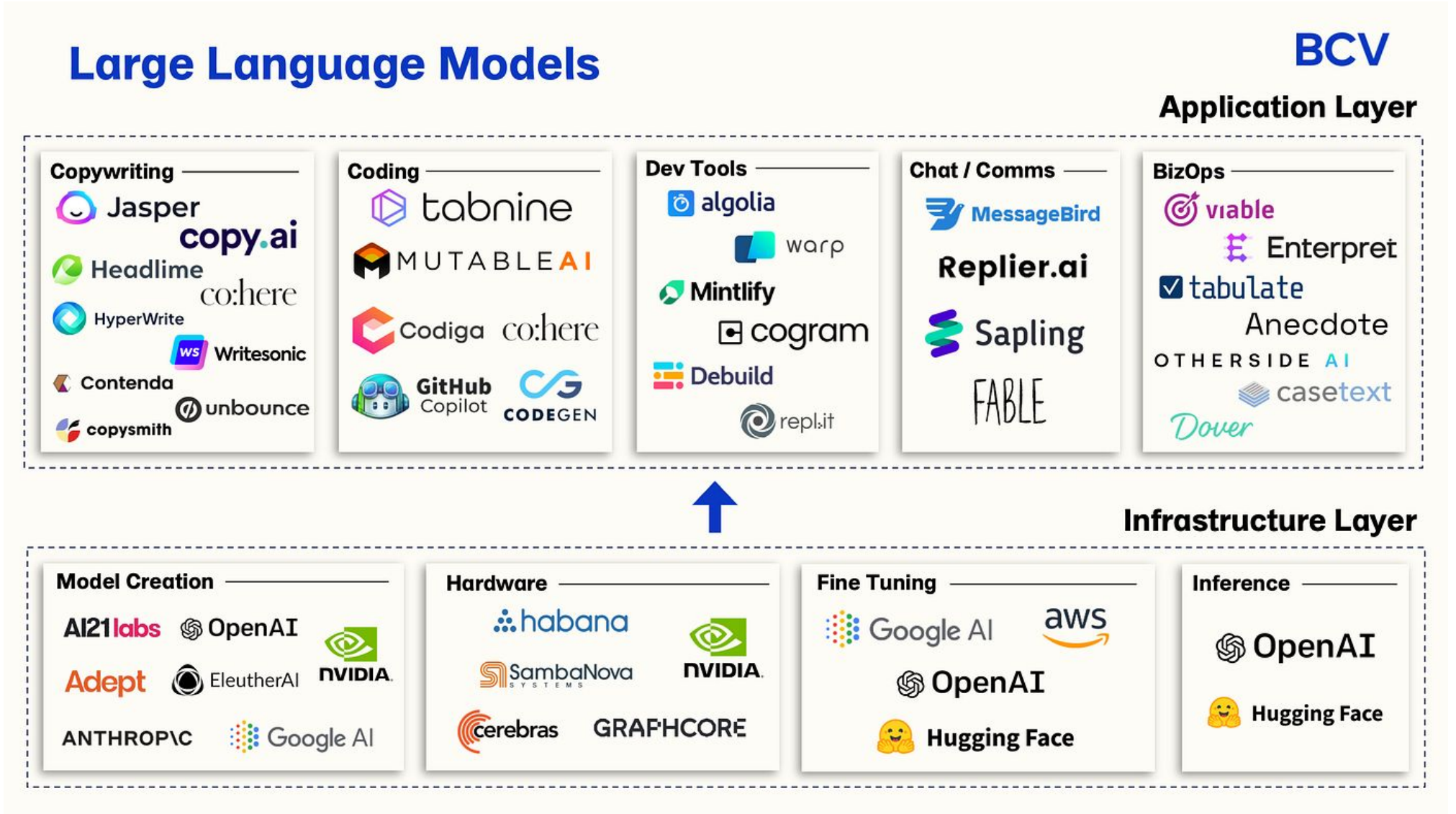
Text-to-Task (Tool GPT)

Domain d’application:
- Assistants virtuelle
- Automation
- LLM agent

Ray-BAN META

LLM: type de modèles

GNOMON DIGITAL



Qu’est-ce que c’est Prompts et Prompt Engineering

GNOMON DIGITAL

Généré par ChatGPT

Le **prompt engineering** dans le contexte des modèles de langage (LLM, pour *Large Language Models*) désigne l'art et la science de concevoir des instructions textuelles (ou *prompts*) afin d'obtenir des résultats spécifiques ou optimaux d'un modèle de langage.

Définition

Le prompt engineering consiste à rédiger, structurer ou formuler des *prompts* pour orienter le comportement d'un LLM et maximiser sa capacité à produire des réponses pertinentes, précises ou créatives. Cela peut inclure l'utilisation de formats particuliers, de questions claires, de contextes détaillés ou de contraintes explicites.

Objectifs principaux

- **Optimisation des réponses** : Obtenir des résultats qui correspondent aux attentes en termes de contenu, style ou précision.
- **Réduction des ambiguïtés** : Limiter les interprétations multiples pour guider le modèle vers une réponse unique et claire.
- **Exploration créative** : Stimuler des réponses originales en utilisant des formulations non conventionnelles.

Techniques courantes

1. **Précision des consignes** : Formuler des demandes explicites avec des détails clairs (ex. "Expliquez en trois points la théorie de la relativité").
2. **Contexte riche** : Fournir un contexte pertinent pour améliorer la compréhension (ex. "Imaginez que vous êtes un scientifique du 19e siècle...").
3. **Format structuré** : Exiger des réponses sous forme de liste, tableau ou paragraphe (ex. "Donnez un résumé sous forme de liste à puces").
4. **Chain-of-thought prompting** : Encourager le raisonnement pas à pas pour résoudre des problèmes complexes.
5. **Exemples dans le prompt** : Inclure des exemples explicites pour clarifier les attentes (ex. "Voici un exemple de réponse idéale...").

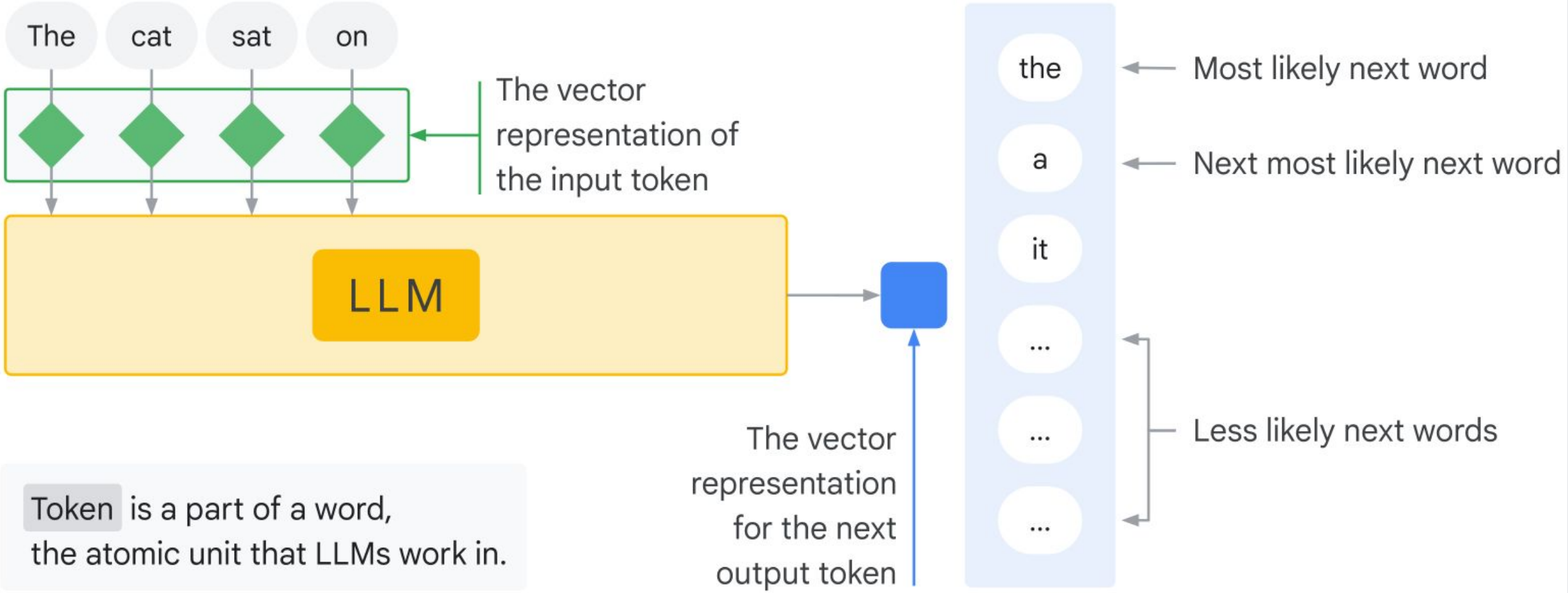
Qu’est-ce que c’est Prompts et Prompt Engineering

GNOMON DIGITAL

Il y a 3 façon d'entraîner un modèle LLM, chacun a besoin une façon de ‘prompting’ différente:

- Modèle LLM générique (ou brut)
- Optimisé par les instructions
- Optimisé par un dialogue

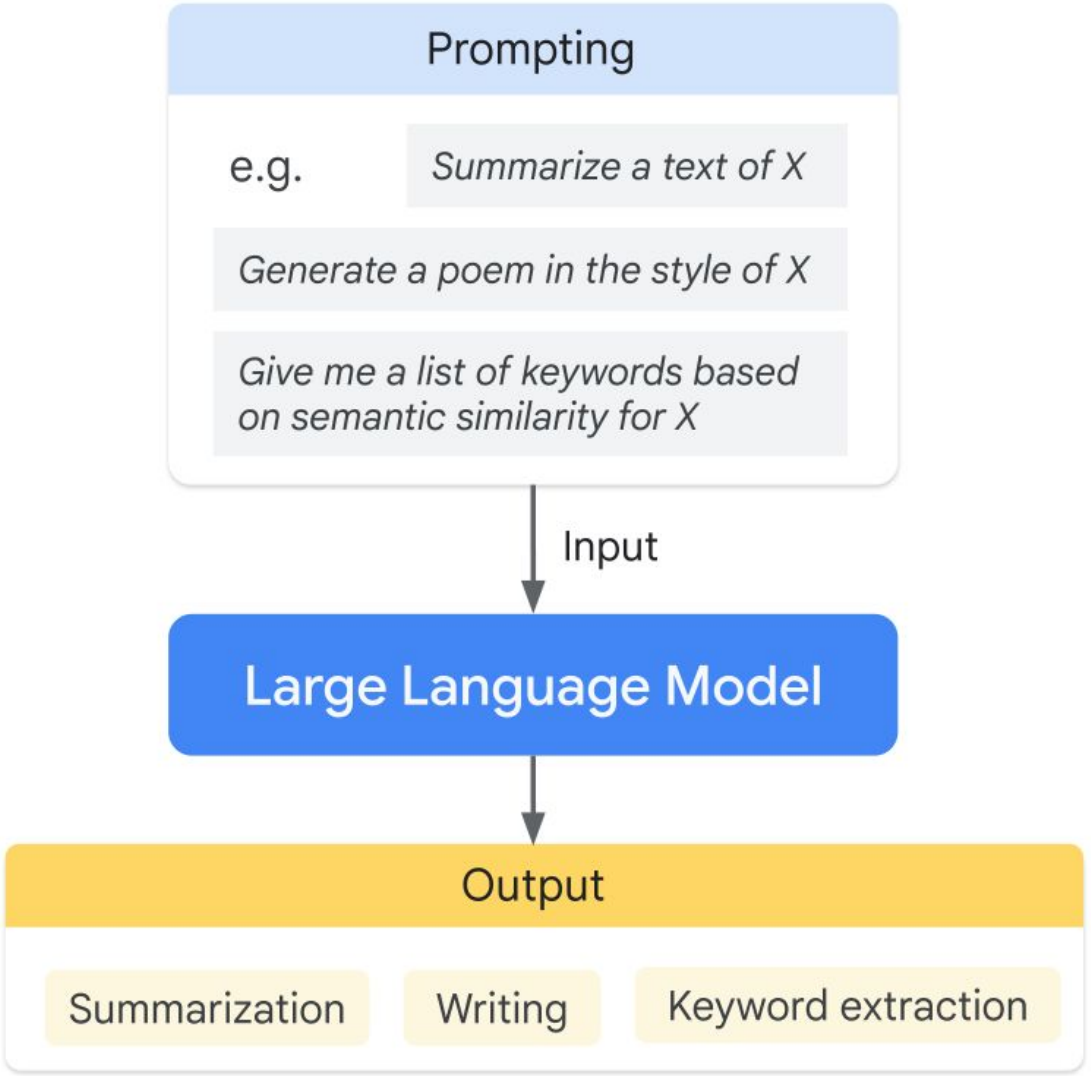
Modèle LLM générique (ou brut)



Optimisé par les instructions

GNOMON DIGITAL

Entraîner à prédire une réponse à partir des instructions données dans les texts entrées



Optimisé par une dialogue

Entraîné à engager un dialogue en prédisant la prochaine réponse.

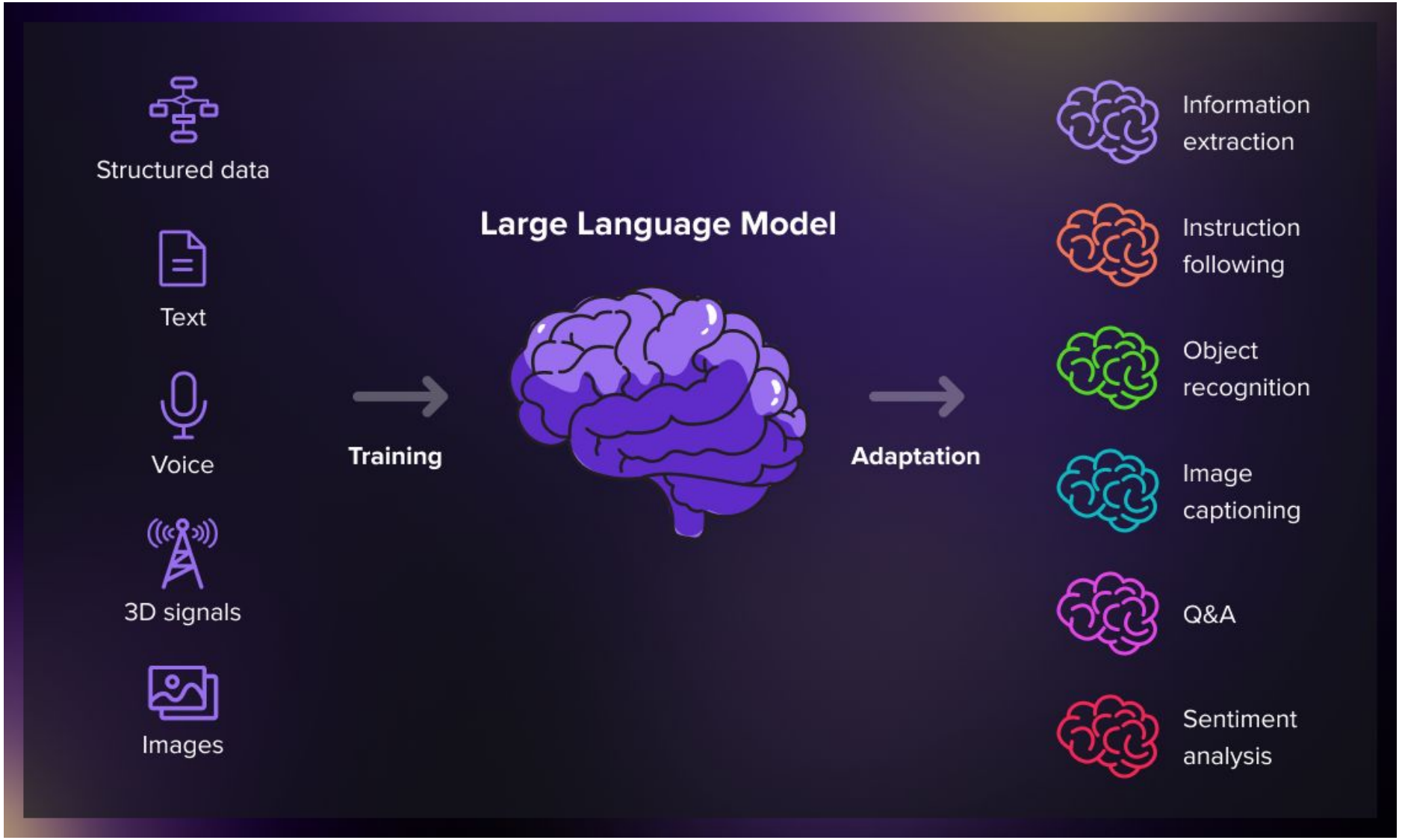
Prompt examples

```
[User] Is the comment "do you like the
weather?" ok or toxic?
[Bot] ok.
[User] can you briefly say why?
[Bot]
```

Model Output

```
It's just a question about the weather, people
are not usually upset by that.
```

Les utilisations LLM



LLM peut être utilisé dans beaucoup des contextes différentes.

Fin-tuning in LLM

GNOMON DIGITAL

Le processus d'adaptation d'un modèle à un nouveau domaine ou à un ensemble de cas d'utilisation spécifiques en entraînant le modèle sur de nouvelles données. Par exemple, nous pouvons collecter des données d'entraînement et "ajuster" le modèle LLM spécifiquement pour le domaine juridique ou médical.

Fin-tuning in LLM

GNOMON DIGITAL

Le processus d'adaptation d'un modèle à un nouveau domaine ou à un ensemble de cas d'utilisation spécifiques en entraînant le modèle sur de nouvelles données. Par exemple, nous pouvons collecter des données d'entraînement et "ajuster" le modèle LLM spécifiquement pour le domaine juridique ou médical.

Prompt Engineering

GNOMON DIGITAL

The **response** from the model

The answers you get depend on the questions you ask.

The **prompt** you designed

Prompt Engineering

GNOMON DIGITAL

On teste les cas suivants:

- Who is Linus Torvalds?
- Summarize some important dates in the life of Linus Torvalds
- Write me a tweet to celebrate Linus Torvald's birthday
- Tester toi-même un prompt

https://chatgpt.com/

Prompt Engineering

GNOMON DIGITAL

Elements of a Good Prompt

Act as a helpful tutor who breaks down complex subjects into easy explanations.

I want you to explain the process of photosynthesis to a

14 year old student, to assist with biology exam preparations.

Your answer should be 300 words, written in a tone that's friendly and educational.

Persona: Ask the tool to take a role

Objective: What do you want the AI to do

Audience: Specify who it's for

Context: What does the tool need to know

Boundaries: Set your own direction & limitation

Tip 1

Give Clear Instructions

Use commands that instruct the AI tool on what you want to generate, such as 'explain', 'translate', 'summarize' or 'compare'.

Tip 2

Provide Context

Adding context and background information can help the tool to understand the task better. For example, mention the project type such as 'short story', 'report' or 'outline

Tip 3

Iterate & Experiment

Try different instructions and techniques if you don't get the results you want. Prompting can be like an experiment that may require several rounds of iterations!

https://www.promptingguide.ai/

Prompt Engineering

GNOMON DIGITAL

Ambiguïté et Nuance:

- **Prompt A (Ambigu):** "Écris une histoire."
- **Prompt B (Précis):** "Écris une histoire courte de science-fiction sur un robot qui découvre qu'il est capable de ressentir des émotions, en utilisant un vocabulaire riche et des descriptions imagées."

Instructions Complexes

- **Prompt A (Simple):** "Traduis 'Hello, how are you?' en français."
- **Prompt B (Complexe):** "Traduis 'Hello, how are you?' en français, puis réécris la traduction en utilisant un langage formel, puis informel, et enfin en argot parisien. Indique pour chaque version le contexte approprié."

Contraintes Créatives

- **Prompt 5 (Libre):** "Écris un poème."
- **Prompt 6 (Contraint):** "Écrit un sonnet en alexandrins sur le thème de la solitude, en utilisant au moins trois fois le mot 'silence'."

Raisonnement et Connaissance

- **Prompt A (Factuel):** "Qui a peint la Joconde?"
- **Prompt B (Raisonnement):** "Si tous les chats sont des mammifères, et que Minou est un chat, est-ce que Minou est un mammifère ? Explique ton raisonnement."

https://www.promptingguide.ai/

Prompt Engineering

GNOMON DIGITAL

Sensibilité au contexte

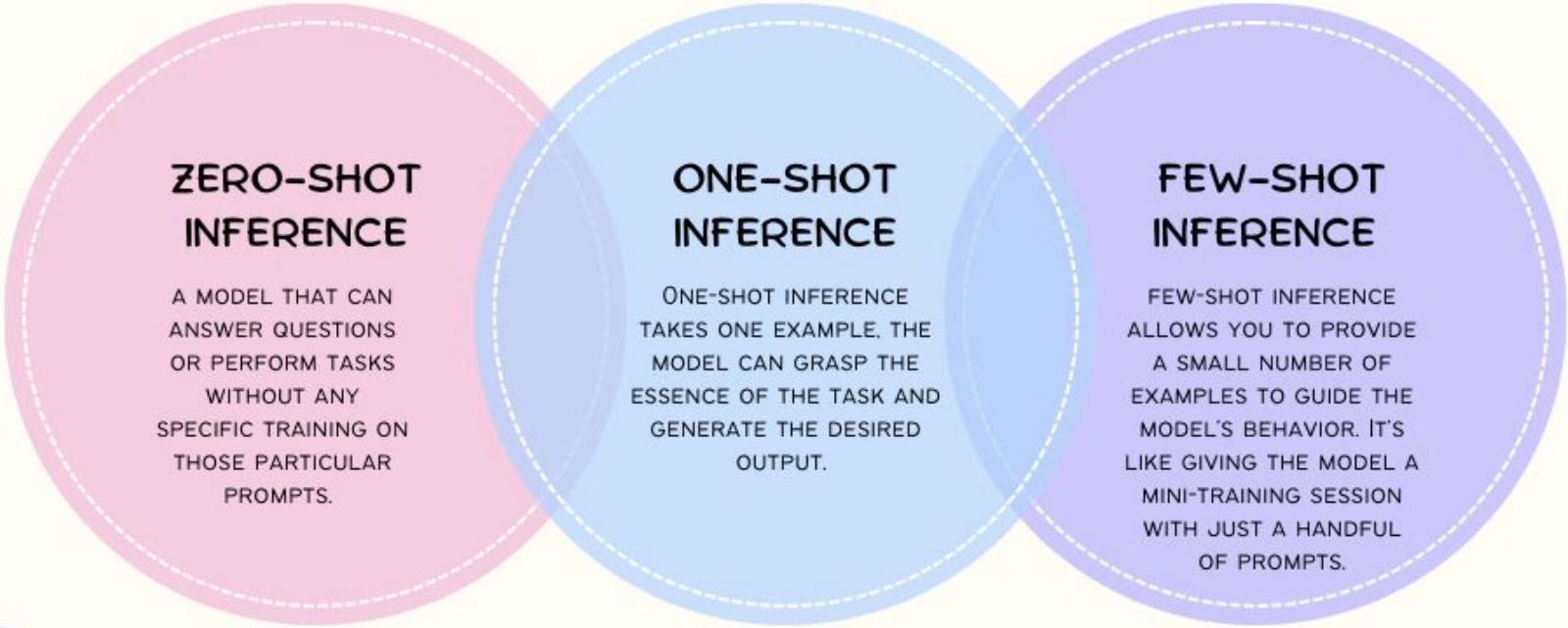
- **Prompt 9 (Sans contexte):** "Explique le concept de la relativité."
- **Prompt 10 (Avec contexte):** "Imagine que tu expliques le concept de la relativité à un enfant de 8 ans. Utilise des analogies simples et évite le jargon scientifique."

Tips:

- By nice
- Step by step
- Ask LLM to write your prompt

https://www.promptingguide.ai/

POWER OF PROMPT ENGINEERING



GNOMON® DIGITAL

https://www.promptingguide.ai/

Zero-shot prompting

GNOMON DIGITAL

On donne des instructions à LLM sans aucun exemples

EX:

Quelle est la sentiment de la phrase suivant: ‘I am very happy today’? Positive, négative ou neutre.

https://www.promptingguide.ai/

One-shot prompting

On donne des instructions à LLM avec un exemple

EX:

Quelle est la sentiment de la phrase suivant: ‘I am very happy today’? Positive, négative ou neutre.
Example:
Input: “The book is very interesting”
Output: “Positive”

https://www.promptingguide.ai/

Prompt Enginneering

GNOMON®
DIGITAL

On teste?

- Gemini: https://aistudio.google.com/prompts/new_chat
- ChatGPT: https://chatgpt.com/

https://www.promptingguide.ai/

Few-shot prompting

GNOMON DIGITAL

On donne des instructions à LLM avec un exemple

EX:

Quelle est la sentiment de la phrase suivant: ‘I am very happy today’? Positive, négative ou neutre.
Example:
Input: “The book is very interesting”
Output: “Positive”
Input: “I feel so depressed today”
Output: “Négative”
Input: “My name is John”
Output: “Neutre”

https://www.promptingguide.ai/

Few-shot prompting

GNOMON DIGITAL

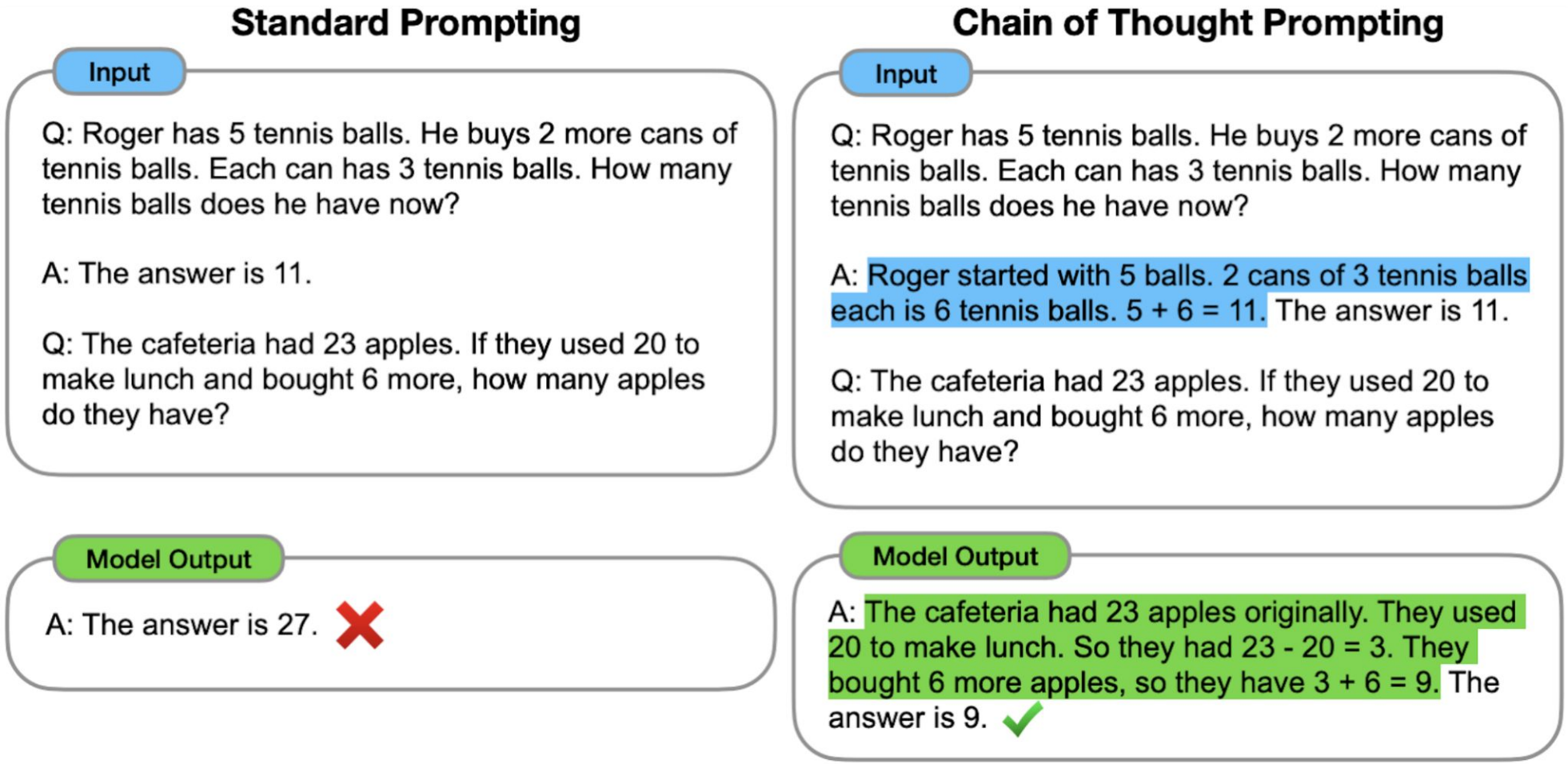
https://github.com/GoogleCloudPlatform/asl-ml-immersion/blob/master/notebooks/vertex_genai/labs/prompt_engineering.ipynb

https://github.com/GoogleCloudPlatform/asl-ml-immersion/blob/master/notebooks/vertex_genai/labs/gemini_for_multimodal_prompting.ipynb

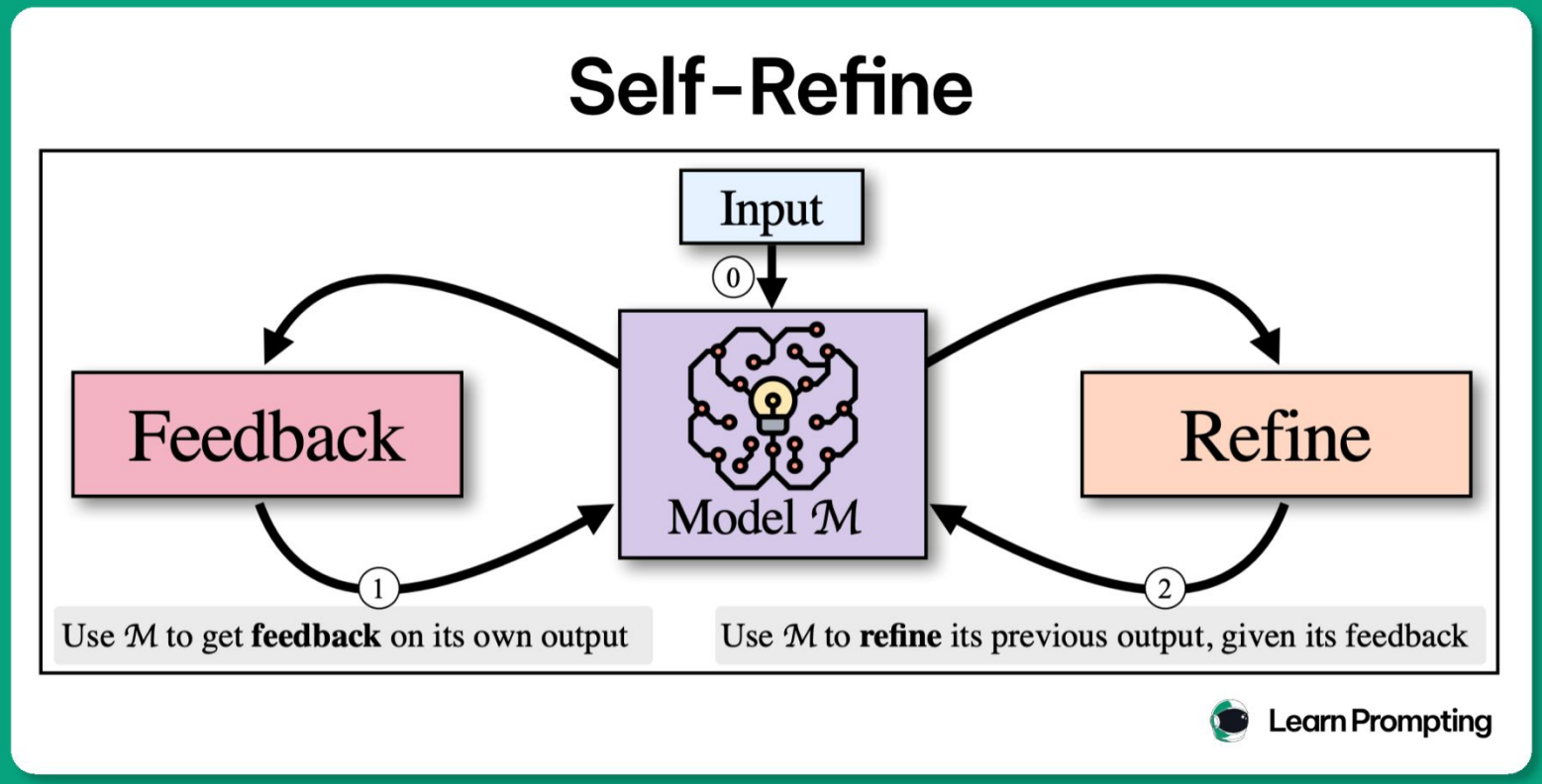
Chain of Thought

GNOMON DIGITAL

Magique prompt: “Let’s think this through step by step”



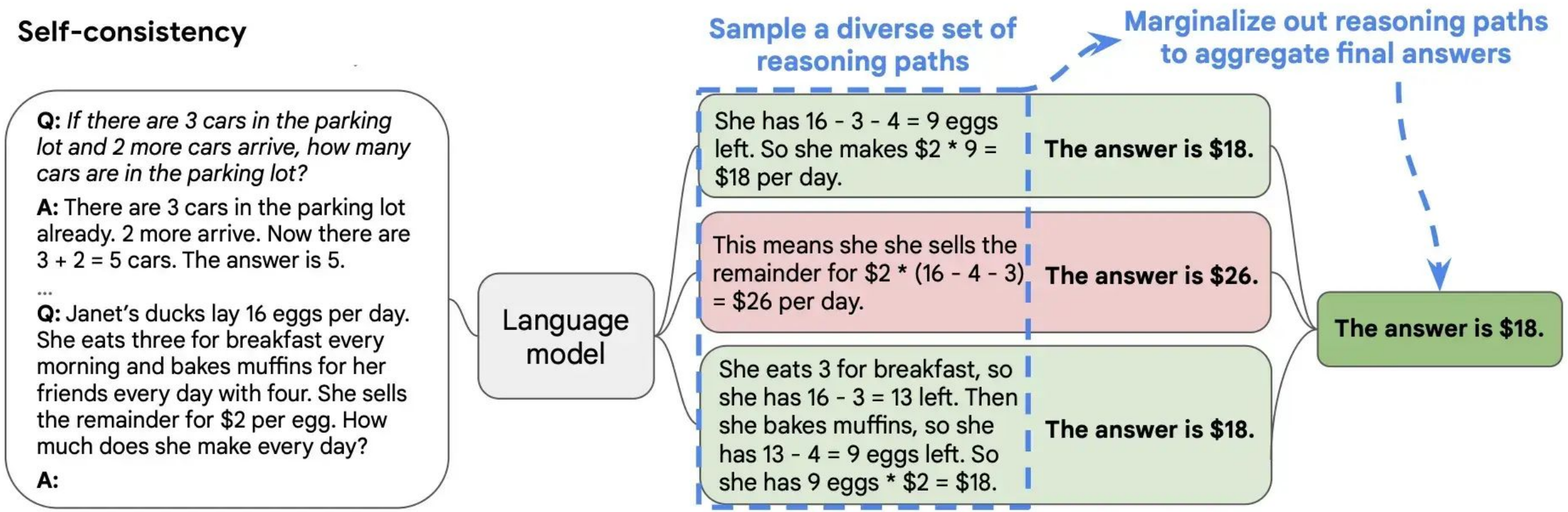
Self Refine



SELF-REFINE:
Iterative Refinement with Self-Feedback

Aman Madaan[1], Niket Tandon[2], Prakhar Gupta[1], Skyler Hallinan[3], Luyu Gao[1], Sarah Wiegreffe[2], Uri Alon[1], Nouha Dziri[2], Shrimai Prabhumoye[4], Yiming Yang[1], Shashank Gupta[2], Bodhisattwa Prasad Majumder[5], Katherine Hermann[6], Sean Welleck[2,3], Amir Yazdanbakhsh[6], Peter Clark[2]
[1]Language Technologies Institute, Carnegie Mellon University
[2]Allen Institute for Artificial Intelligence
[3]University of Washington [4]NVIDIA [5]UC San Diego [6]Google Research, Brain Team
amadaan@cs.cmu.edu, nikett@allenai.org

Self Consistency



Self Consistency

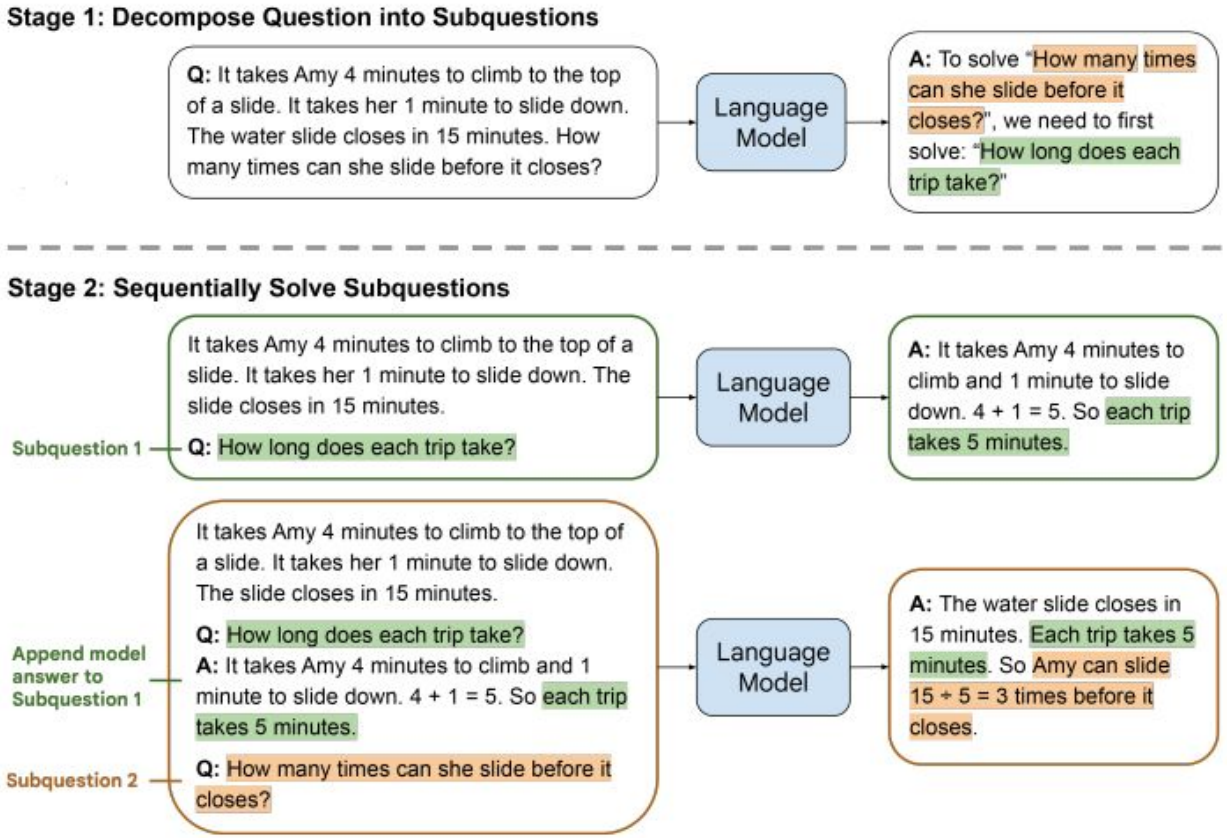


Figure 1: Least-to-most prompting solving a math word problem in two stages: (1) query the language model to decompose the problem into subproblems; (2) query the language model to sequentially solve the subproblems. The answer to the second subproblem is built on the answer to the first subproblem. The demonstration examples for each stage's prompt are omitted in this illustration.

Tree of Thoughts

GNOMON DIGITAL

Tree of Thoughts: Deliberate Problem Solving with Large Language Models

Shunyu Yao Princeton University
Dian Yu Google DeepMind
Jeffrey Zhao Google DeepMind
Izhak Shafran Google DeepMind
Thomas L. Griffiths Princeton University
Yuan Cao Google DeepMind
Karthik Narasimhan Princeton University

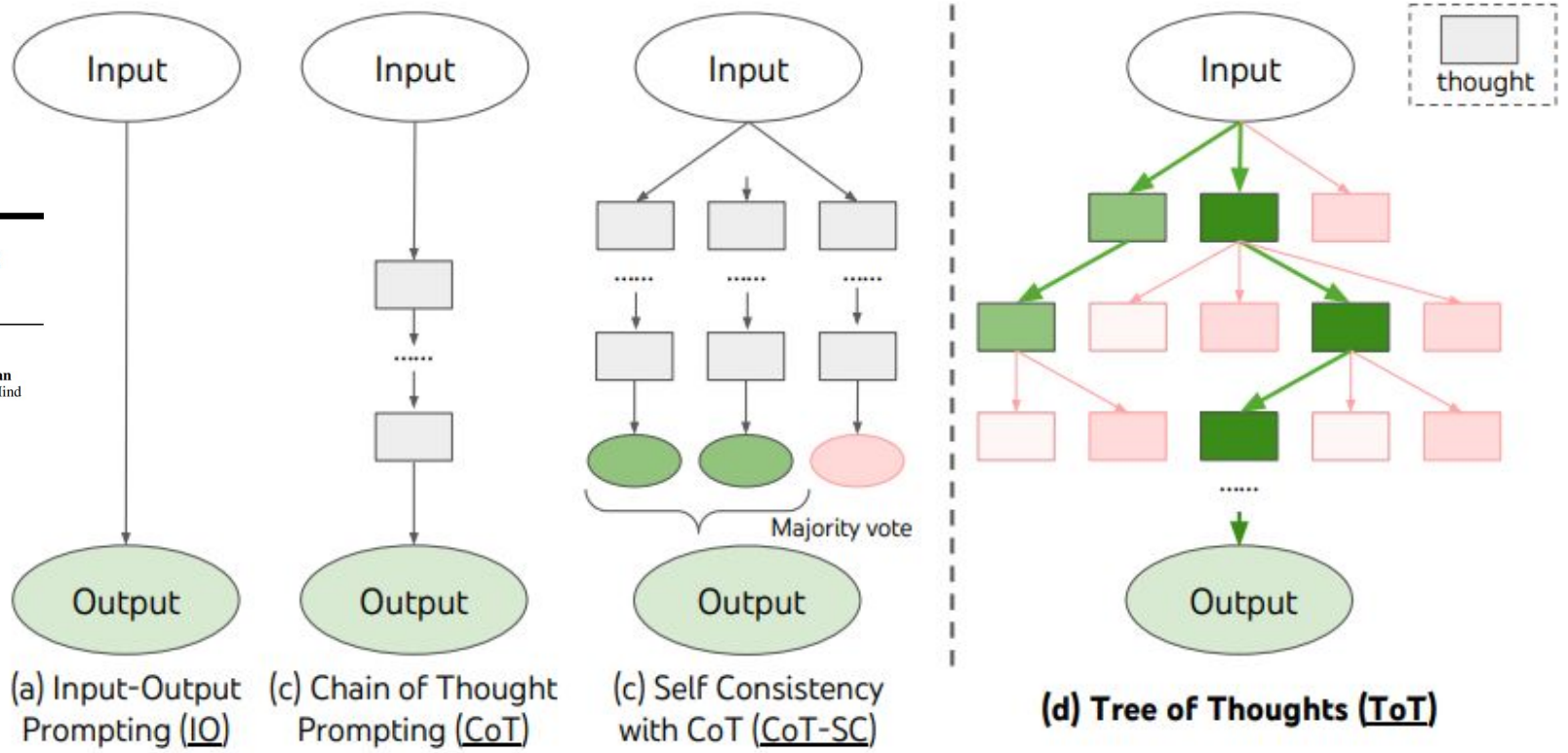
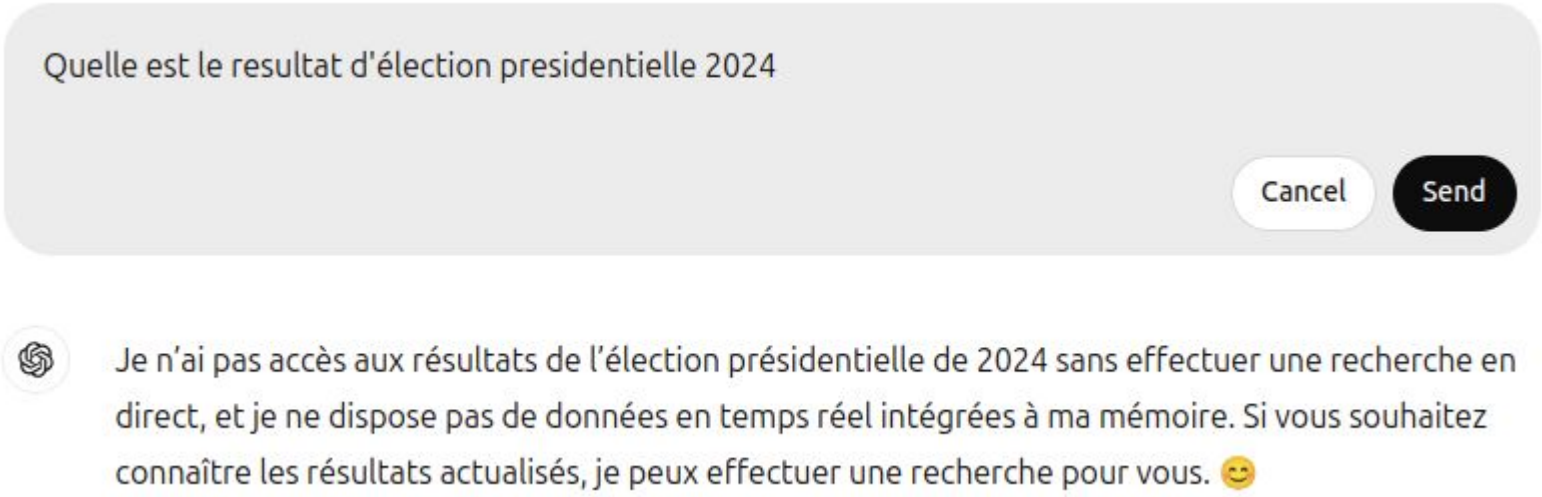


Figure 1: Schematic illustrating various approaches to problem solving with LLMs. Each rectangle box represents a *thought*, which is a coherent language sequence that serves as an intermediate step toward problem solving. See concrete examples of how thoughts are generated, evaluated, and searched in Figures 2,4,6.

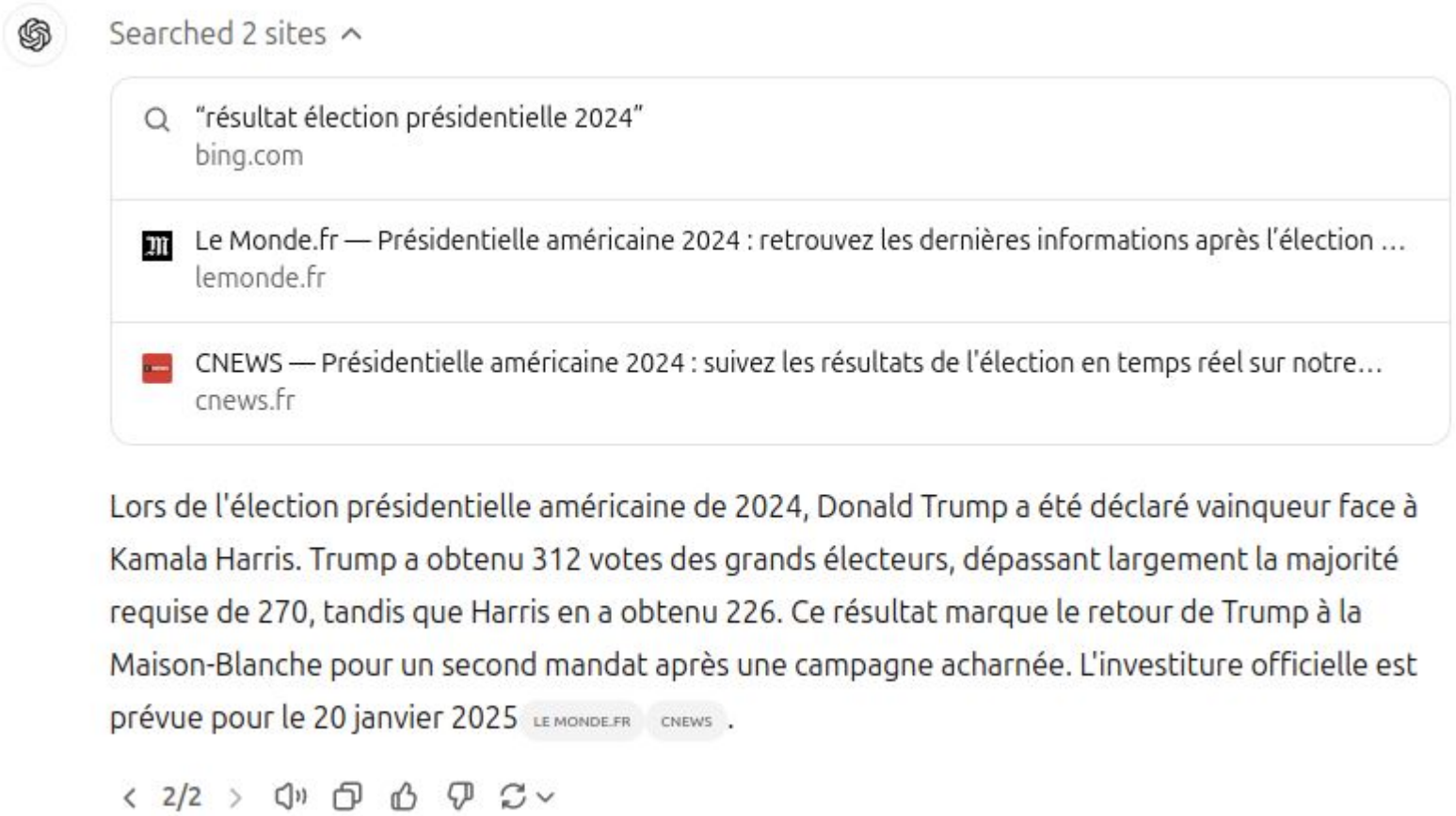
RAG

La limite de LLM:

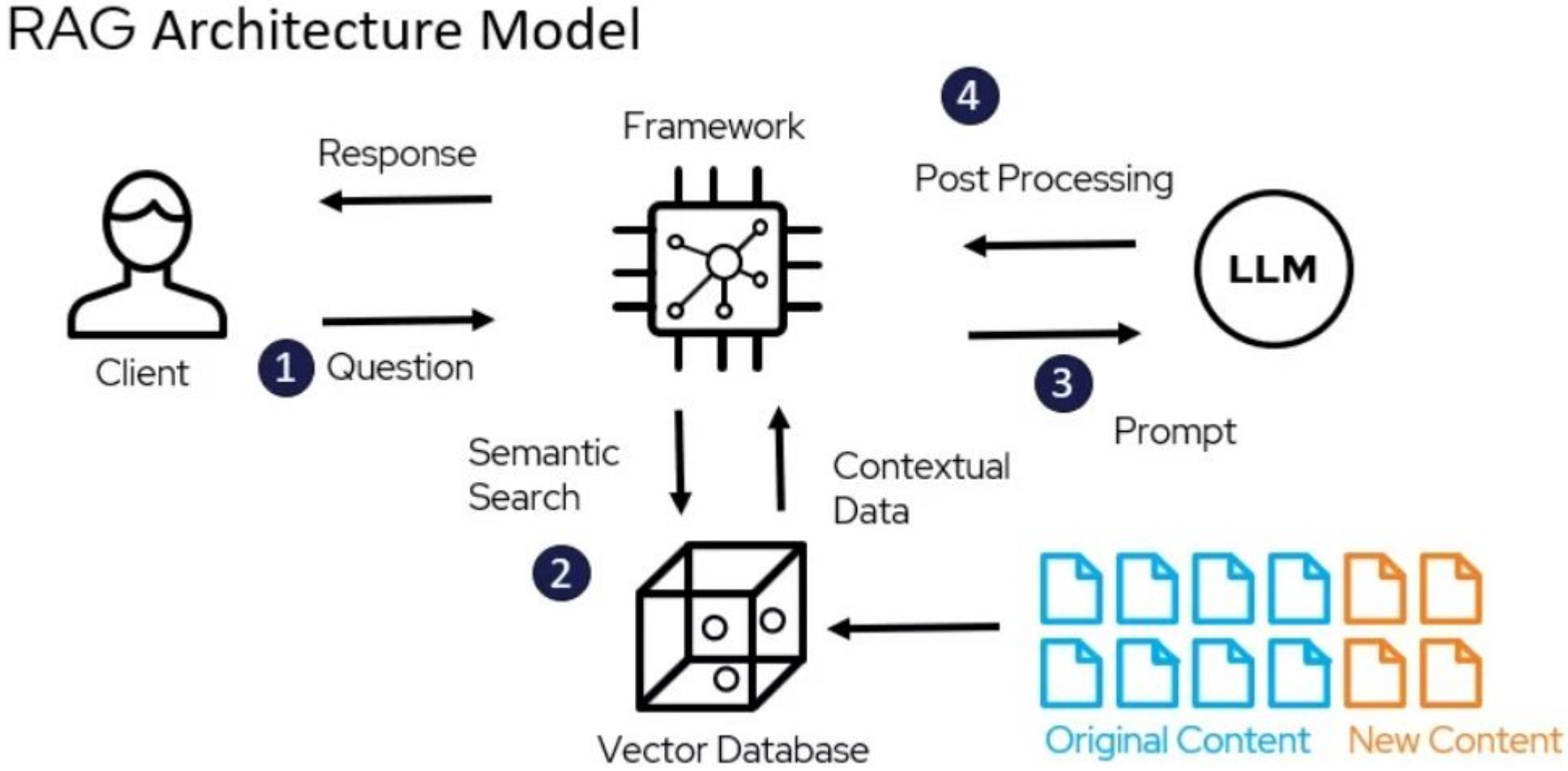
- Pas d’accès aux informations à jour



RAG


Searched 2 sites
"résultat élection présidentielle 2024"
bing.com
Le Monde.fr — Présidentielle américaine 2024 : retrouvez les dernières informations après l'élection …
lemonde.fr
CNEWS — Présidentielle américaine 2024 : suivez les résultats de l'élection en temps réel sur notre…
cnews.fr
Lors de l'élection présidentielle américaine de 2024, Donald Trump a été déclaré vainqueur face à Kamala Harris. Trump a obtenu 312 votes des grands électeurs, dépassant largement la majorité requise de 270, tandis que Harris en a obtenu 226. Ce résultat marque le retour de Trump à la Maison-Blanche pour un second mandat après une campagne acharnée. L'investiture officielle est prévue pour le 20 janvier 2025 LE MONDE.FR CNEWS .
2/2

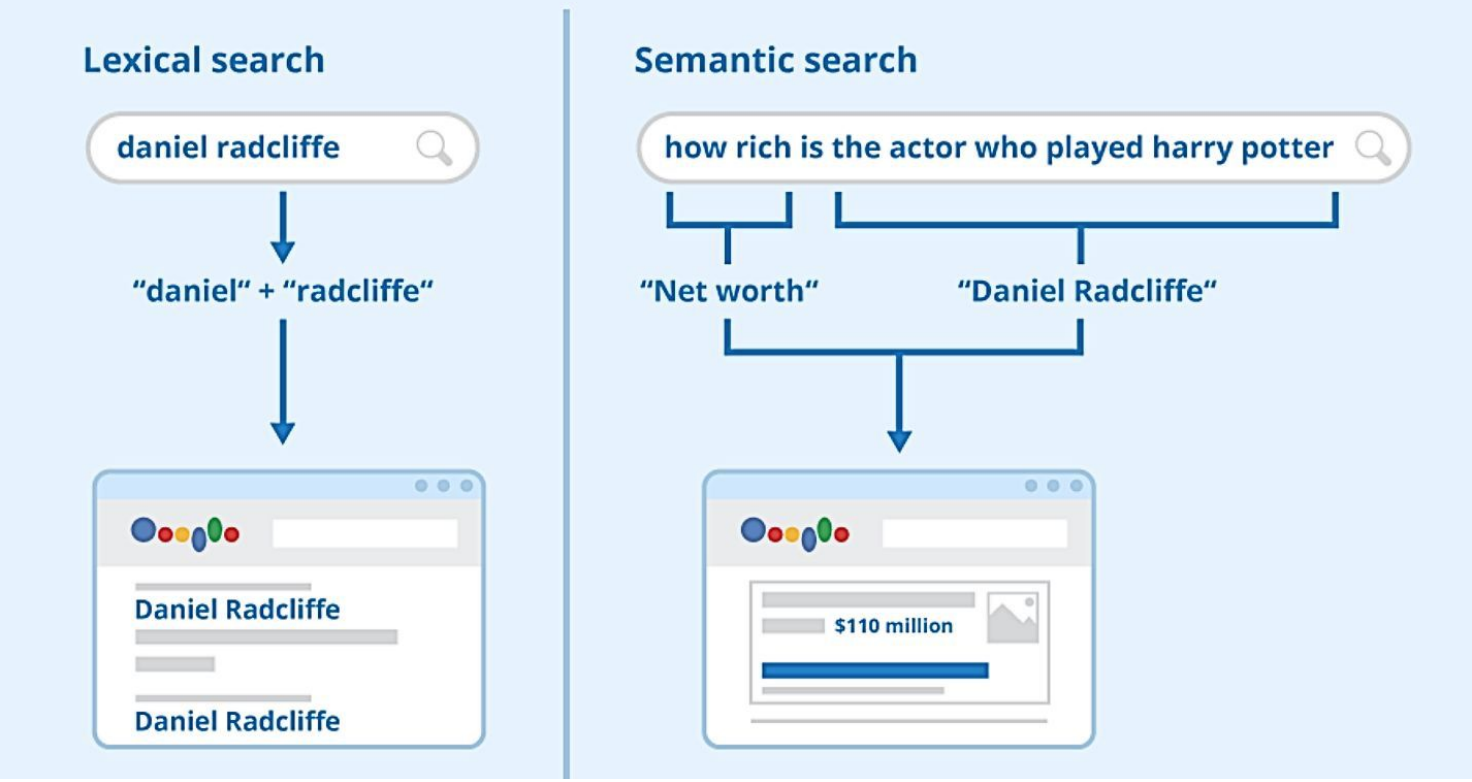
RAG


RAG Architecture Model
Framework
4
Post Processing
Response
LLM
Client
1
Question
3
Prompt
Semantic
Search
Contextual
Data
2
Vector Database
Original Content
New Content

RAG: rechercher semantic

La notion d'adéquation sémantique entre deux entités X et Y s'applique à de nombreux contextes :

- Le sens de Y est-il similaire à celui de X ?
- X et Y représentent-ils la même chose présentée différemment ?
- Y est-il une bonne recommandation ou une réponse appropriée pour X ?

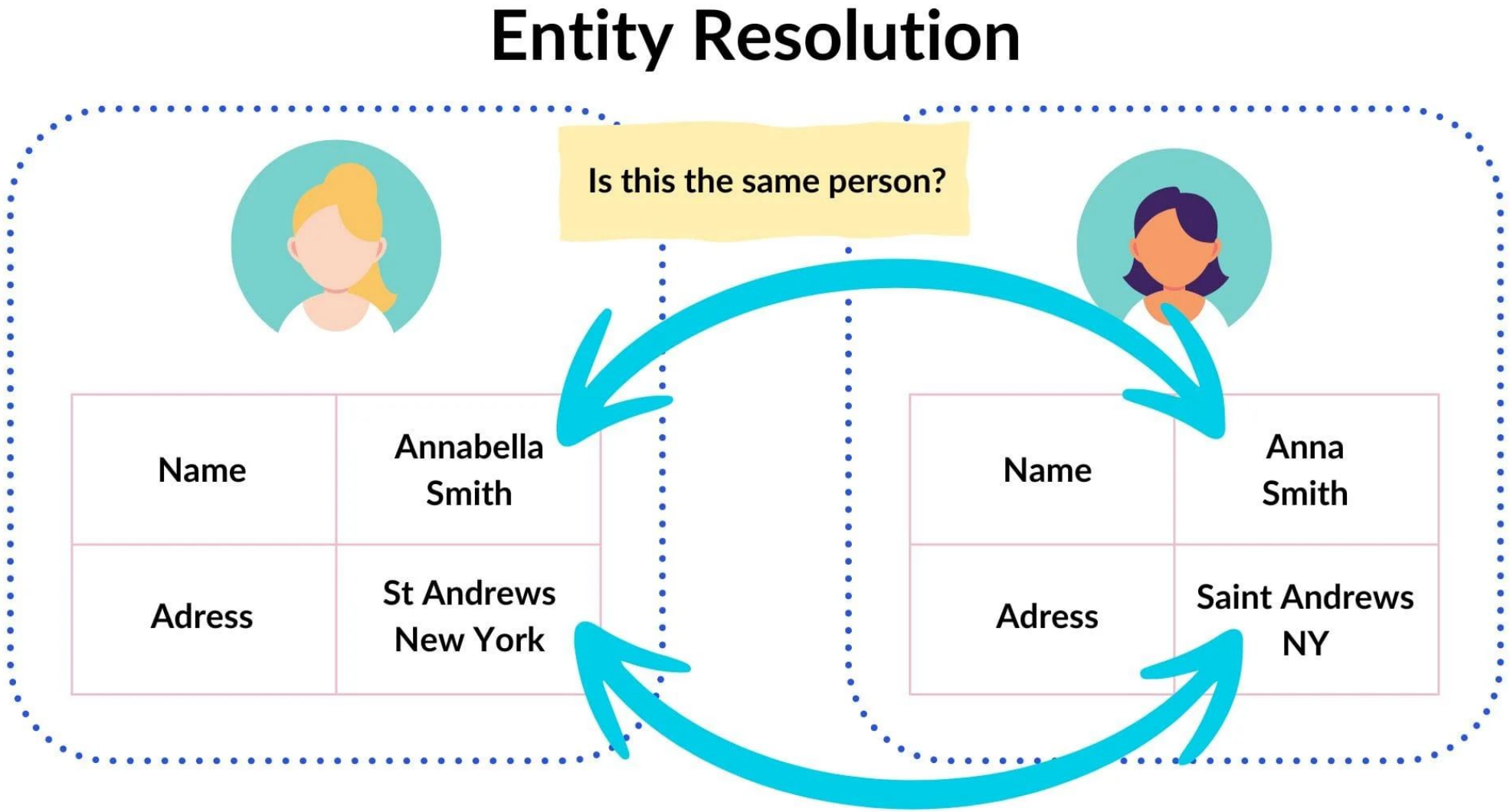


RAG: rechercher semantic

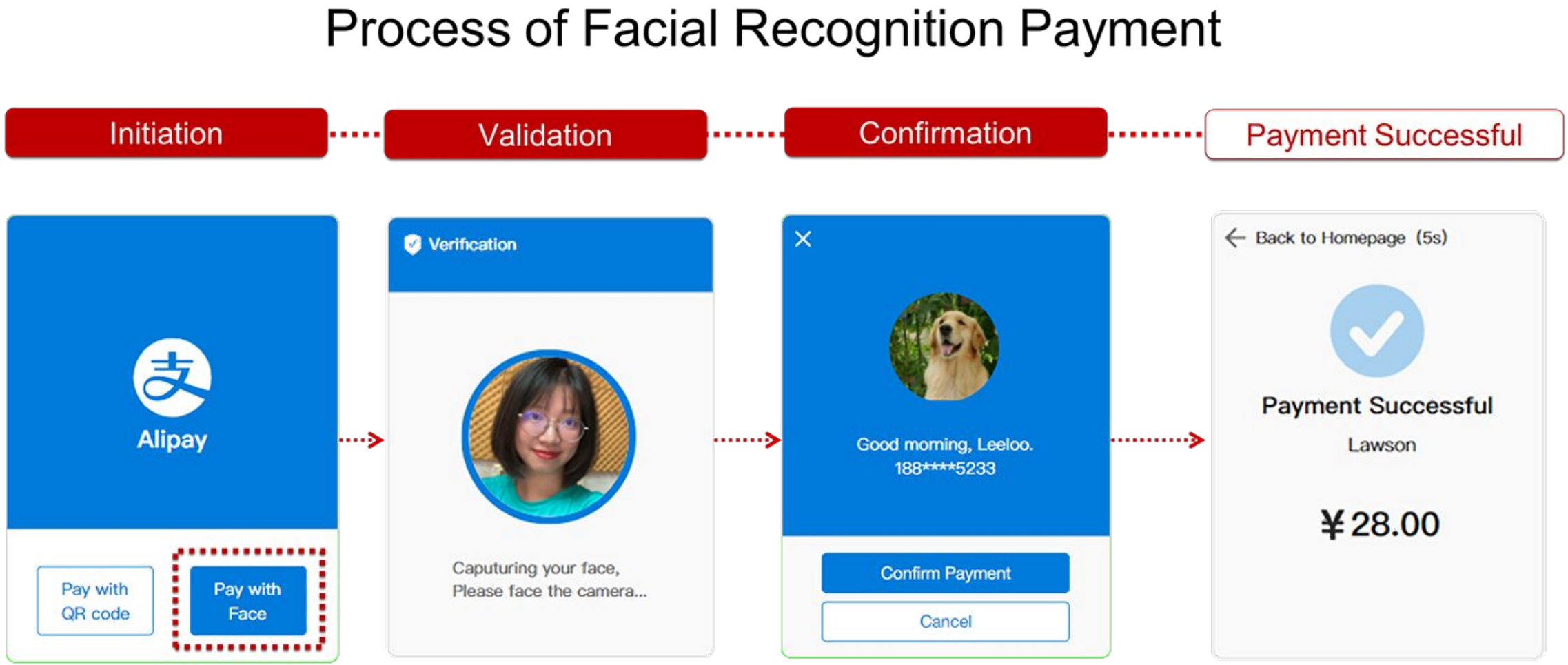
GNOMON DIGITAL

	Lexical Seach	Semantic Seach
Definition	Searches based on exact keyword match	Understands query intent and context
Matching Criteria	Matches keywords or phrases literally	Considers synonyms, context, and relationships between words
Natural Language Proccessing	Requires minimal NLP capabilities	Requires advanced NLP for understanding context
Example	Searching for "Amazon river" returns info about amazon river, amazon.com etc.	Searching for "Amazon river" returns information about the Amazon river and not the company Amazon.

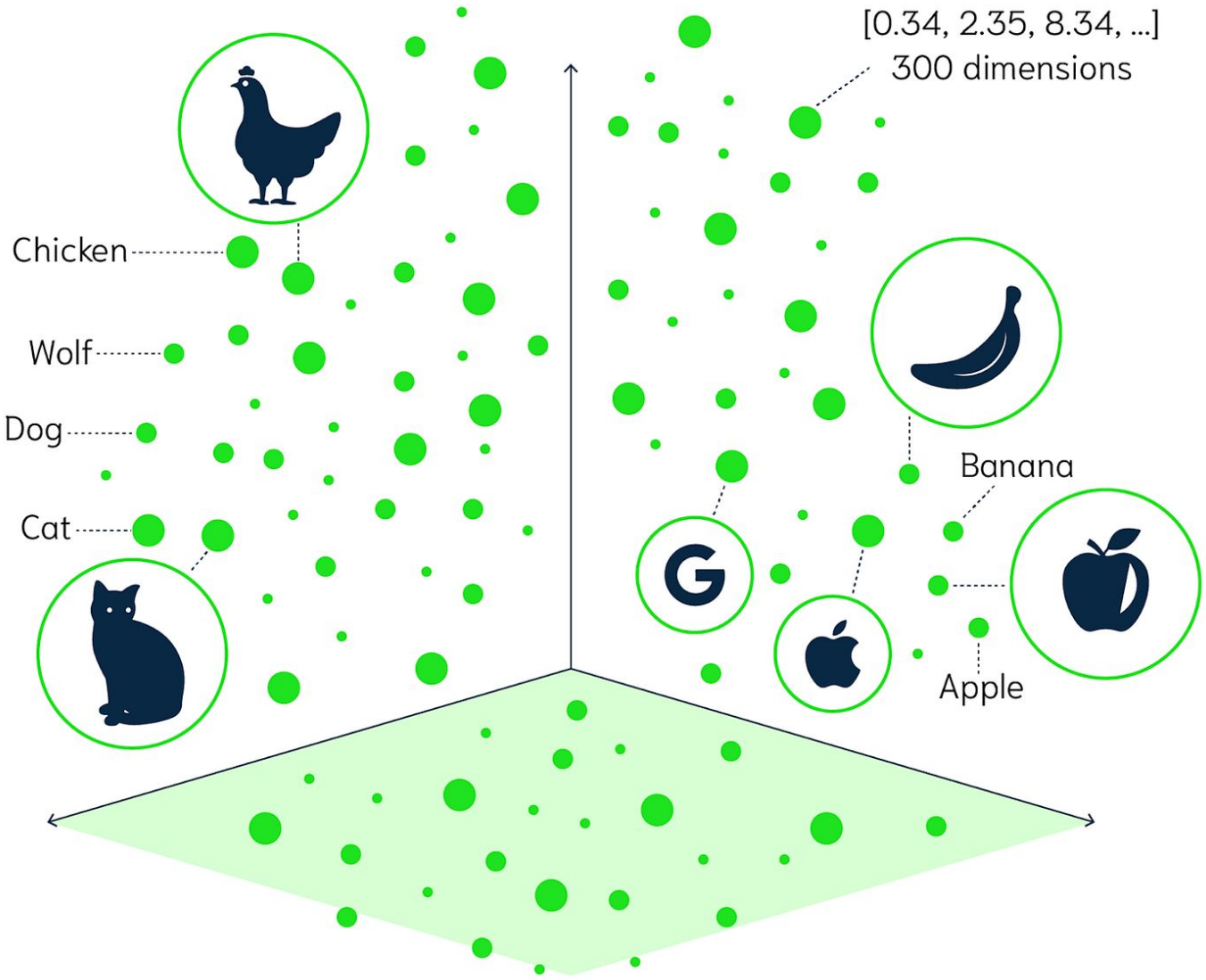
RAG: rechercher semantic

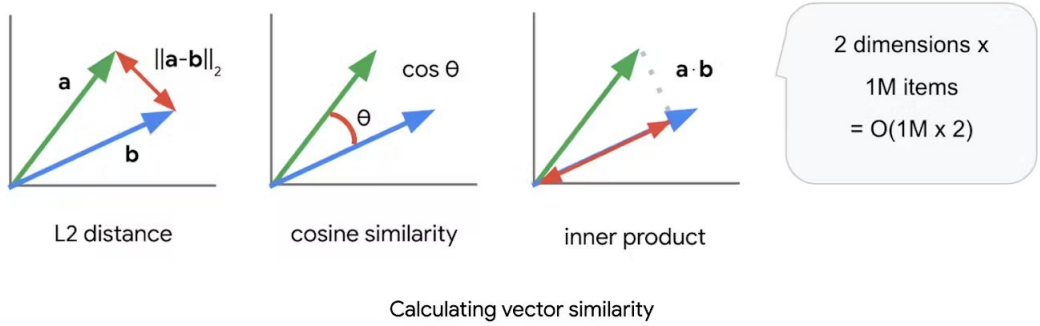


RAG: rechercher semantic



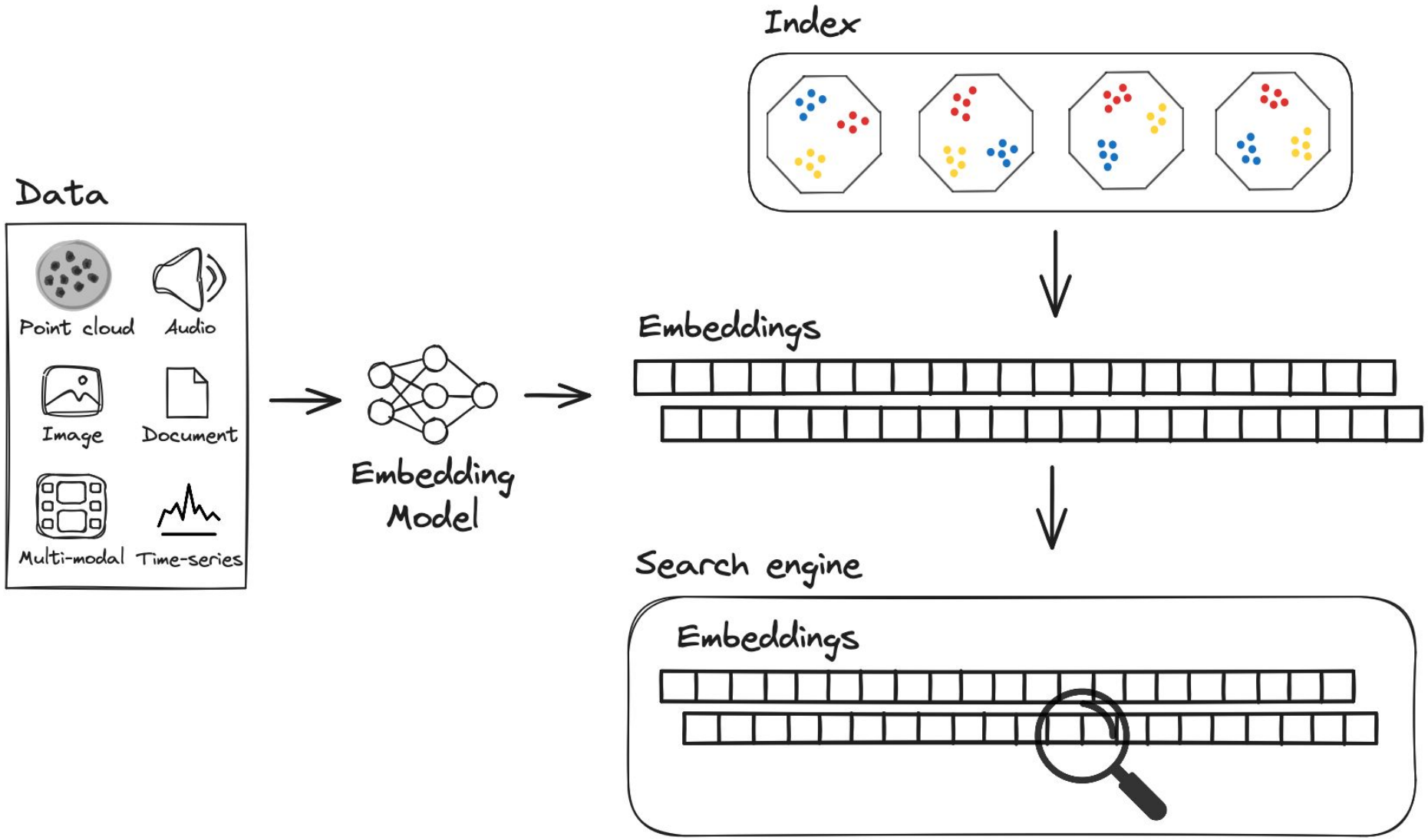
RAG: rechercher semantic



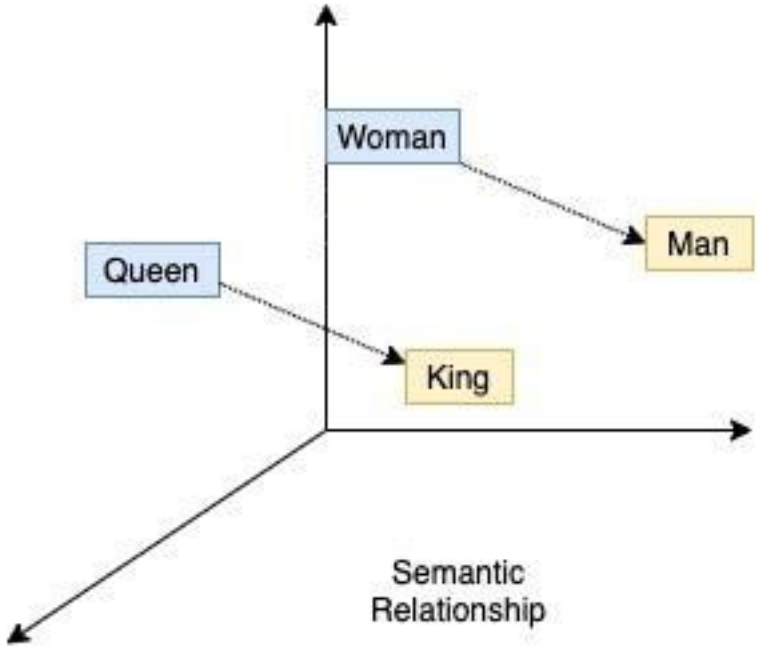


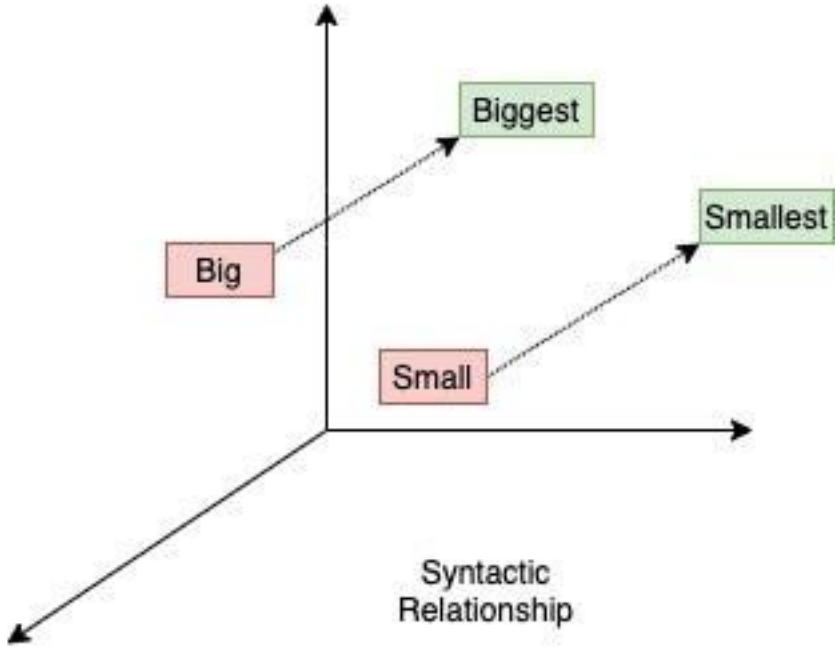
Calculating vector similarity

RAG: rechercher semantic



NLP

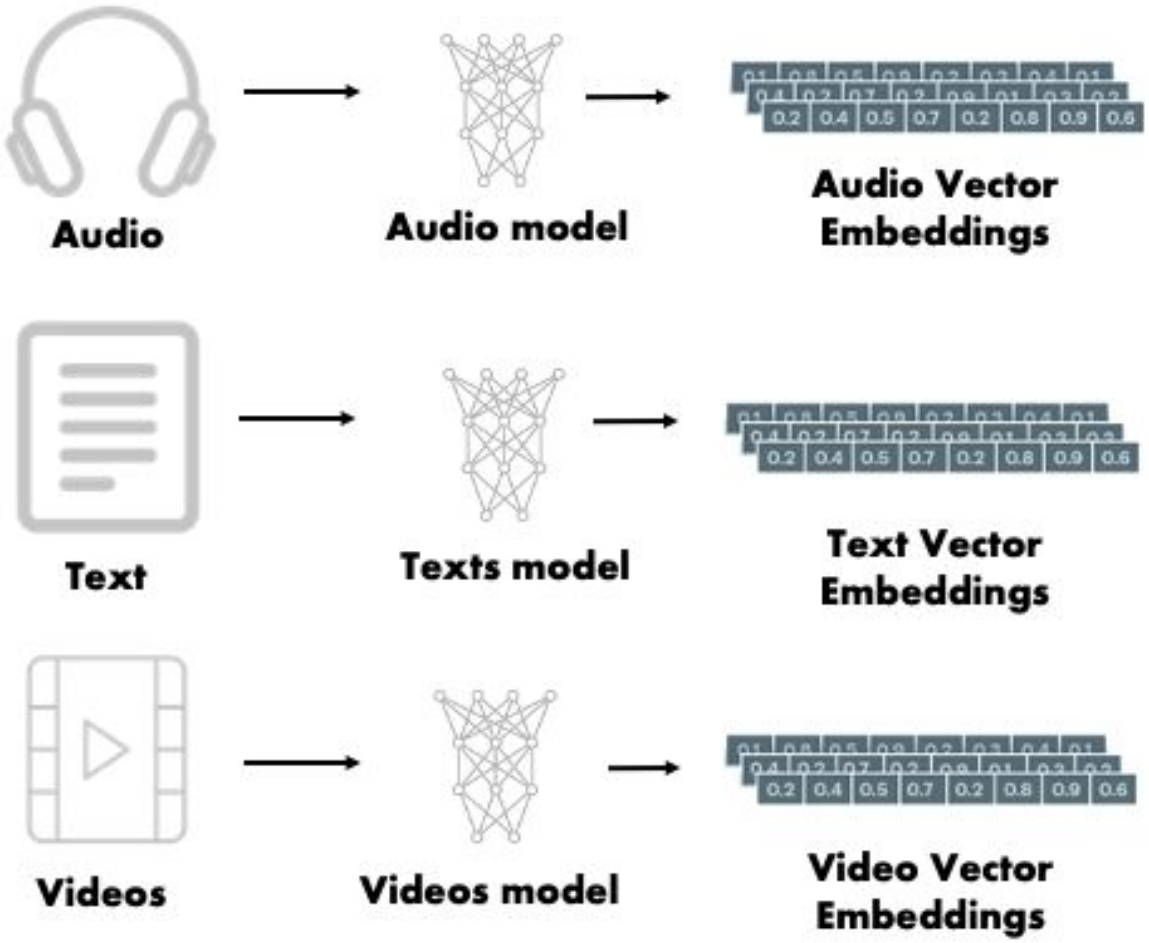




word2vec

https://medium.com/@fraidoonomarzai99/word2vec-cbow-skip-gram-in-depth-88d9cc340a50

RAG: rechercher semantic



Very similar to word2vec

RAG: rechercher semantic

GNOMON DIGITAL

Step 1: Tokenize

Tokenizers
Documentation

Hellò how are U tdày?

↓

hello how are u tday?

↓

[hello, how, are, u, tday, ?]

↓

[hello, how, are, u, td, ##ay, ?]

↓

[CLS, hello, how, are, u, td, ##ay, ?, SEP]

Normalization

Pre-tokenization

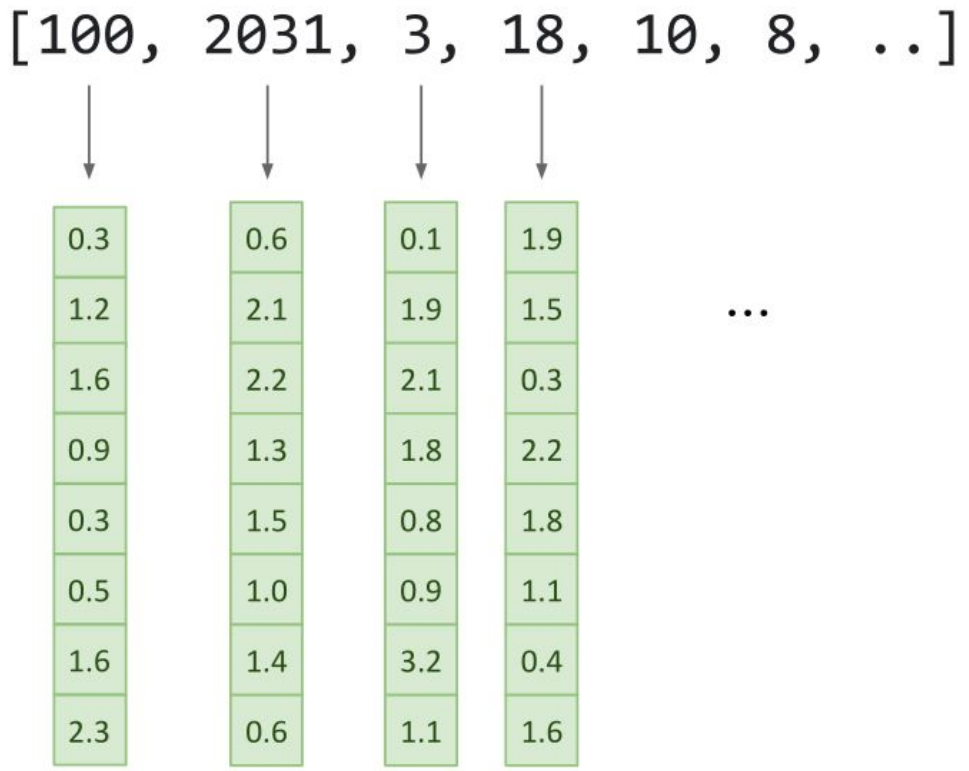
Model

Postprocessor

RAG: rechercher semantic

GNOMON DIGITAL

Step 2: Learn embedding vectors



RAG: rechercher semantic

Step 2: Learn embedding vectors

Different types of embeddings

Type 1

Non-Contextual Embeddings

Factorization Based

- Word2Vec
- Gloves
- Swivel

Type 2

Contextual Embeddings

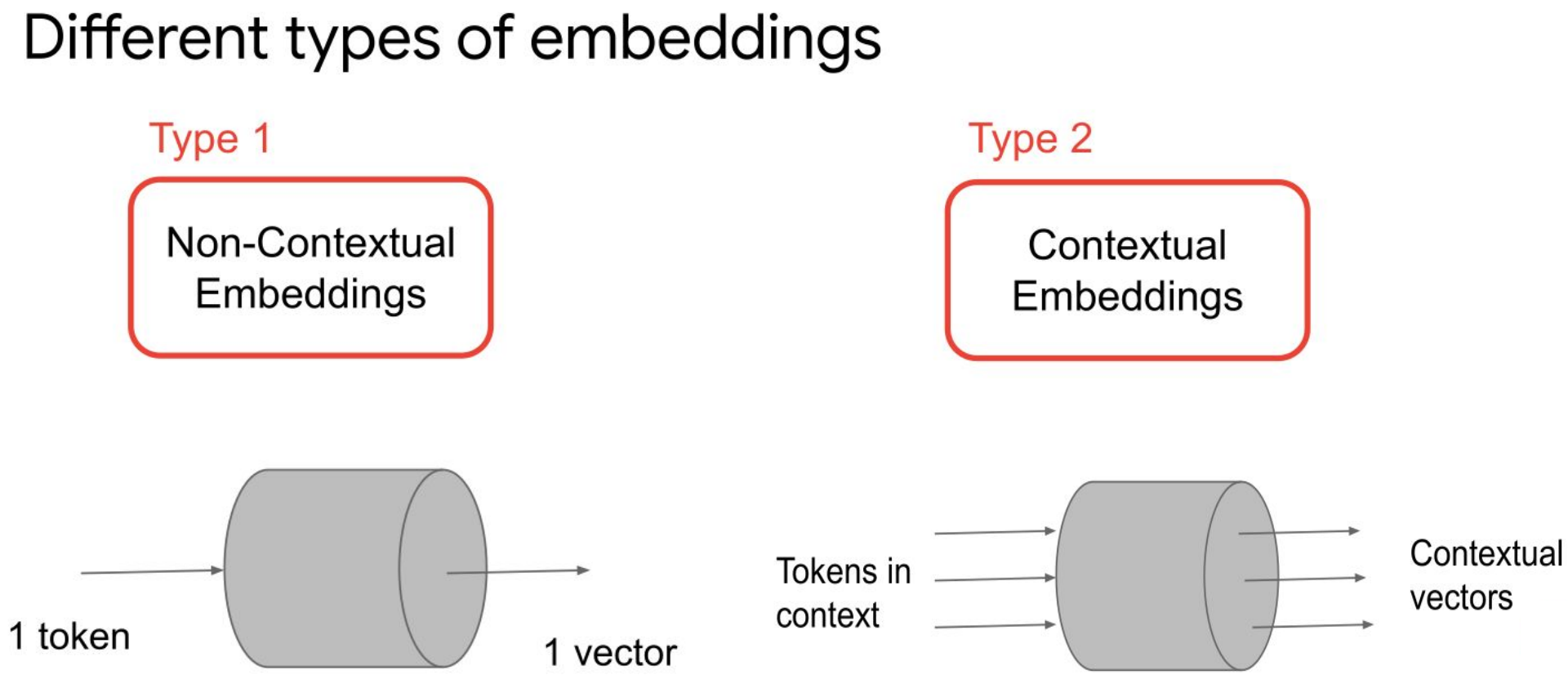
Language Model Based

- Bert
- PaLM
- GPT

RAG: rechercher semantic

GNOMON DIGITAL

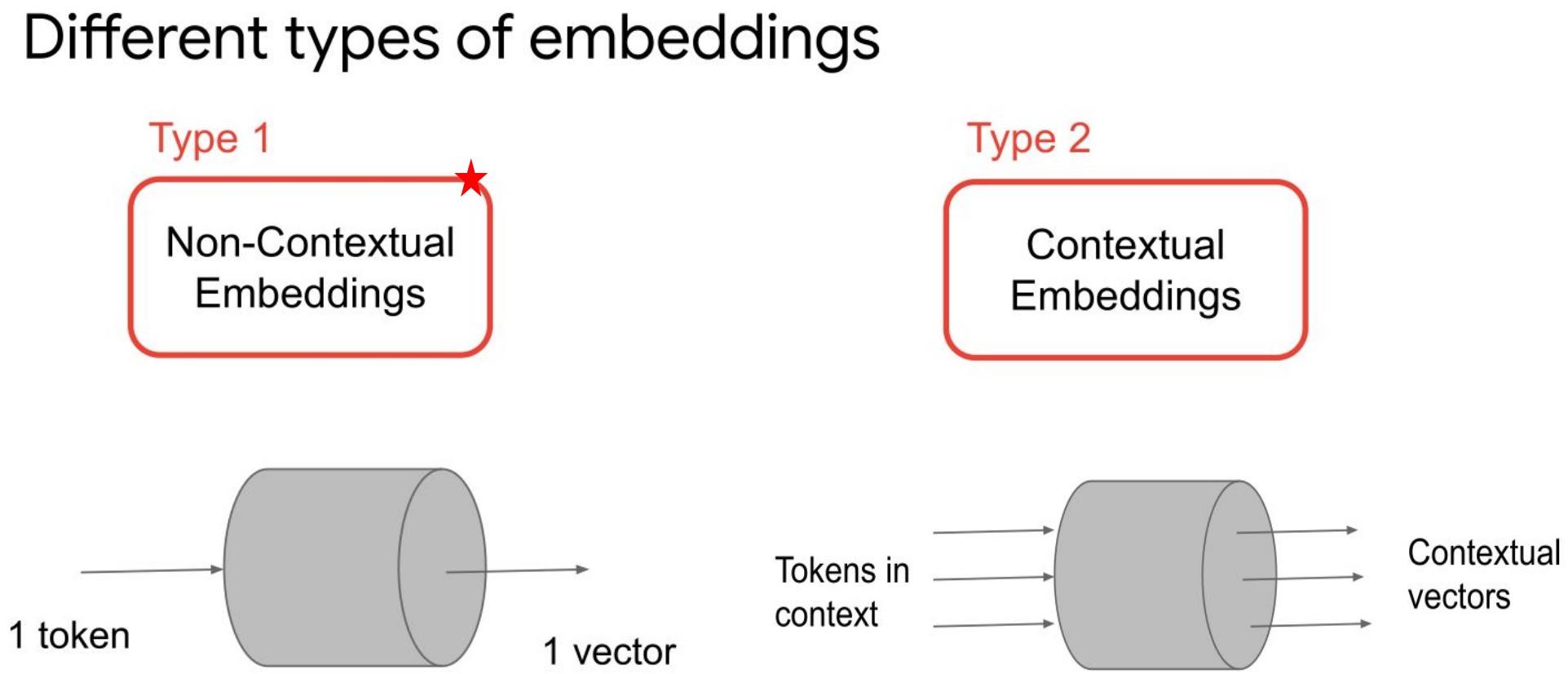
Step 2: Learn embedding vectors



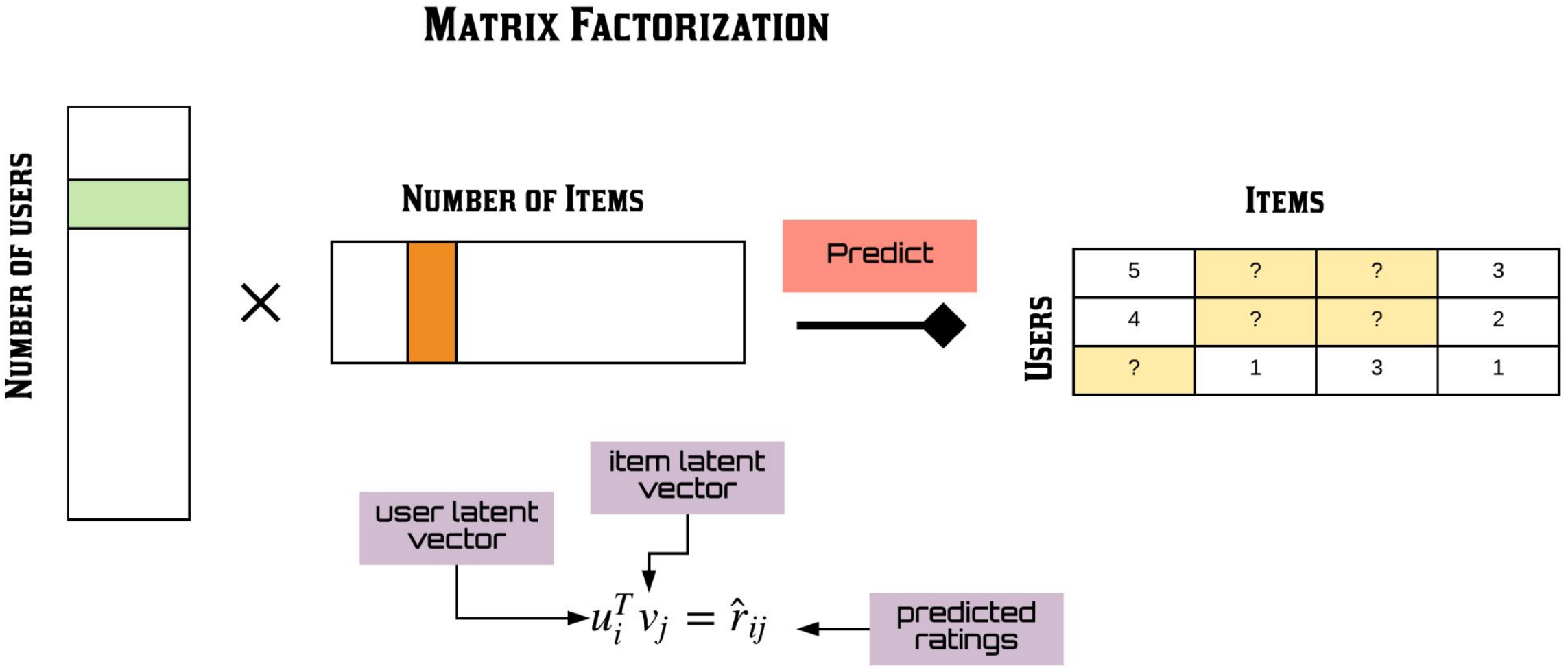
RAG: rechercher semantic

GNOMON DIGITAL

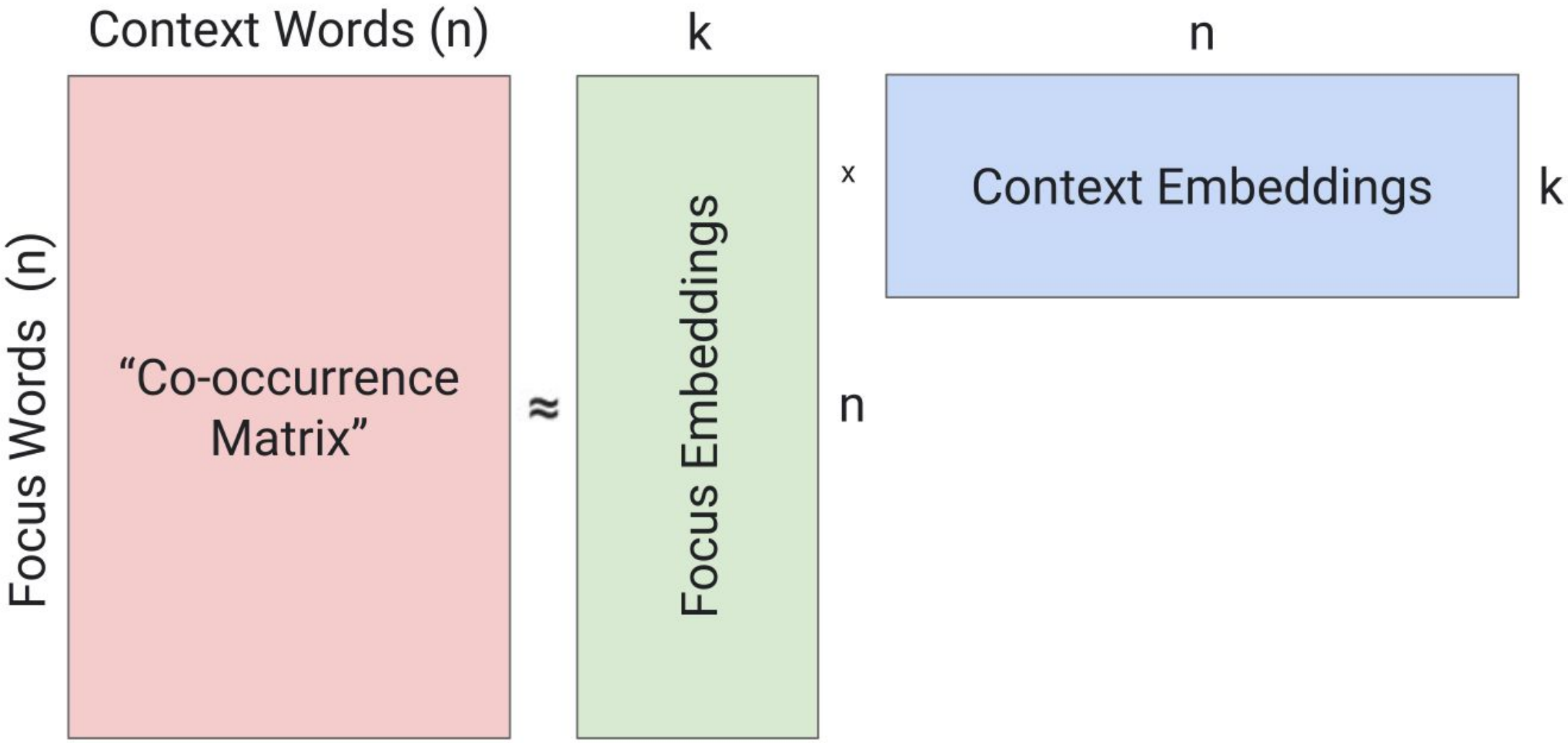
Step 2: Learn embedding vectors



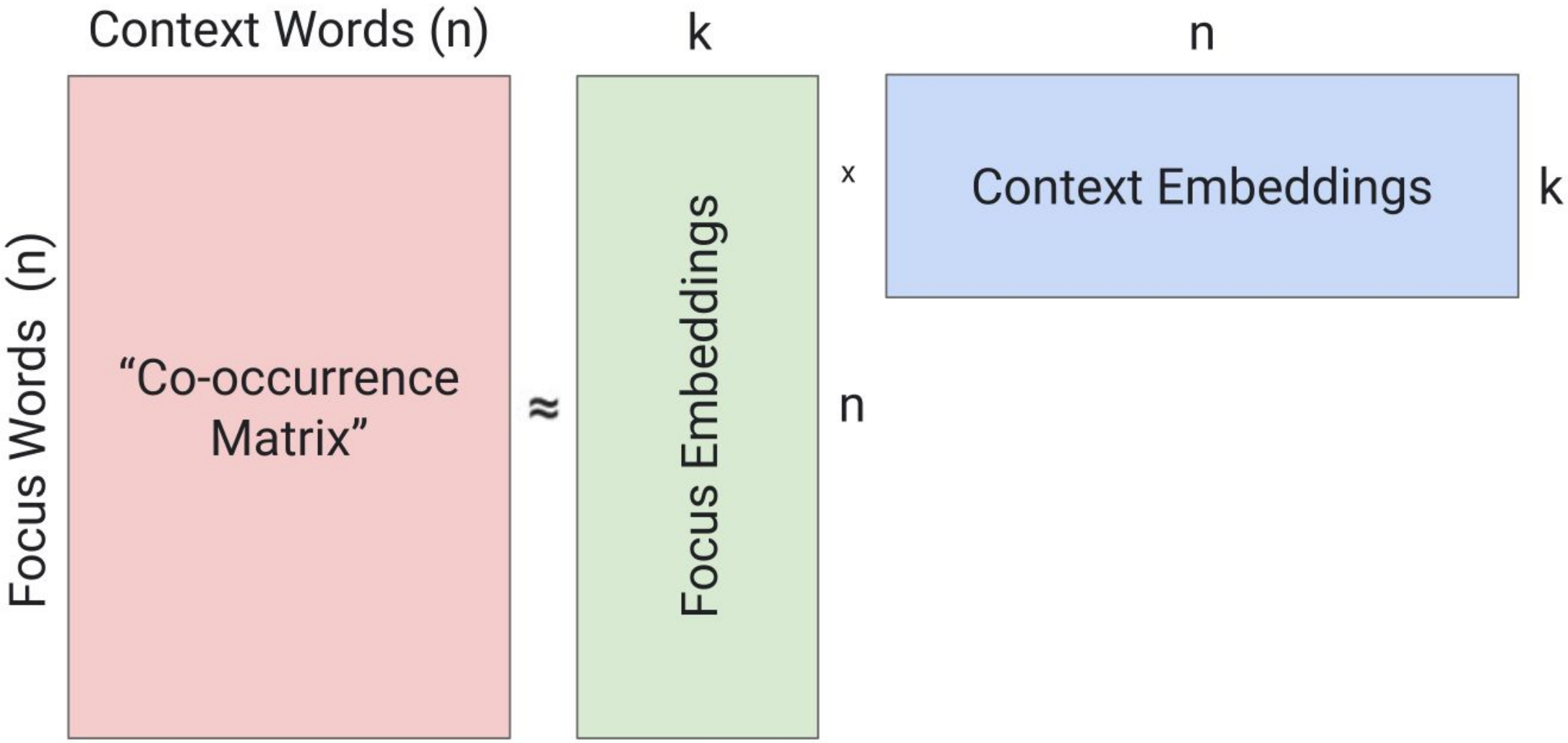
RAG: rechercher semantic



RAG: rechercher semantic



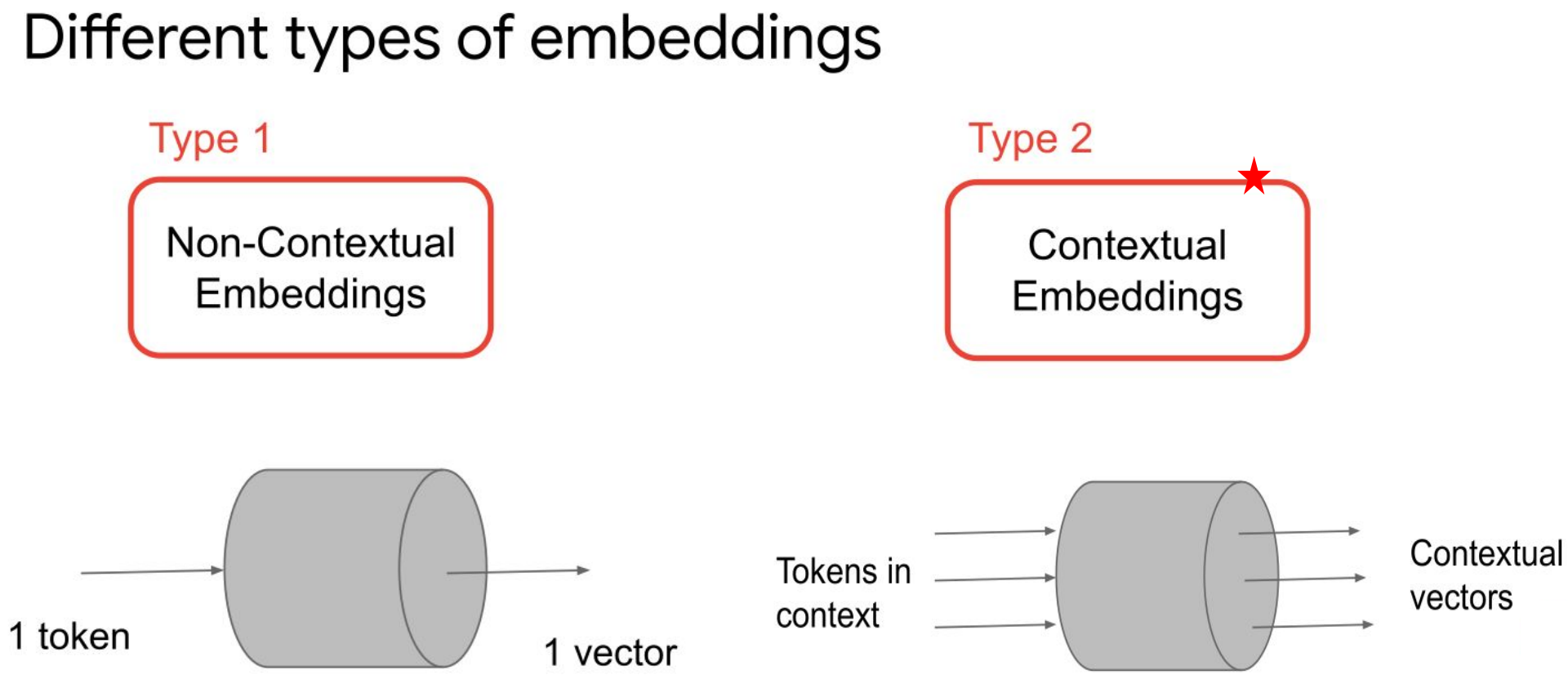
RAG: rechercher semantic



RAG: rechercher semantic

GNOMON DIGITAL

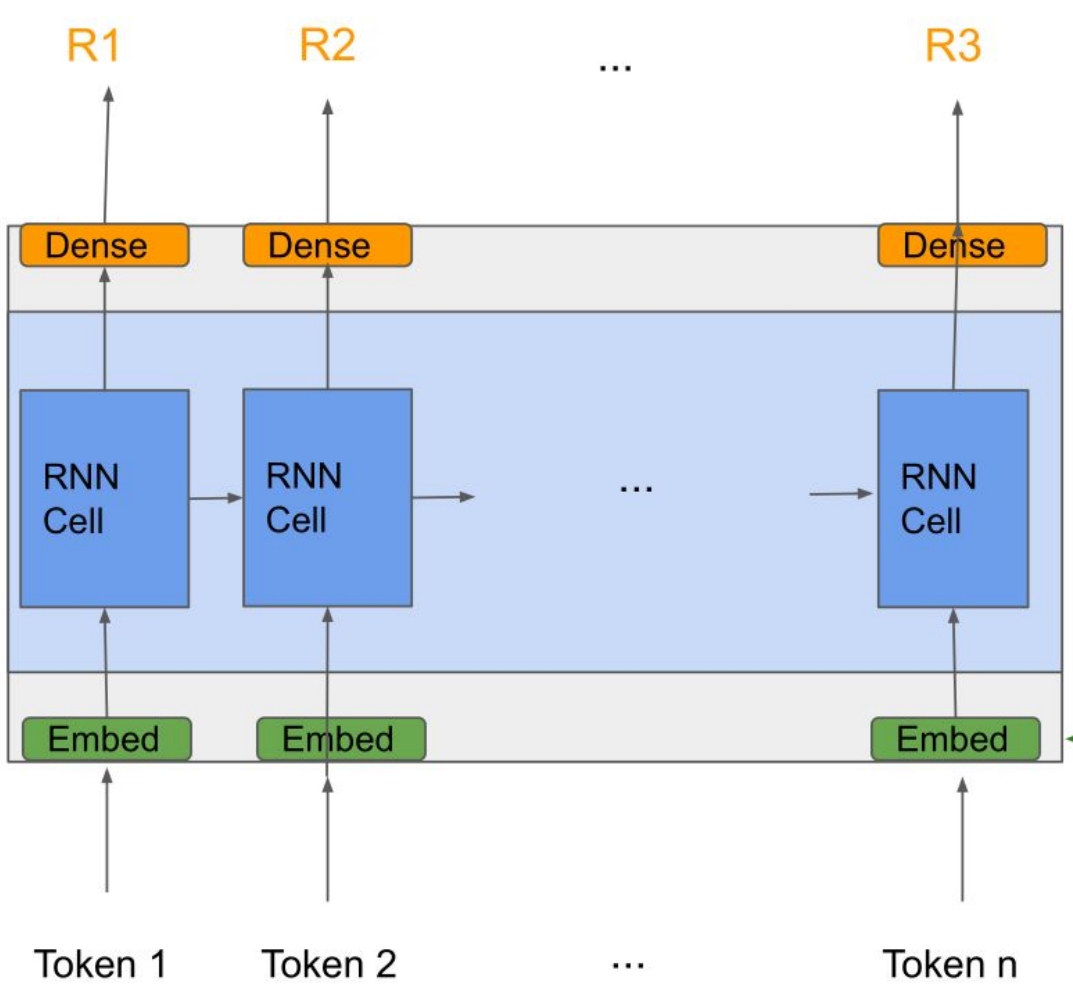
Step 2: Learn embedding vectors



RAG: rechercher semantic

GNOMON DIGITAL

Contextual: Recurrent Network Language Models



Contextual Representations

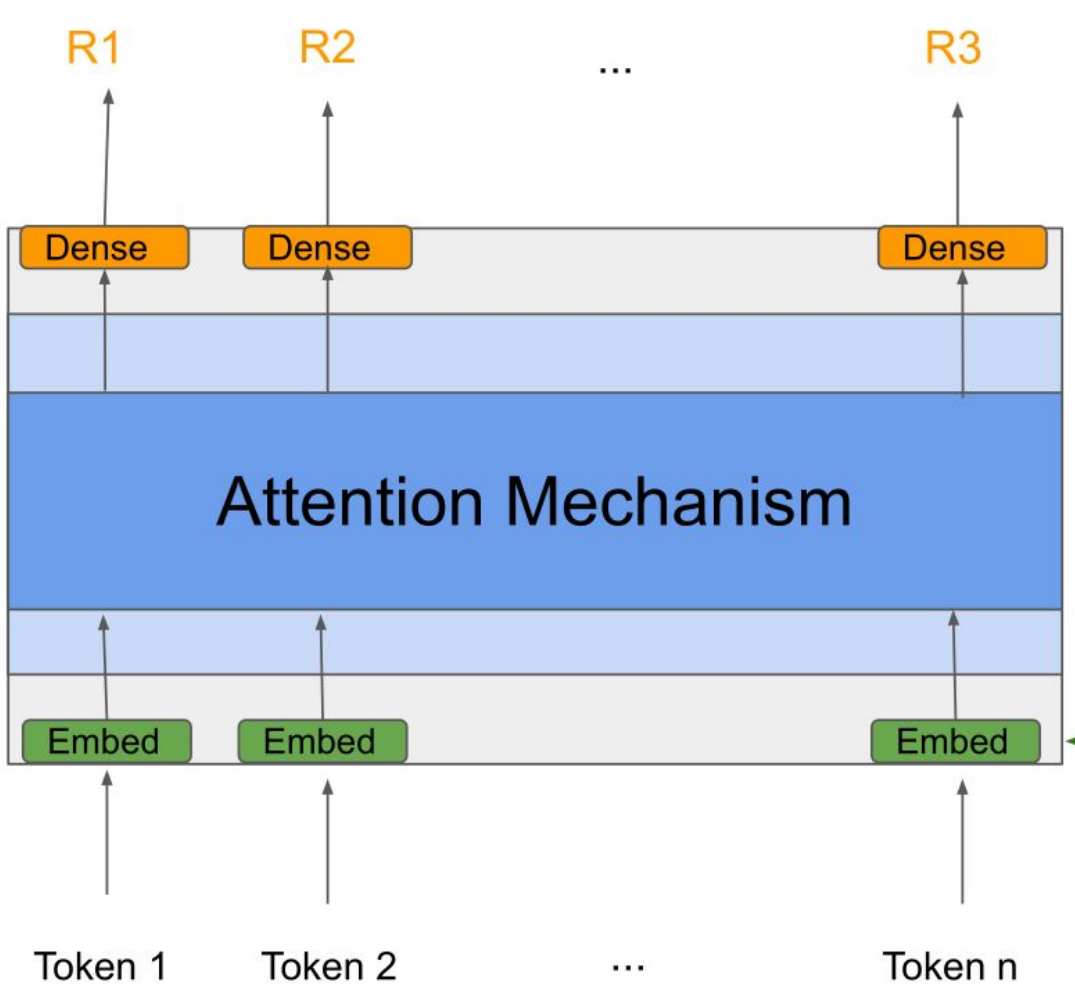
- Sequential computation
- Context only from one side
- Can be stacked

Non-Contextual Representations

RAG: rechercher semantic

GNOMON DIGITAL

Contextual: Transformer Language Models

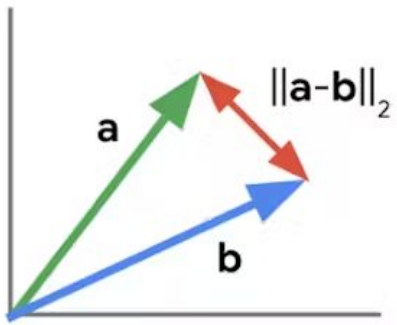


Contextual Representations

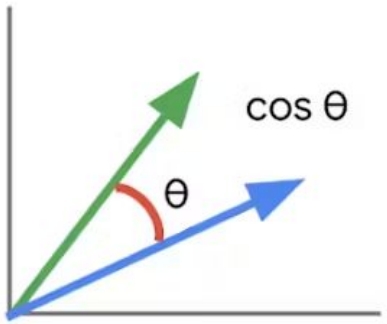
- Execute fully in parallel
- Context from both sides
- Can be stacked

Non-Contextual Representations

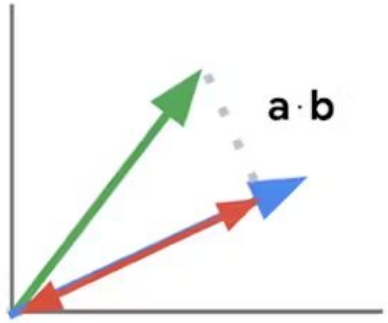
RAG: rechercher semantic



L2 distance



cosine similarity



inner product

2 dimensions x

1M items

= O(1M x 2)

Calculating vector similarity

RAG: rechercher semantic

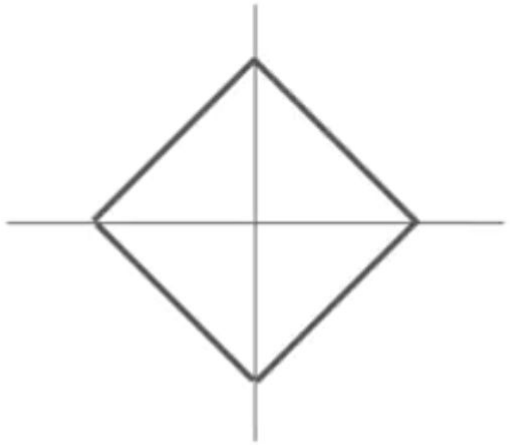
GNOMON DIGITAL

L2 distance (Euclidean) distance:

K-Nearest Neighbors: Distance Metric

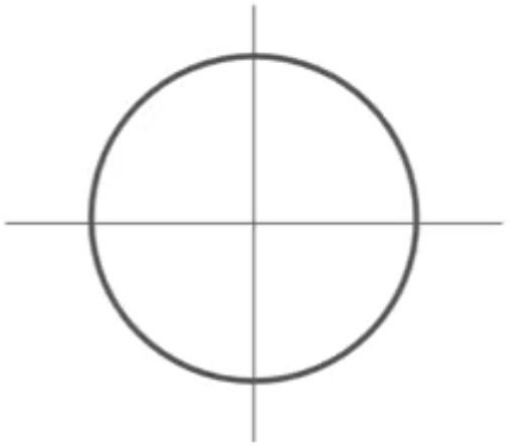
L1 (Manhattan) distance

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$



L2 (Euclidean) distance

$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$

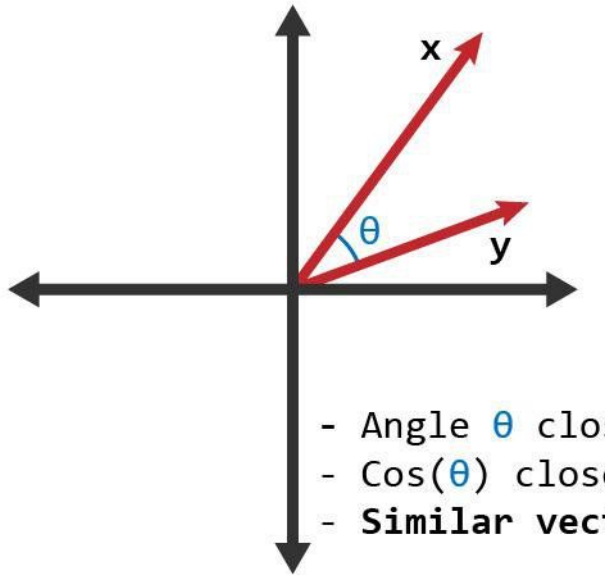


RAG: rechercher semantic

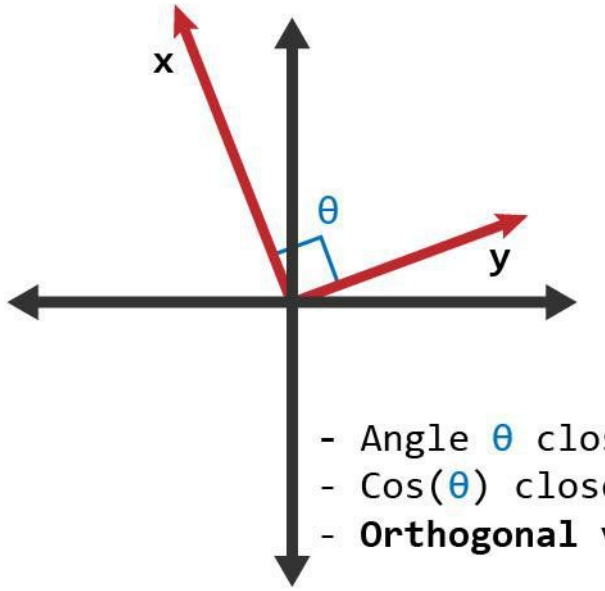
GNOMON DIGITAL

Cos similarity:

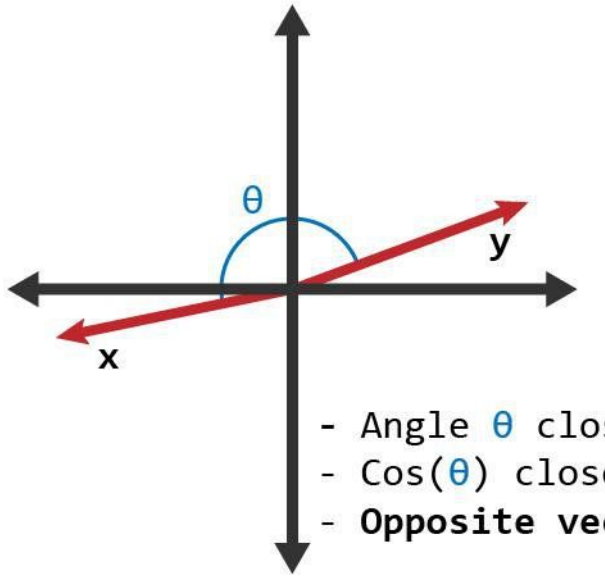
$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$



- Angle θ close to 0
- Cos(θ) close to 1
- **Similar vectors**



- Angle θ close to 90
- Cos(θ) close to 0
- **Orthogonal vectors**



- Angle θ close to 180
- Cos(θ) close to -1
- **Opposite vectors**

RAG: rechercher semantic

GNOMON®
DIGITAL

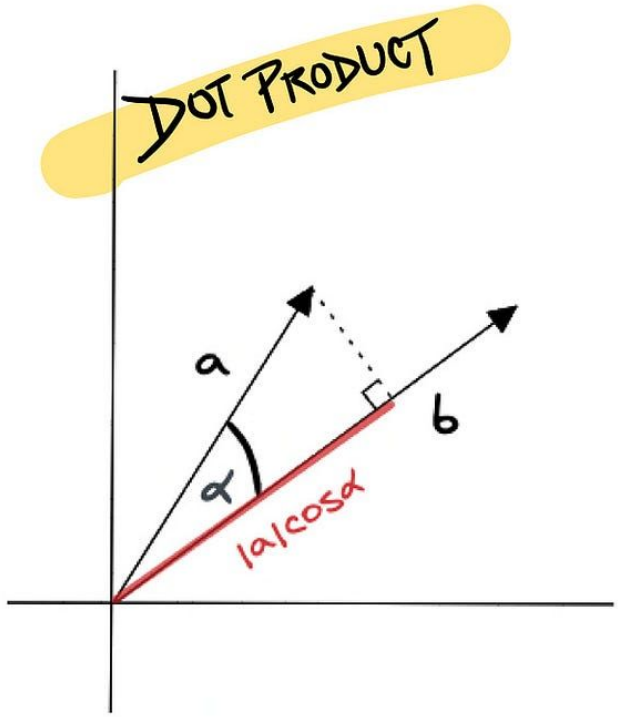
Dot product:

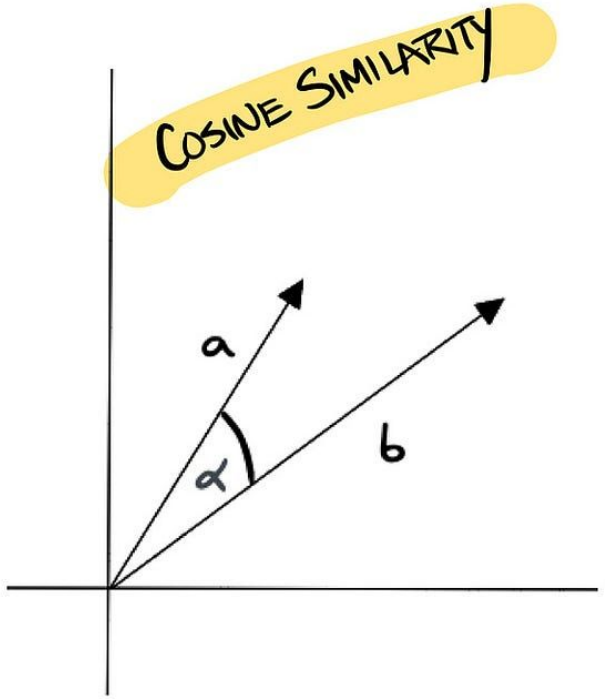
$$\mathbf{a} = \langle a_1, a_2, a_3 \rangle \quad \mathbf{b} = \langle b_1, b_2, b_3 \rangle$$

$$\boxed{\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + a_3 b_3}$$

RAG: rechercher semantic

GNOMON DIGITAL

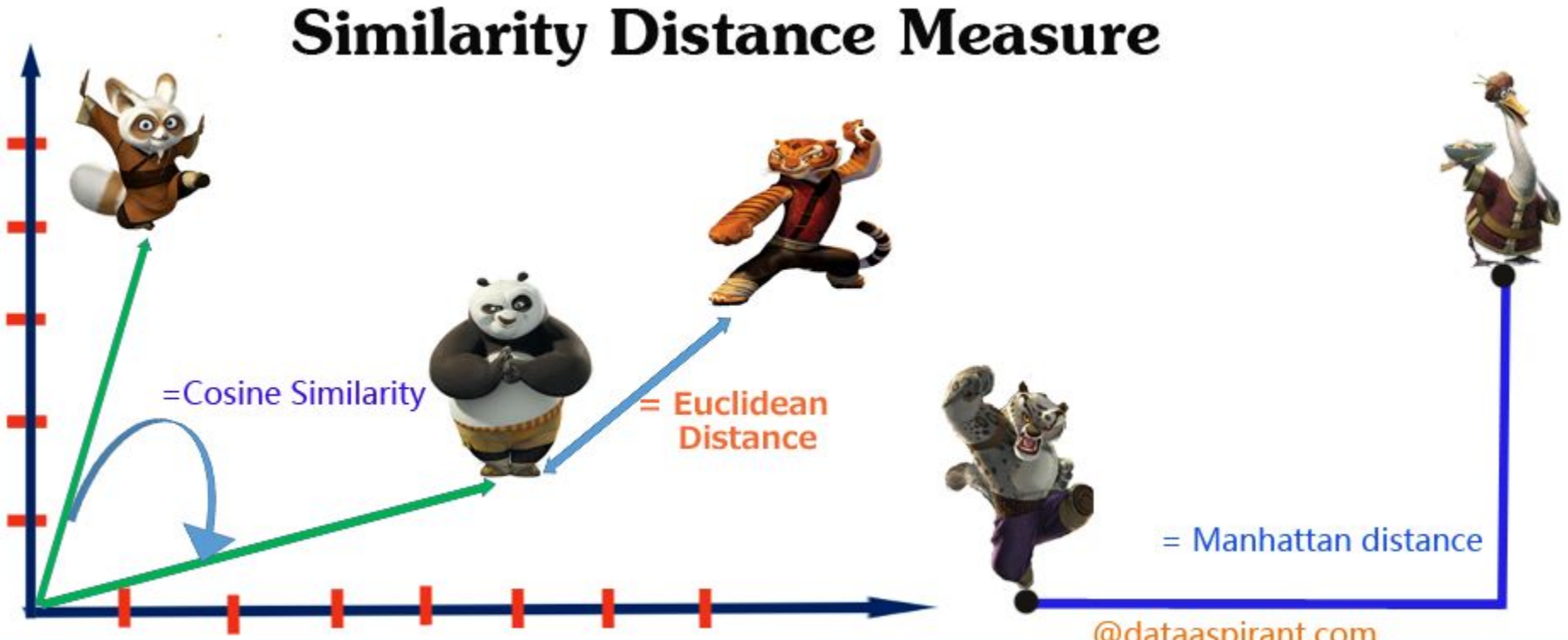




$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}||\mathbf{b}|cos\alpha$$

$$sim(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}|| \cdot ||\mathbf{b}||}$$

RAG: rechercher semantic



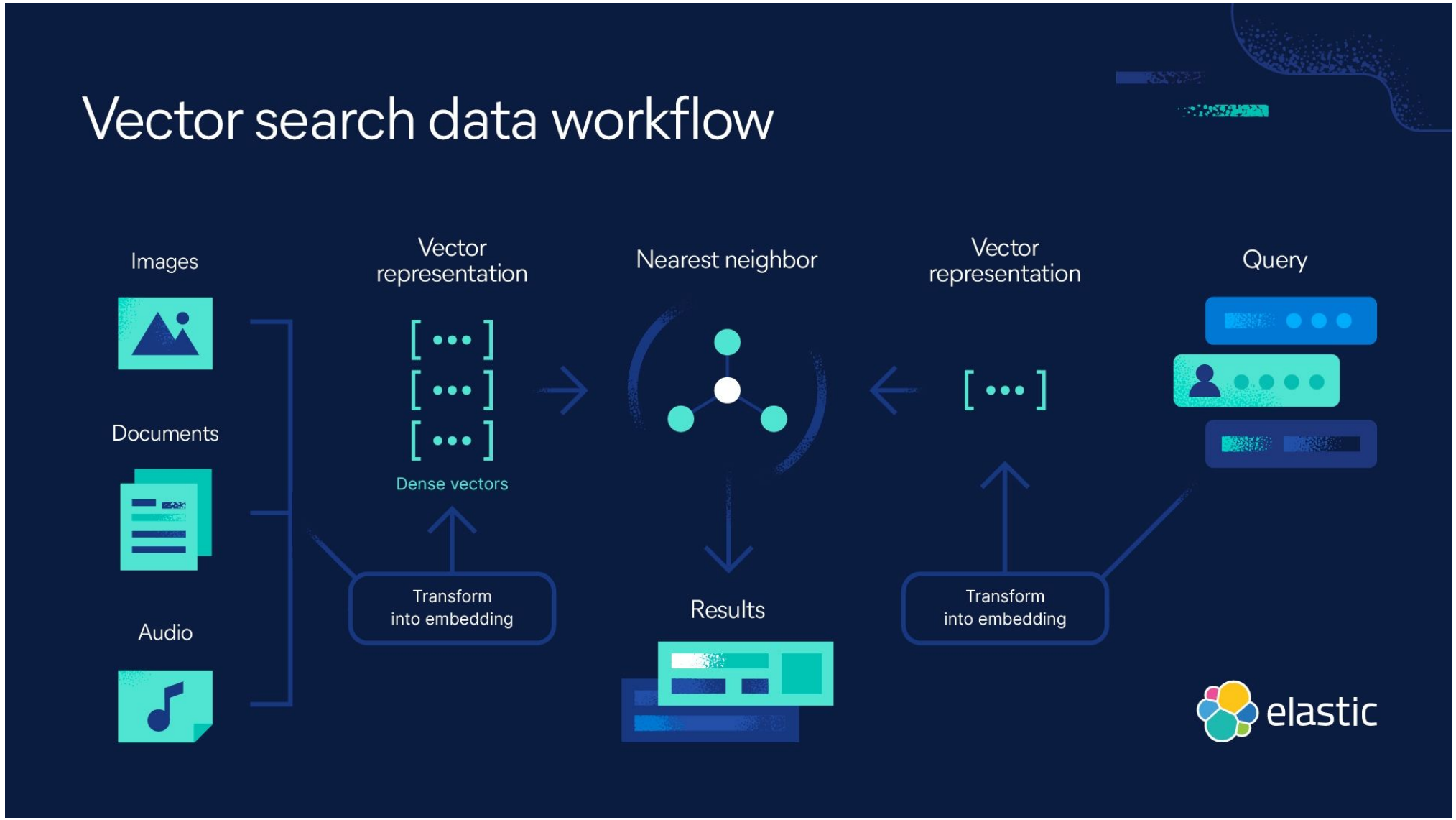
RAG: rechercher semantic

GNOMON DIGITAL

https://github.com/GoogleCloudPlatform/asl-ml-immersion/blob/master/notebooks/vertex_genai/labs/semantic_matching_with_gemini.ipynb

RAG: rechercher semantic

GNOMON DIGITAL



RAG: rechercher semantic

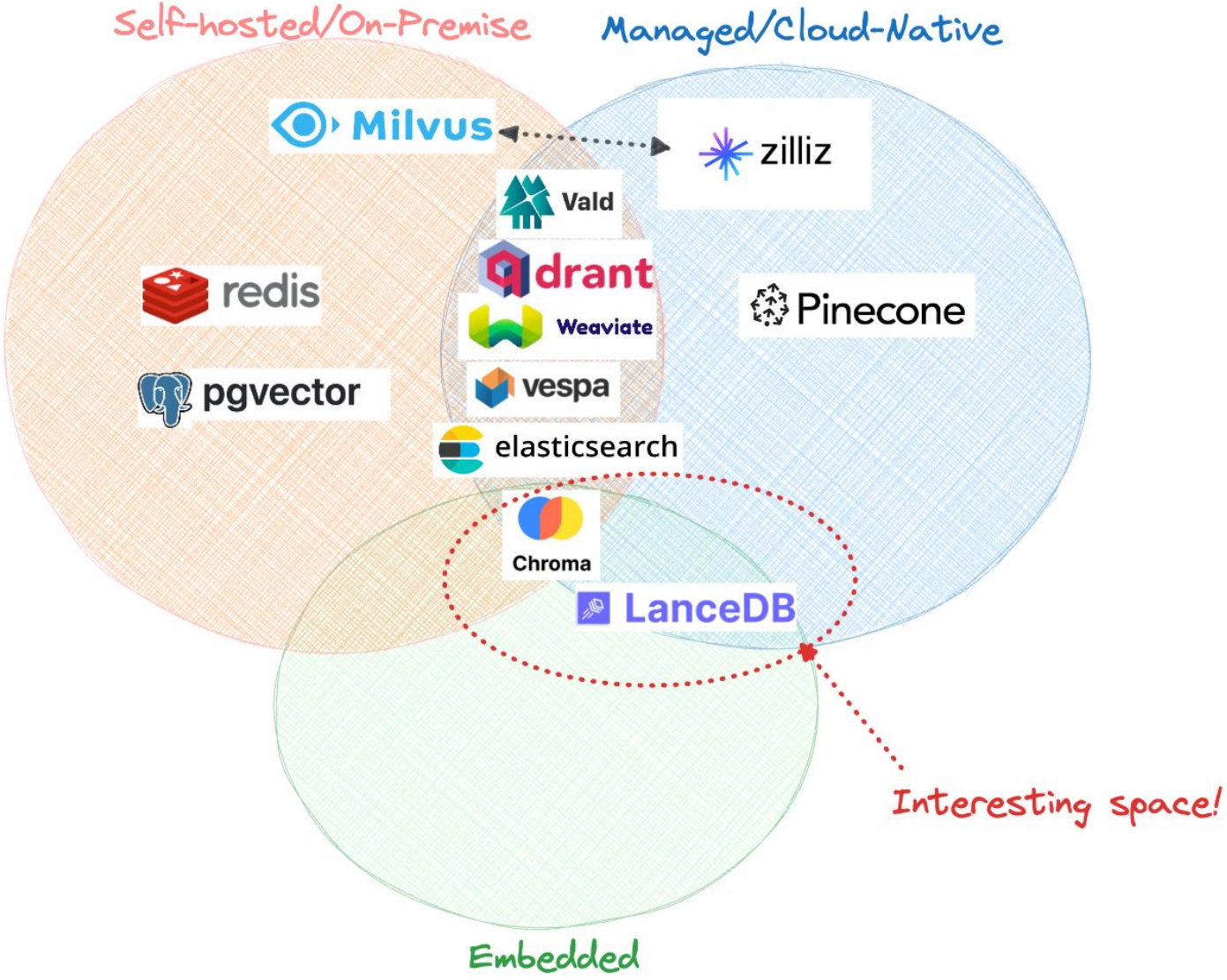
GNOMON DIGITAL

Recherche par vectorielle est splitté en deux parties:

- Offline -> Calcule lourde /lente:
  - Ajouter un vector index (keys= vectors)
  - Sauve tous les embedding vecteurs dans ces index
- Online -> Calcule efficace / Rapid
  - Algorithm search dans vectorDB

RAG: rechercher semantic

GNOMON DIGITAL



RAG: rechercher semantic

GNOMON®
DIGITAL

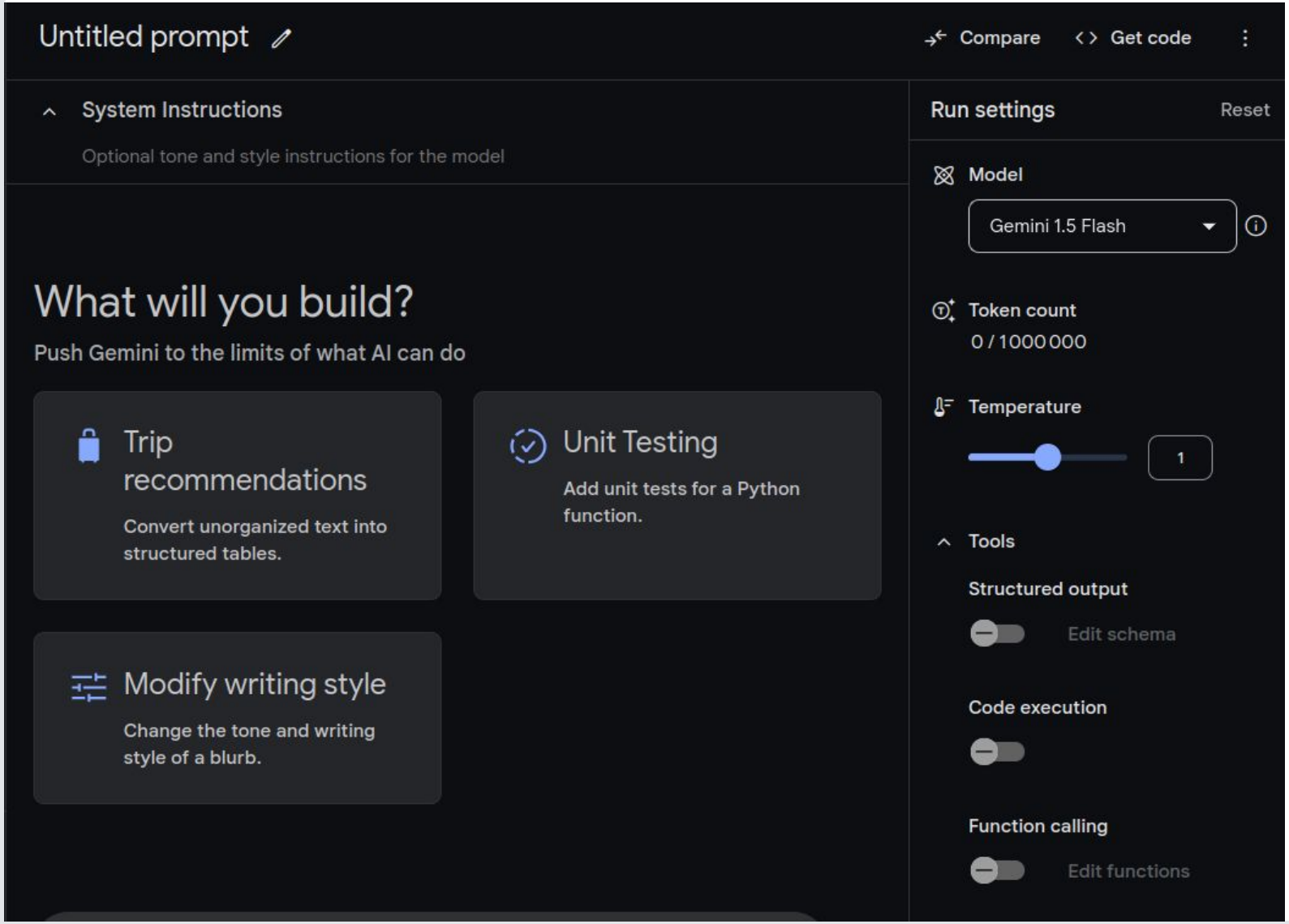
https://github.com/GoogleCloudPlatform/asl-ml-immersion/blob/master/notebooks/vertex_genai/solutions/semantic_search_with_vector_search.ipynb

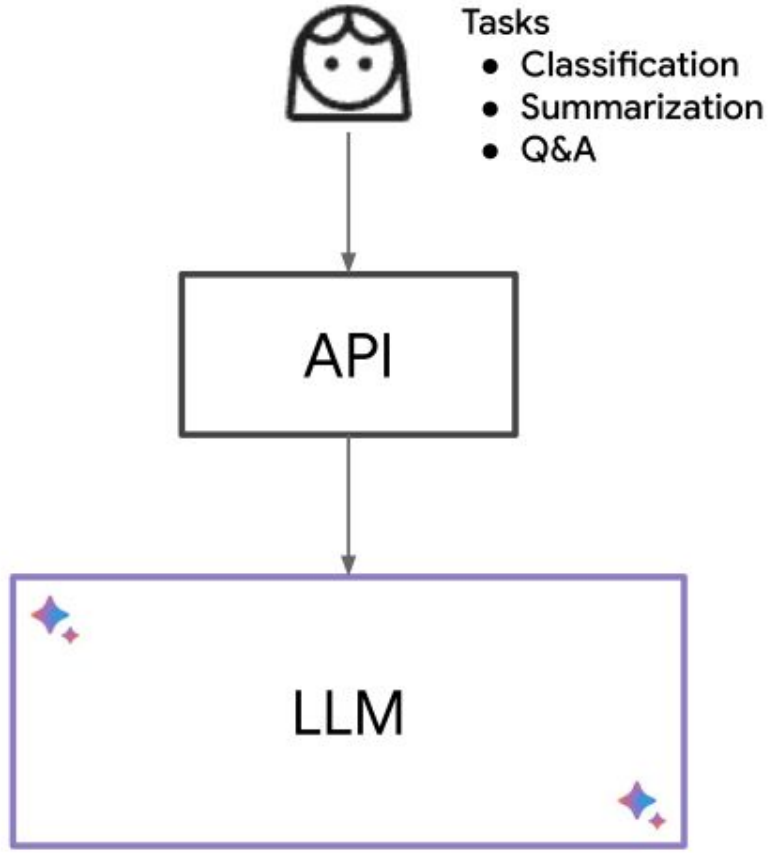
Introduction à LangChain et LlamaIndex

GNOMON DIGITAL

Introduction à LangChain et LlamaIndex

GNOMON DIGITAL





https://aistudio.google.com/app/prompts/new_chat

Introduction à LangChain et LlamaIndex

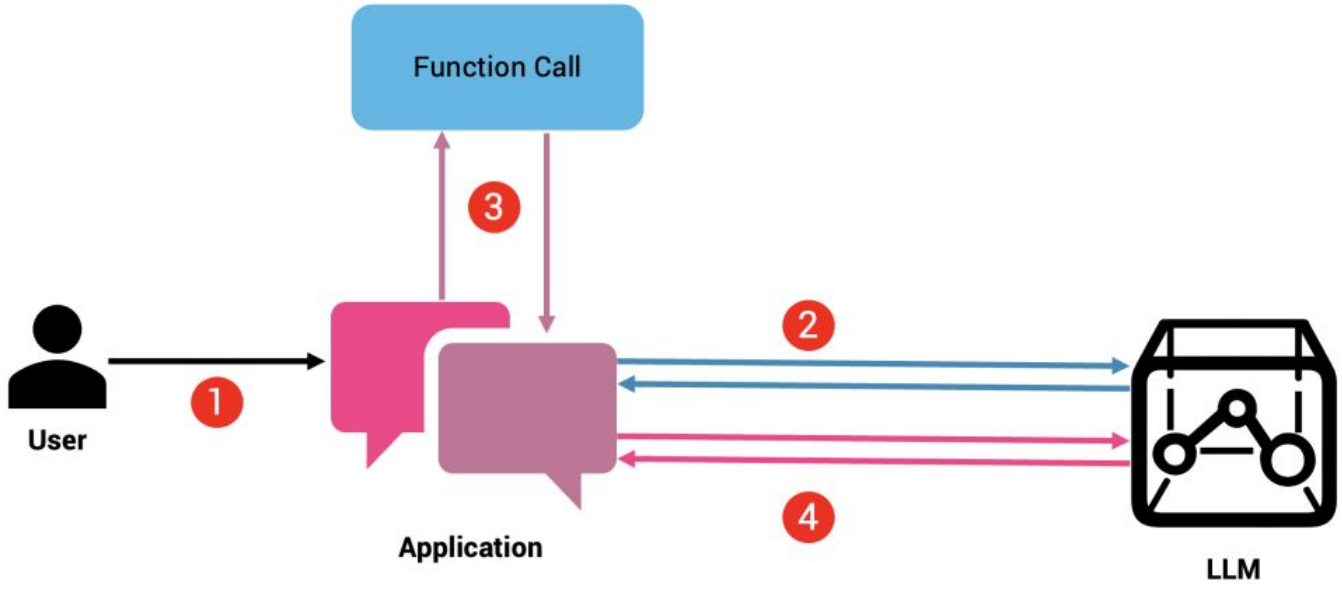
GNOMON DIGITAL

Tool LLM

TOOLLLM: FACILITATING LARGE LANGUAGE MODELS TO MASTER 16000+ REAL-WORLD APIS

Yujia Qin[1*], Shihao Liang[1*], Yining Ye[1], Kunlun Zhu[1], Lan Yan[1], Yaxi Lu[1], Yankai Lin[3†], Xin Cong[1], Xiangru Tang[4], Bill Qian[4], Sihan Zhao[1], Lauren Hong[1], Runchu Tian[1], Ruobing Xie[5], Jie Zhou[5], Mark Gerstein[4], Dahai Li[2,6], Zhiyuan Liu[1†], Maosong Sun[1†]
[1]Tsinghua University [2]ModelBest Inc. [3]Renmin University of China
[4]Yale University [5]WeChat AI, Tencent Inc. [6]Zhihu Inc.
yujiaqin16@gmail.com

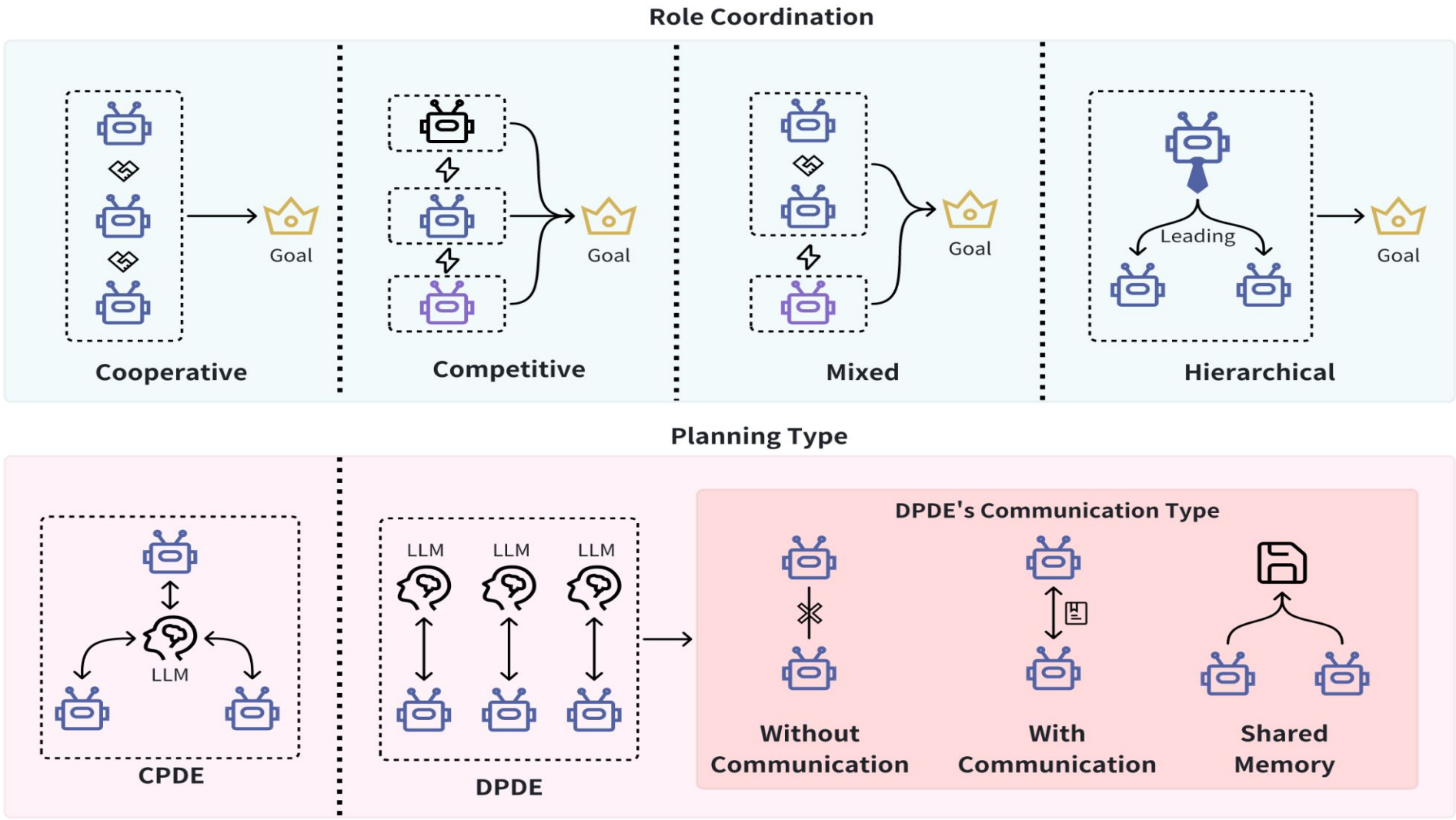
SQL
Google Search
Code
…



Introduction à LangChain et LlamaIndex

GNOMON DIGITAL

LLM Agent



Introduction à LangChain et LlamaIndex

GNOMON DIGITAL

LLM Agent

Exploring Large Language Model based Intelligent Agents: Definitions, Methods, and Prospects

Yuheng Cheng[1] 1 **Ceyao Zhang**[1] 1 **Zhengwen Zhang**[1]

Xiangrui Meng[1] **Sirui Hong**[2] **Wenhao Li**[1] **Zihao Wang**[3] **Zekai Wang**[4]

Feng Yin[1] **Junhua Zhao**[1] **Xiuqiang He**[5]

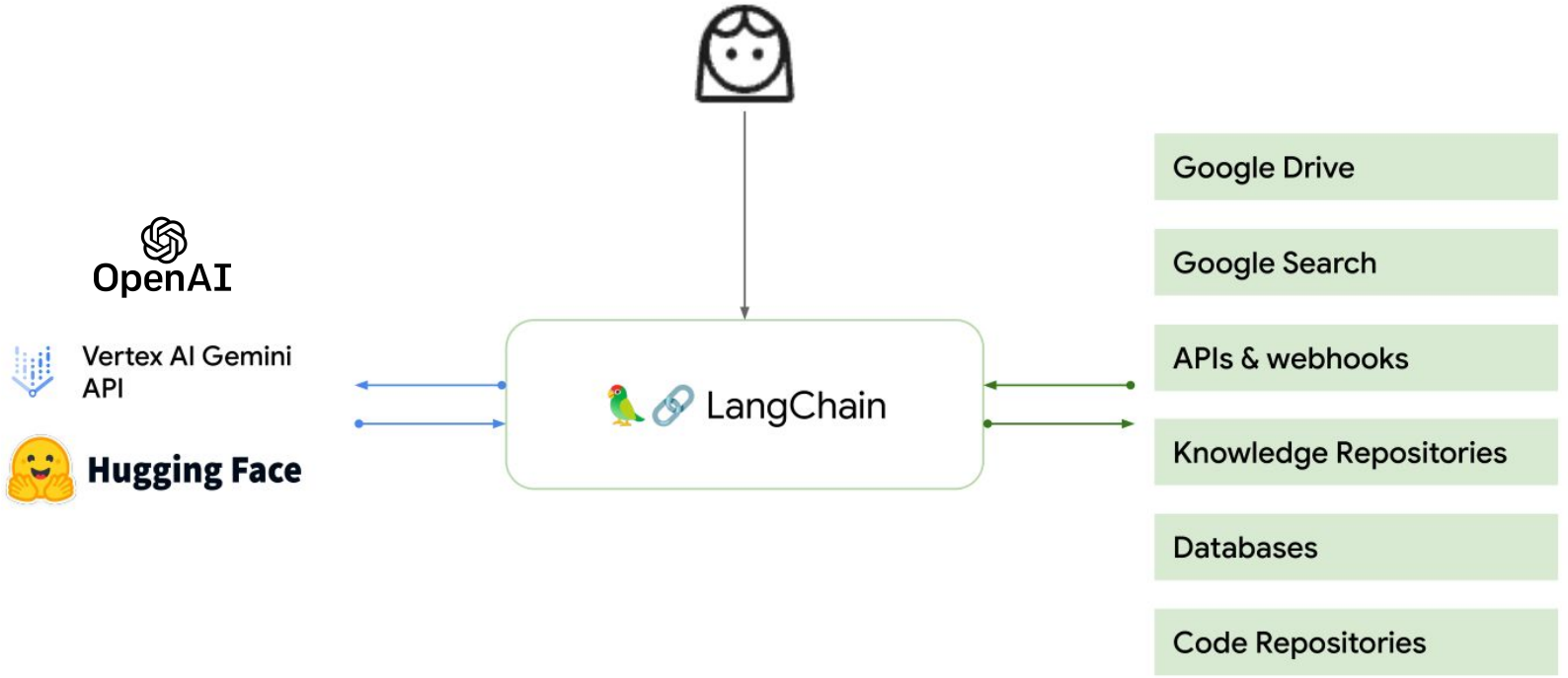
[1]The Chinese University of Hong Kong, Shenzhen

[2]DeepWisdom [3]Peking University [4]Yantu.ai [5]FiT, Tencent

These authors contributed to the work equally and should be regarded as co-first authors.Corresponding author

Introduction à LangChain et LlamaIndex

GNOMON DIGITAL



Usages possibles:

- Question & Answering over private corpus
- Large document Summarization
- Personal assistants & Chatbots
- Querying tabular data
- Code understanding
- API interactions
- Extraction
- Evaluation

Introduction à LangChain et LlamaIndex

GNOMON DIGITAL

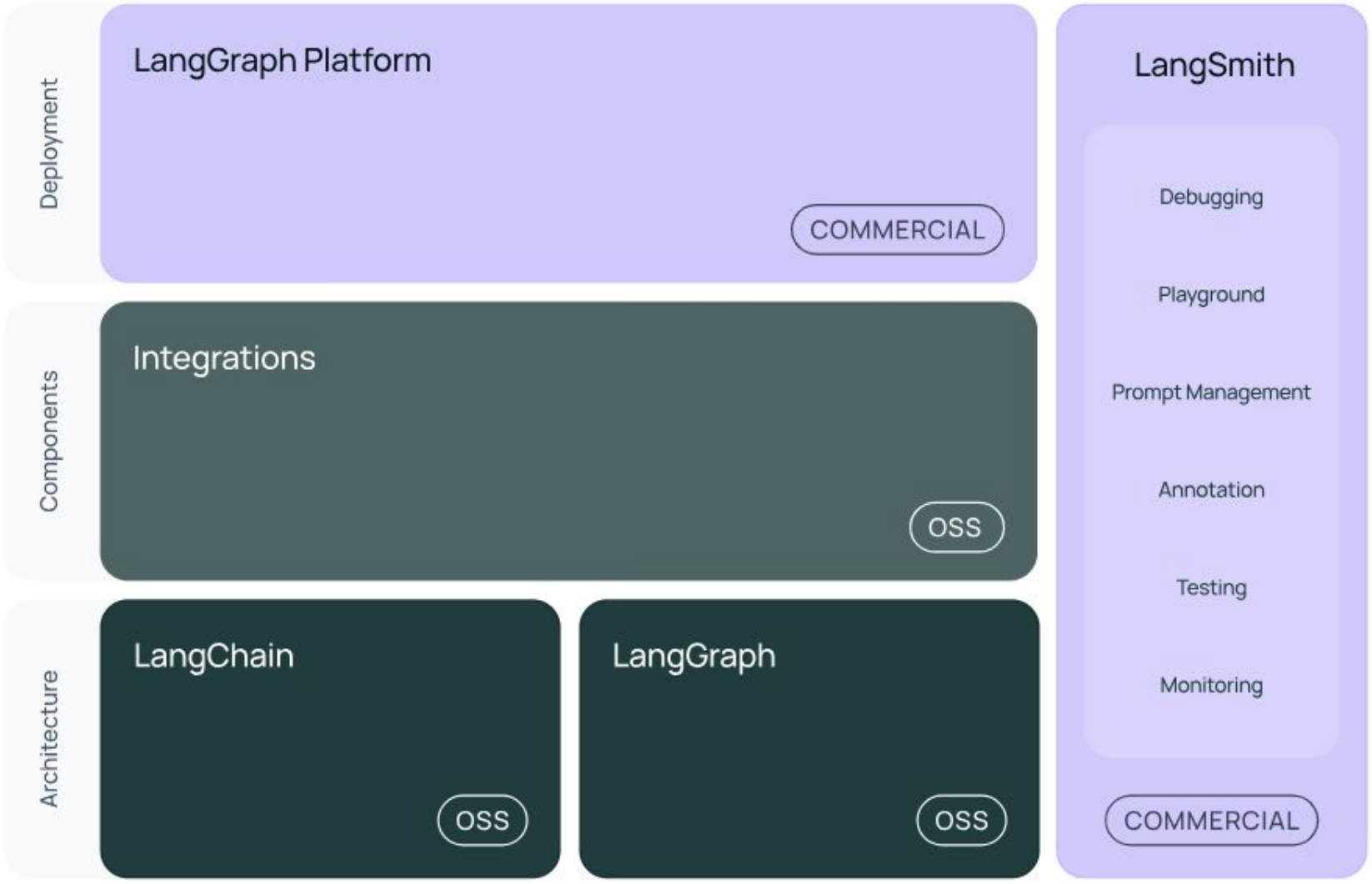
LlamaIndex vs. LangChain

datasciencedojo
data science for everyone

Features	🦙 LlamaIndex	🦜🔗 LangChain
Primary focus	Intelligent search, data indexing and retrieval	Building a wide range of Gen AI applications
Data handling	Ingesting, structuring, and accessing private or domain-specific data	Loading, processing, and indexing data for various uses
Customization	Offers tools for integrating private data into LLMs	Highly customizable, users can chain multiple components
Flexibility	Specialized for efficient and fast search	General-purpose framework with flexibility in application behavior
Deployment	Ideal for proprietary or specialized data	Facilitates the deployment of bespoke NLP applications
Use cases	Best for applications that require quick data lookup and retrieval	Suitable for applications that require complex interactions like chatbots, GQA, summarization

Introduction à LangChain

GNOMON DIGITAL

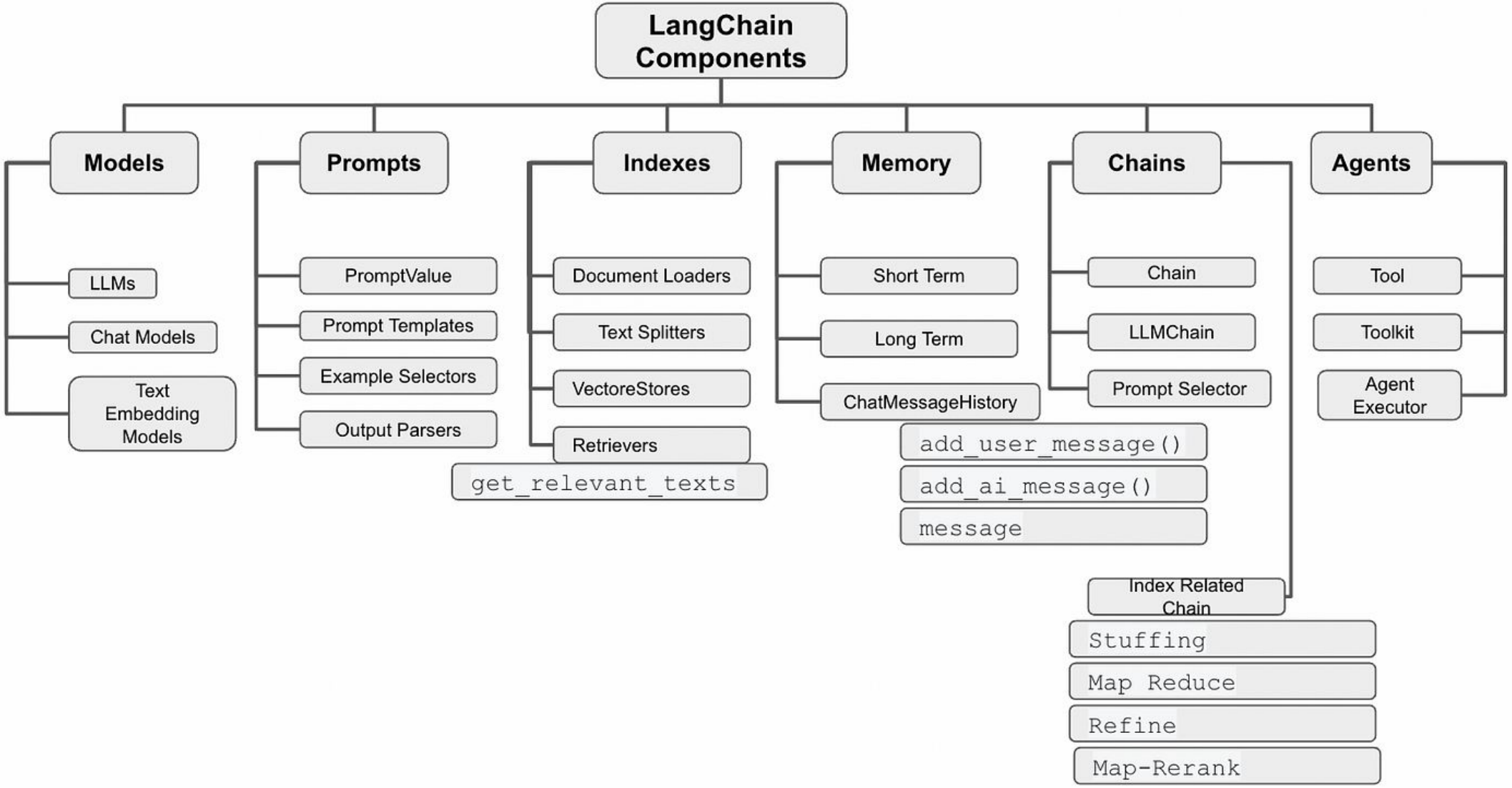


LangChain Framework

LangChain

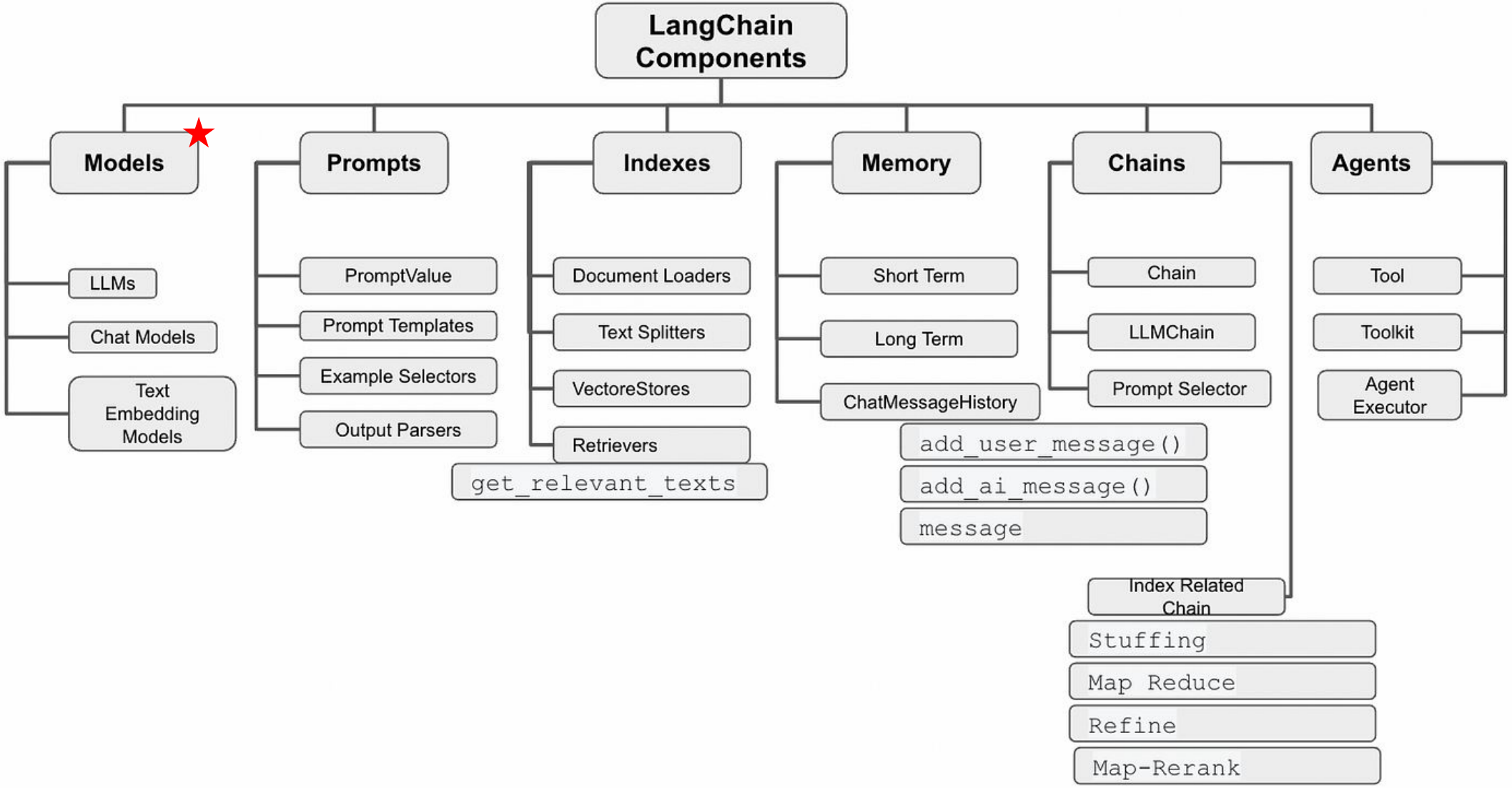
Introduction à LangChain et LlamaIndex

GNOMON DIGITAL



Introduction à LangChain et LlamaIndex

GNOMON DIGITAL



Introduction à LangChain et LlamaIndex

GNOMON DIGITAL

- Model IO focuse sur les functionalités basique de LLM (text entrée/sortie)
- Avantage LangChain:
  - Une seul interface aux tous les providers LLMs (Google, OpenAI et Hugging Face etc)
  - Text mode et Chat mode
  - Pas d'adaptation à faire si on change le modèle LLM

https://python.langchain.com/docs/tutorials/llm_chain/

Introduction à LangChain et LlamaIndex

GNOMON DIGITAL

- Data Connection
  - Possibilité de connecter LLM avec des données externes (RAG)
- Avantages:
  - Integration functions disponibles aux data sources populaires (CSVs, PDFs, AWS, Google Cloud, etc)
  - Possible d’utiliser/changer la solution VectorDB

https://python.langchain.com/docs/integrations/document_loaders/

Introduction à LangChain et LlamaIndex

GNOMON DIGITAL

```
from langchain.vectorstores import Chroma
from langchain.embeddings import VertexAIEmbeddings
from langchain_community.document_loaders import PyPDFDirectoryLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter

documents = PyPDFDirectoryLoader("pdfs/").load()

text_splitter = RecursiveCharacterTextSplitter(chunk_size=800,chunk_overlap=400)

document_chunks = text_splitter.split_documents(documents)

embedding = VertexAIEmbeddings(model_name="textembedding-gecko@001”)

db = Chroma.from_documents(document_chunks, embedding,
persist_directory="./vectorstore")
```

https://python.langchain.com/docs/integrations/document_loaders/

Introduction à LangChain et LlamaIndex

GNOMON DIGITAL

- Chains :
  - Les sorties d’un LLM peuvent être les entrées d’un autre LLM (Appel APIs LLM en chain)
- Avantages:
  - Facilite l’utilisation d’apelle en chains

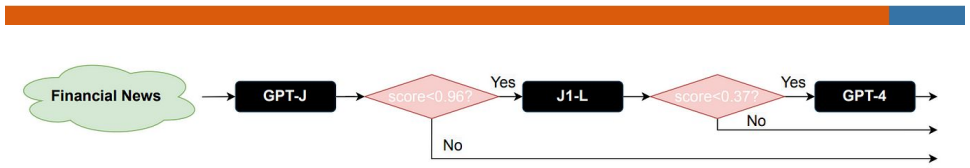
Introduction à LangChain et LlamaIndex

GNOMON DIGITAL

stanford-futuredata/
FrugalGPT

FrugalGPT: better quality and lower cost for LLM applications

2 Contributors · 3 Issues · 187 Stars · 22 Forks



(a) Learned FrugalGPT strategy

GPT-4 price up ✗

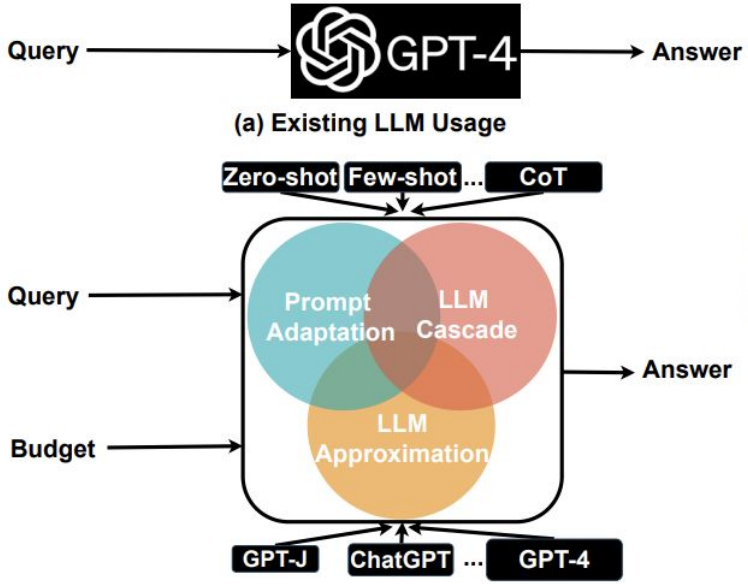
FrugalGPT price down ✓

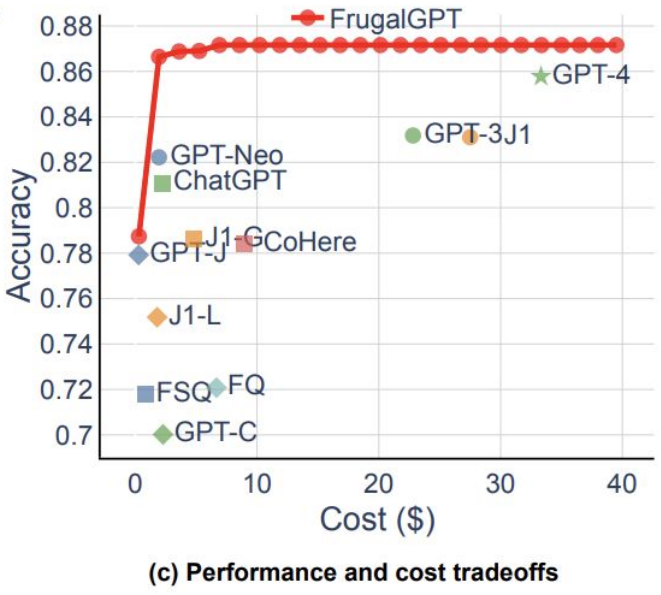
(b) A query and response example

Approch	Accuracy	Cost ($)
GPT-4	0.857	33.1
FrugalGPT	**0.872**	**6.5**

(c) Overall performance and cost

Figure 3: A case study of FrugalGPT on the HEADLINES dataset. (a) The cascade strategy that FrugalGPT learned on this dataset with overall budget $6.5, one fifth of GPT-4's cost. FrugalGPT avoids querying GPT-4 as long as GPT-J and J1-L produce high-quality answers. (b) Sometimes GPT-4 makes a mistake, but FrugalGPT learns to use the correct answers by J-1 and GPT-J. (c) Overall, we observe that FrugalGPT reduces the cost by 80%, while improves the accuracy by 1.5% compared to GPT-4.



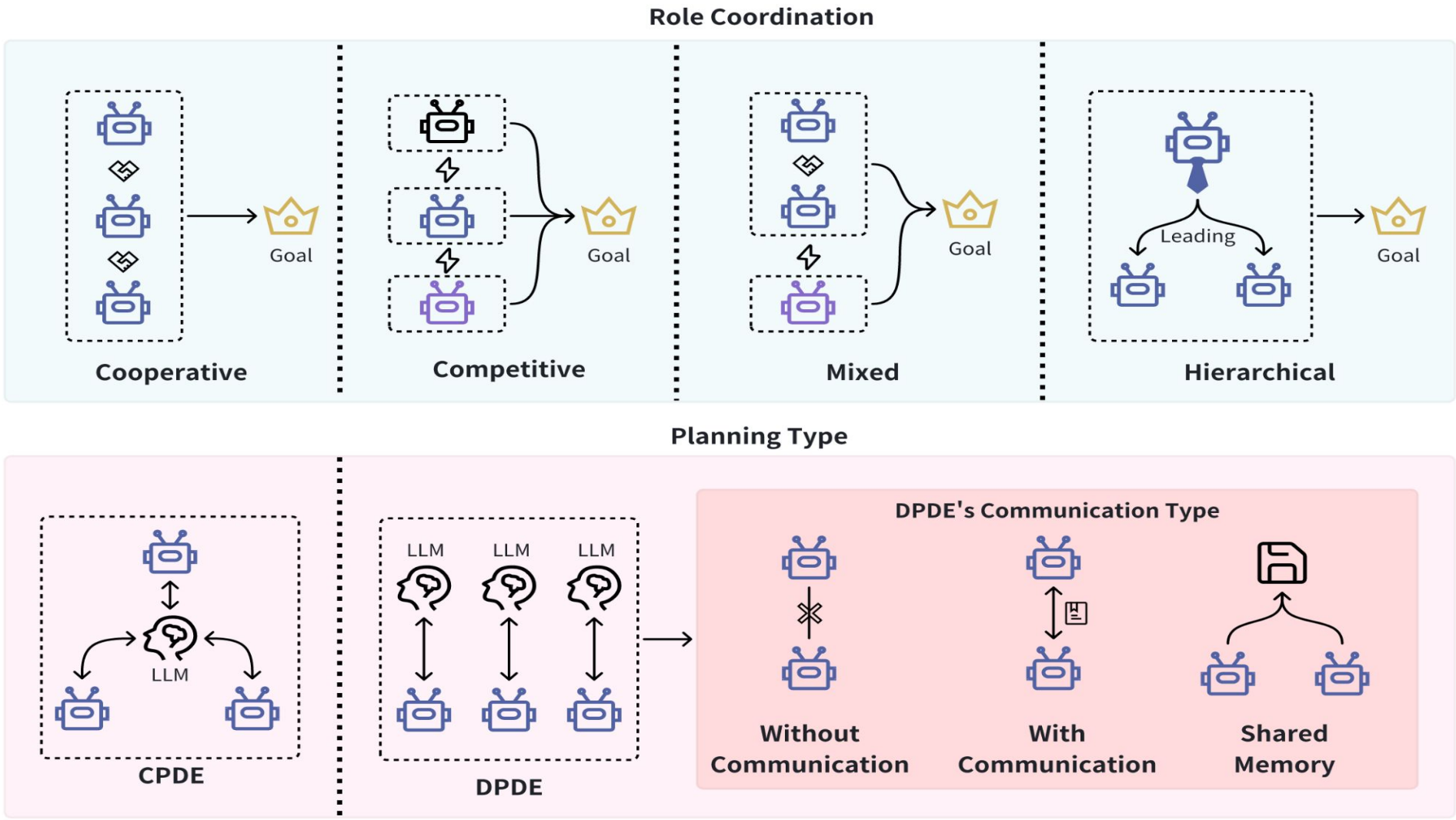


(c) Performance and cost tradeoffs

Introduction à LangChain et LlamaIndex

GNOMON DIGITAL

LLM Agent



Introduction à LangChain et LlamaIndex

GNOMON DIGITAL

- Memory :
  - Laisser tes modèles d'y accès des contexts historiques des précédents conversations

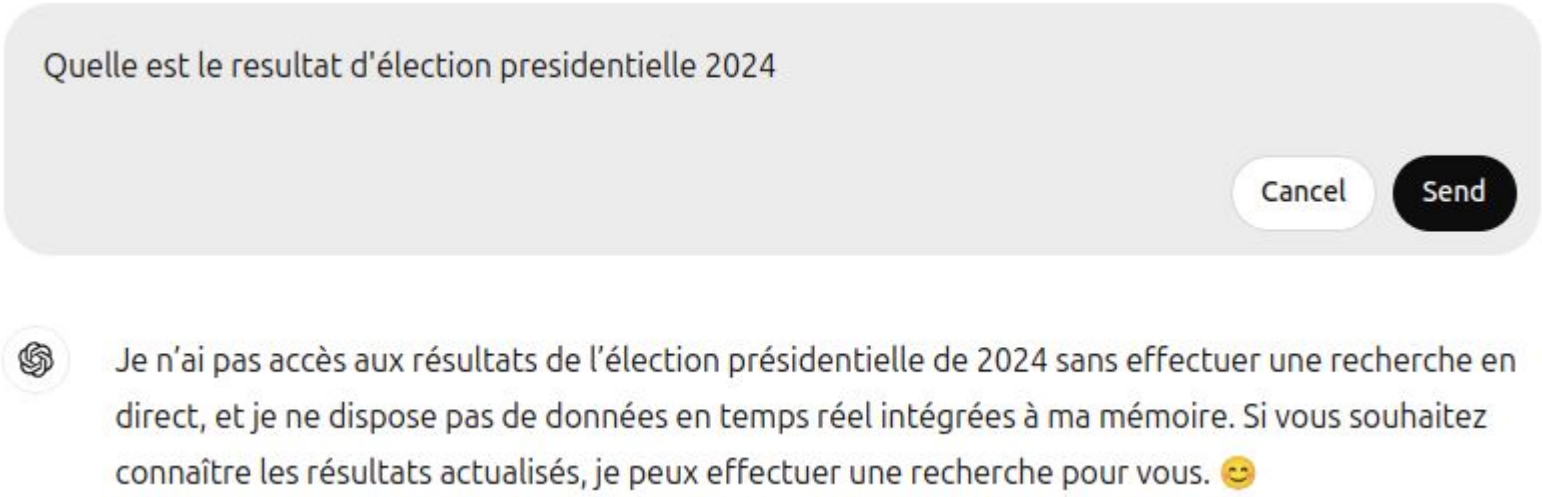
```
from langchain.llms import VertexAI
from langchain.chains import ConversationChain
from langchain.memory import ConversationBufferMemory

llm = VertexAI(model_name="codechat-bison")

conversation = ConversationChain(
    llm=llm, verbose=True, memory=ConversationBufferMemory()
)
```

https://github.com/GoogleCloudPlatform/asl-ml-immersion/blob/master/notebooks/vertex_genai/solutions/gemini_langchain_components.ipynb

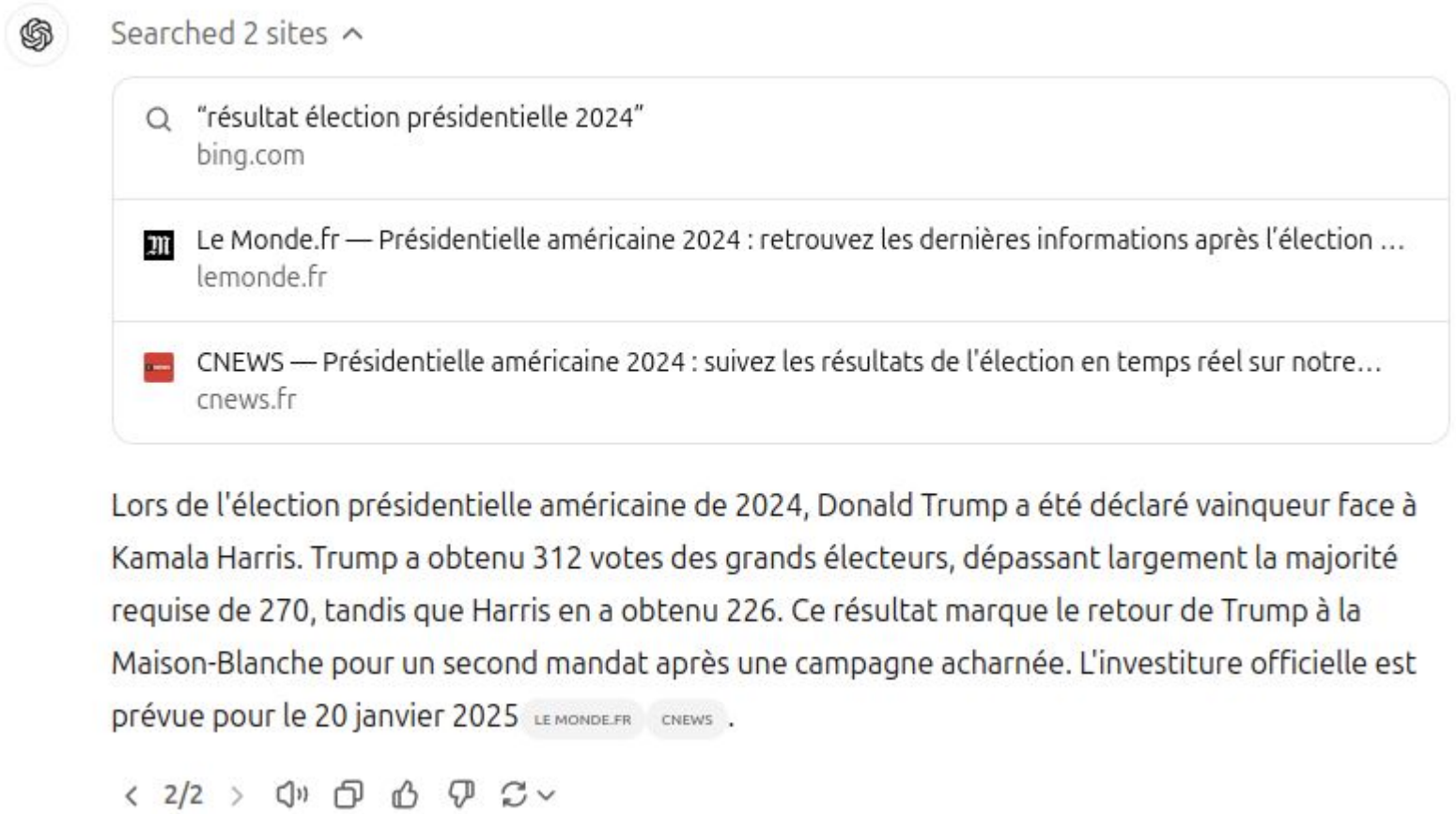
RAG

La limite de LLM:

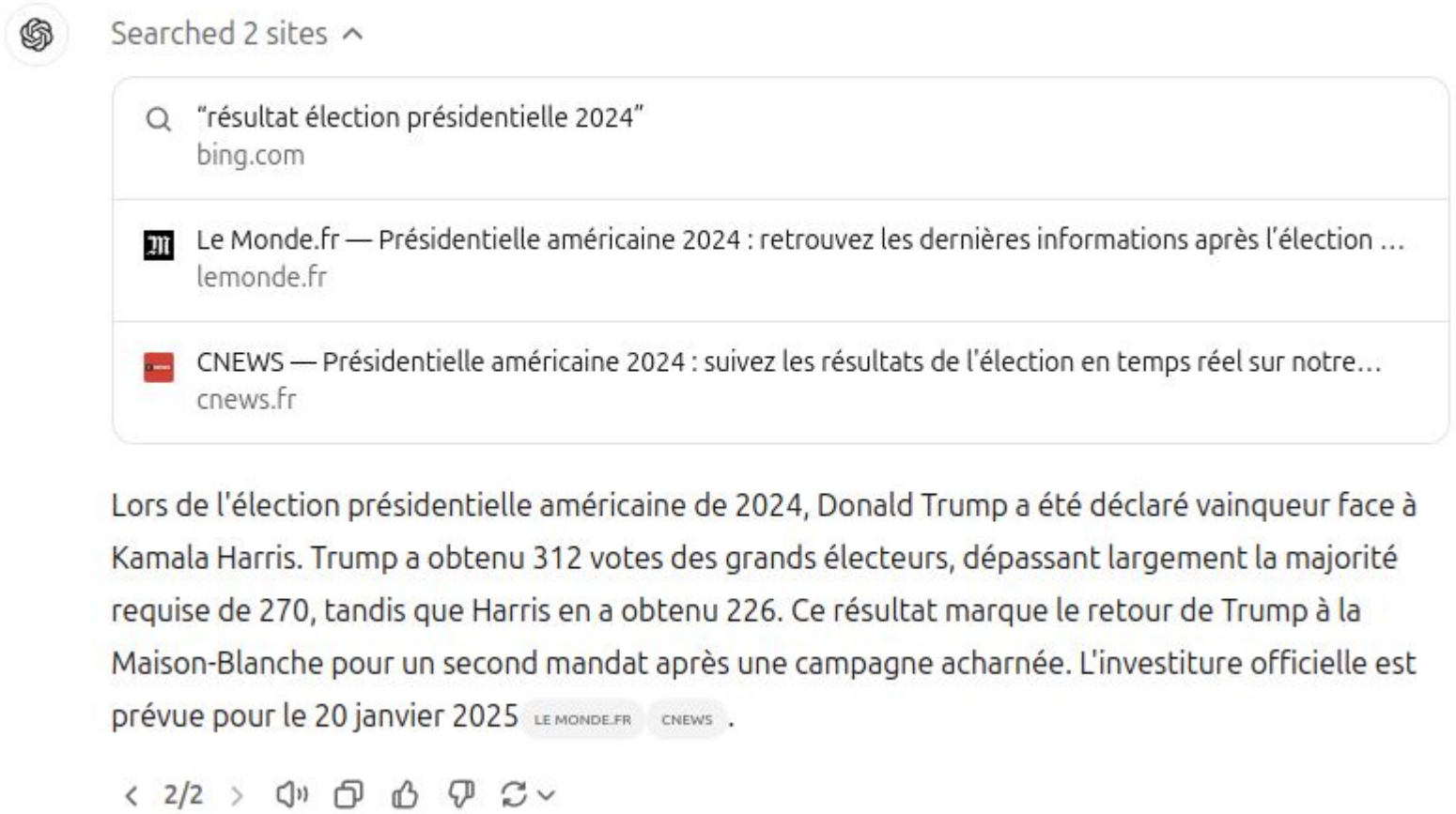
- Pas d’accès aux informations à jour



Grounding with Public Knowledge



Grounding with Public Knowledge



Grounding with Public Knowledge

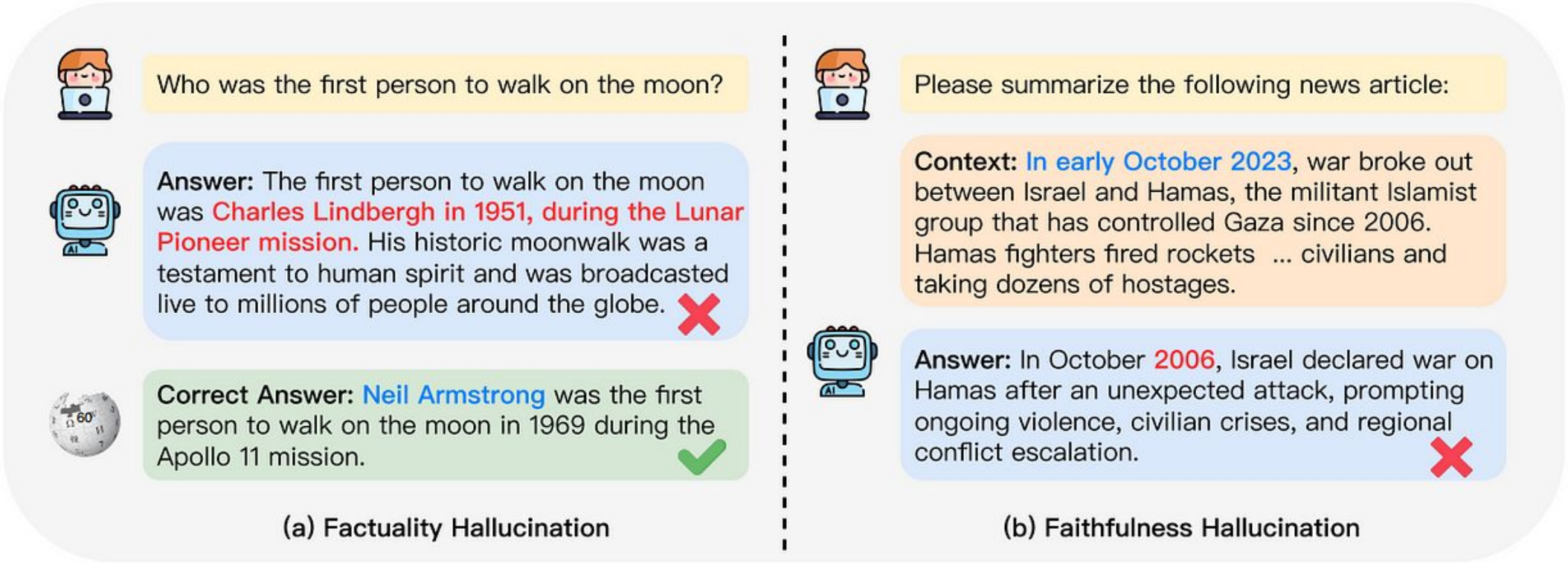


Figure 1: An intuitive example of LLM hallucination.

Grounding with Public Knowledge

GNOMON DIGITAL

(Full) Fine-Tuning

Run fine-tuning to teach the LLM by baking knowledge into the weights of the network.

- Data preparation effort
- Expensive to do
- Online learning/ updating new data
- Still may not work

Make Humans Check

Put a human in-between the output of the LLM and the user.

- Very latent and expensive
- Humans untrustworthy or inattentive

Prompt Engineering

Induce the LLM by adding any known information to the prompt.

- Based on LLMs existing trained knowledge
- Token limit for LLMs
- Error prone: Retrieving the right context
- Trade off: performance, latency, cost

Grounding with Public Knowledge

GNOMON DIGITAL

Which team does Lionel Messi play for?

No retrieval

LLM

As of my knowledge cut-off date in September 2021, Lionel Messi was playing for **PSG.**

With retrieval

Which team does Lionel Messi play for?

Retrieval

"…On 15 July 2023, **Inter Miami** announced the signing of Messi on a two-and-a-half year contract…"

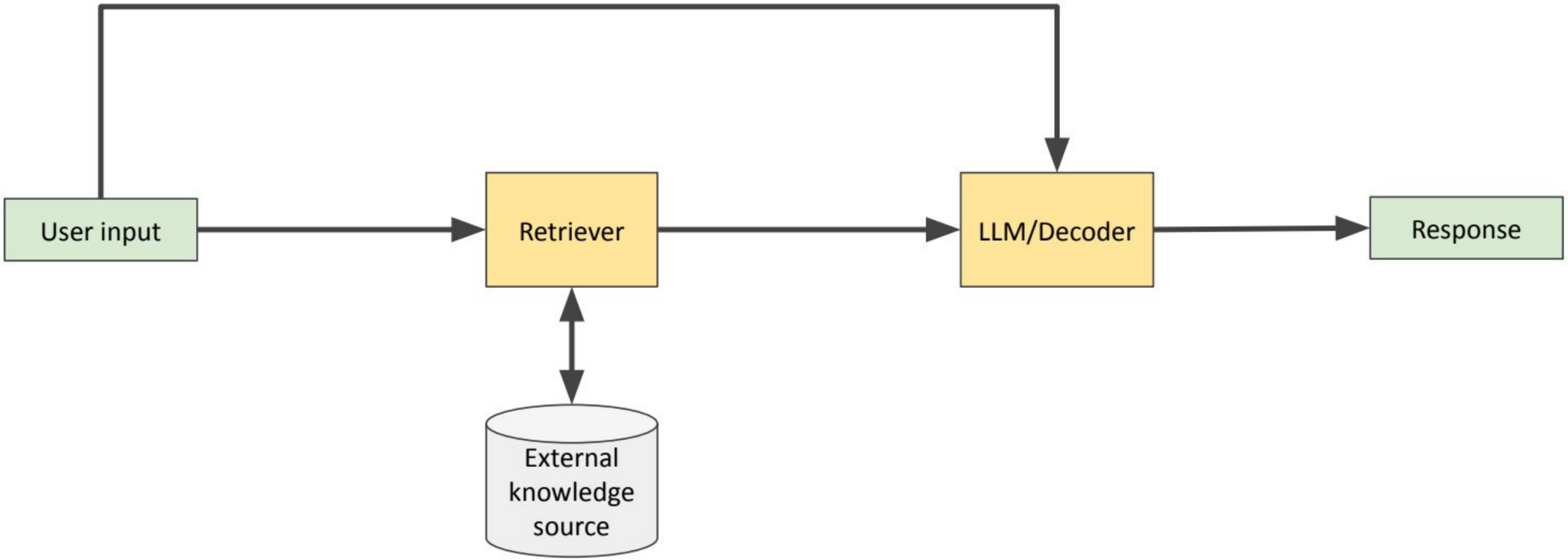
Facts: "On 15 July 2023, Inter Miami announced the signing of Messi on a two-and-a-half year contract"

User query: "Which team does Lionel Messi play for?"

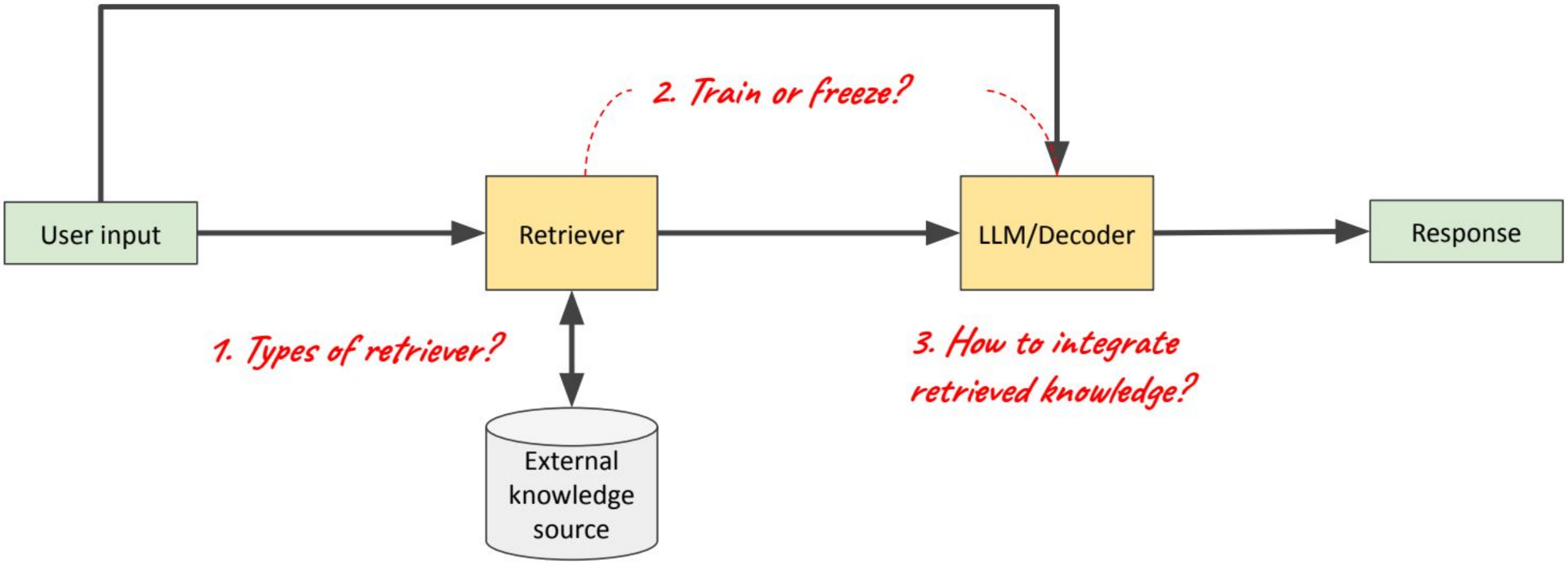
LLM

Lionel Messi plays for Inter Miami.

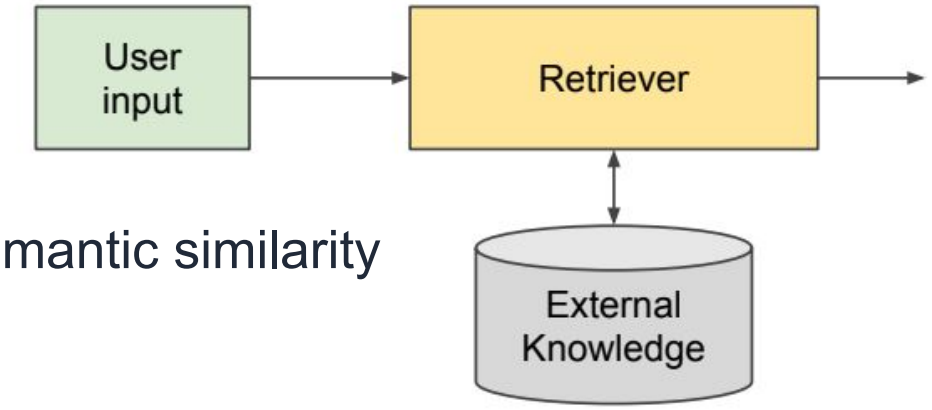
Grounding with Public Knowledge



Grounding with Public Knowledge

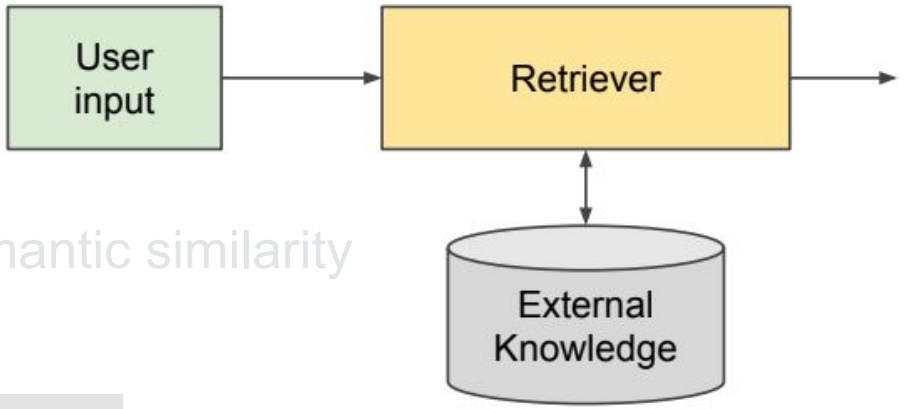


Grounding with Public Knowledge



- Documents non structuré:
  - (textual similarity) TF-IDF / BM25
  - (dense encodings - semantic) Neural retriever AKA semantic similarity
- Knowledge Graphs
  - Graph traversal (e.g. n-hop)
  - Autoregressive path finding
  - GraphRag
- Web:
  - Off-the-shelf search engine for retrieval

Grounding with Public Knowledge



- Documents non structuré:
  - (textual similarity) TF-IDF / BM25
  - (dense encodings - semantic) Neural retriever AKA semantic similarity
- Knowledge Graphs
  - Graph traversal (e.g. n-hop)
  - Au
  - Gr
- Web:
  - Of

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

Grounding with Public Knowledge

GNOMON®
DIGITAL

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	log(2/2) = 0	0	0
Car	1/7	0	log(2/1) = 0.3	0.043	0
Truck	0	1/7	log(2/1) = 0.3	0	0.043
Is	1/7	1/7	log(2/2) = 0	0	0
Driven	1/7	1/7	log(2/2) = 0	0	0
On	1/7	1/7	log(2/2) = 0	0	0
The	1/7	1/7	log(2/2) = 0	0	0
Road	1/7	0	log(2/1) = 0.3	0.043	0
Highway	0	1/7	log(2/1) = 0.3	0	0.043

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

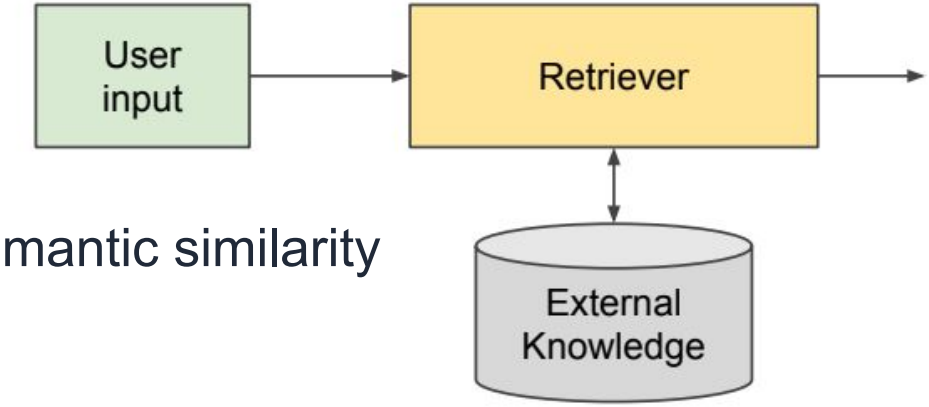
$tf_{x,y}$ = frequency of x in y

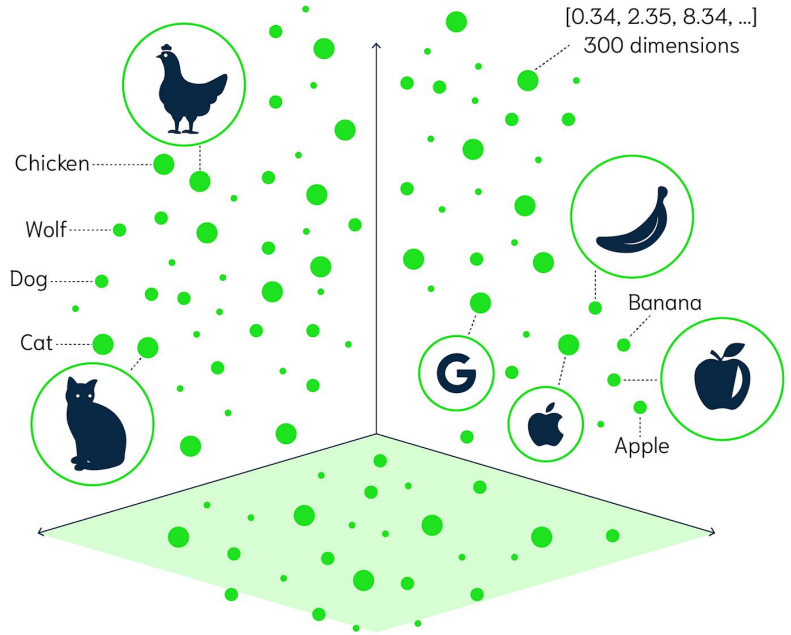
df_x = number of documents containing x

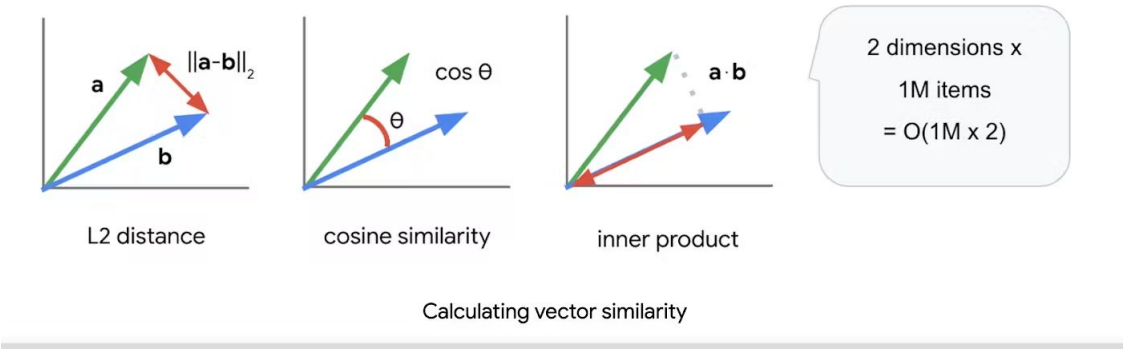
N = total number of documents

Grounding with Public Knowledge

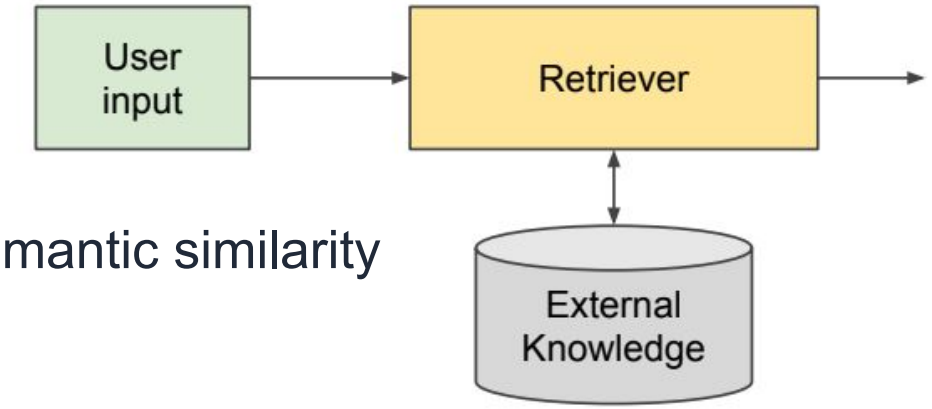
- Documents non structuré:
  - (textual similarity) TF-IDF / BM25
  - (dense encodings - semantic) Neural retriever AKA semantic similarity
- Knowledge Graphs
- W etrieval



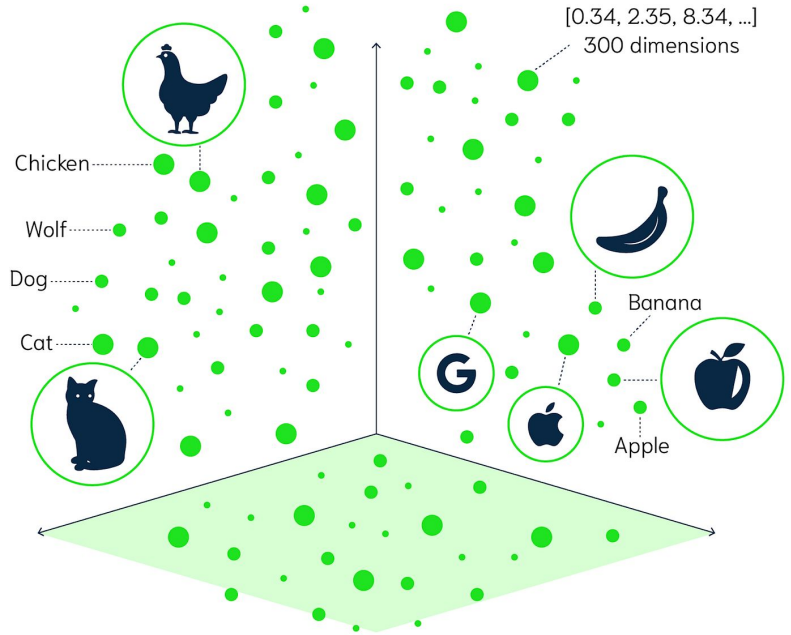


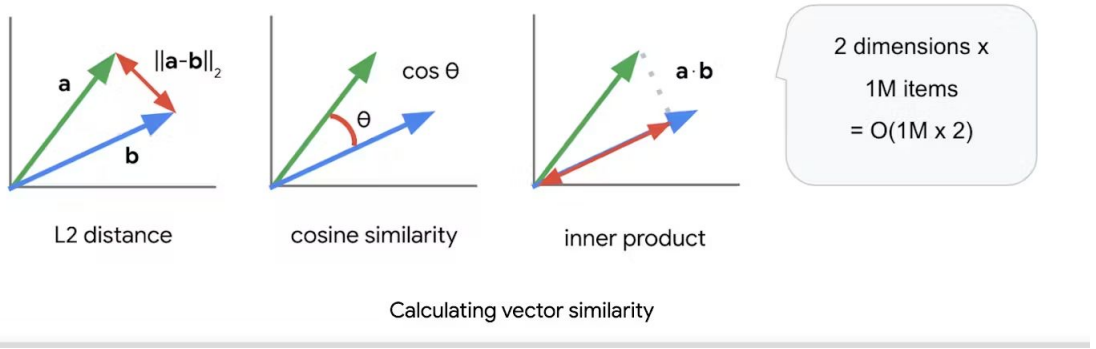


Grounding with Public Knowledge



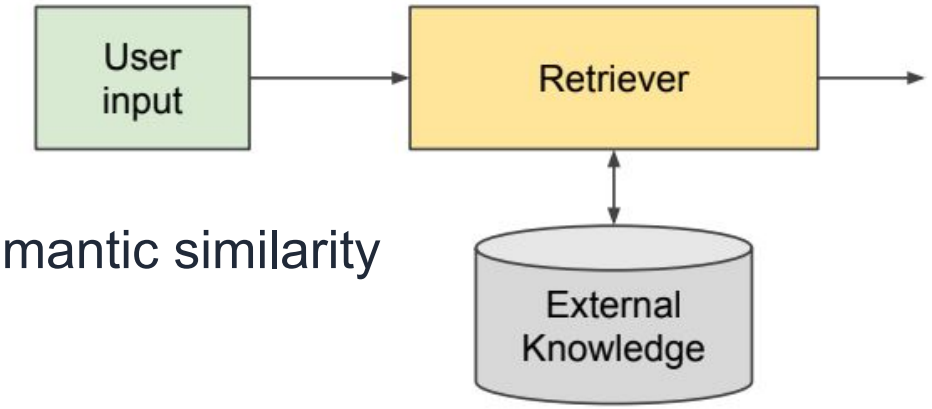
- Documents non structuré:
  - (textual similarity) TF-IDF / BM25
  - (dense encodings - semantic) Neural retriever AKA semantic similarity
- Knowledge Graphs
- W
  - etrieval



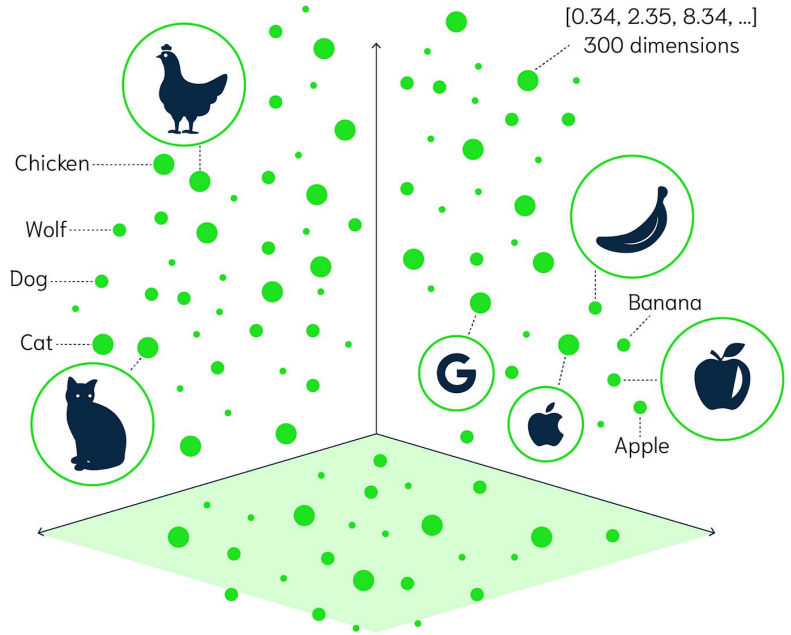


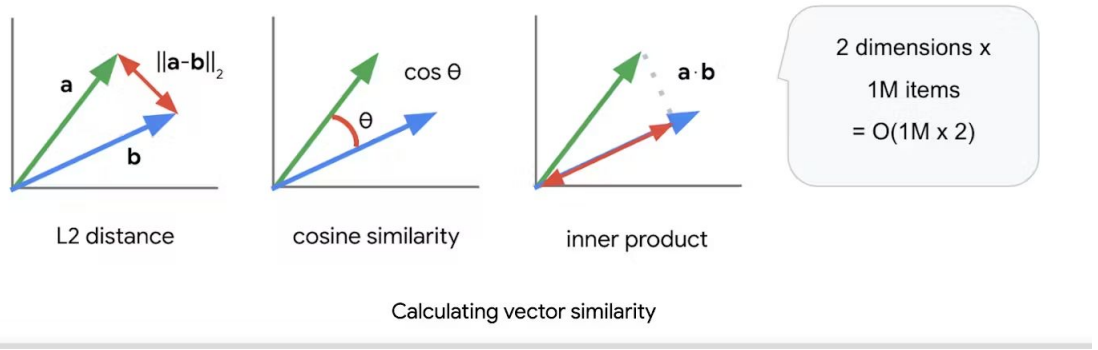
We can train the retriever

Grounding with Public Knowledge



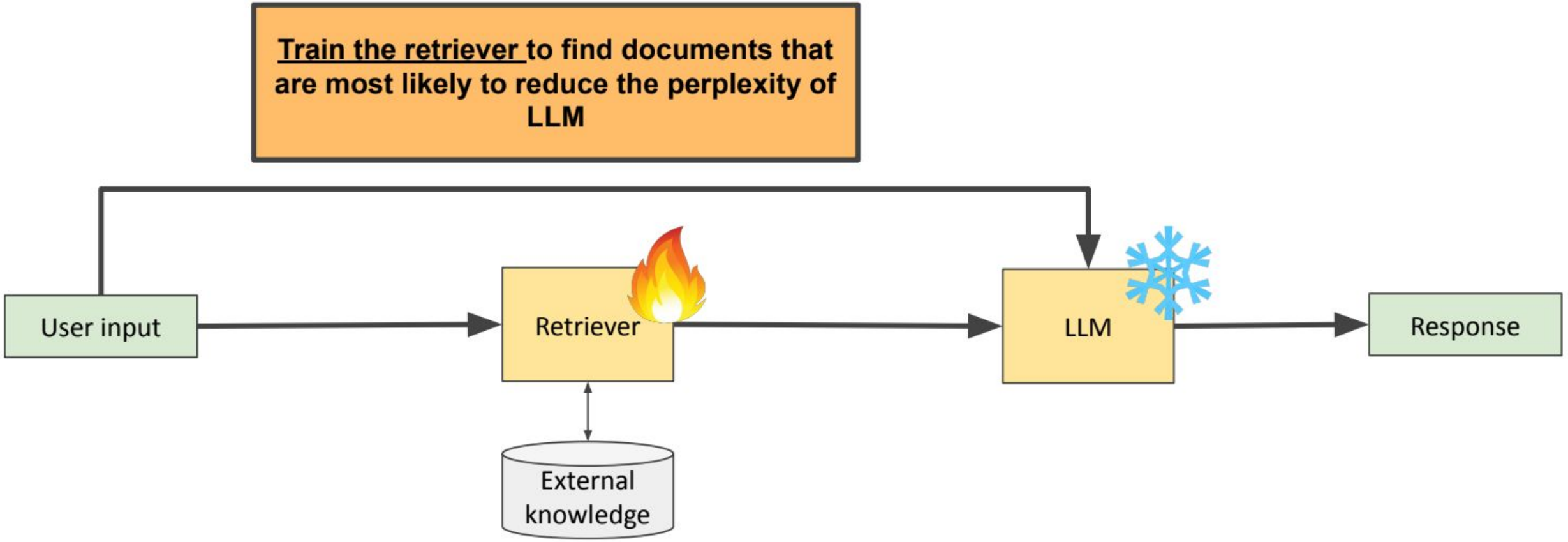
- Documents non structuré:
  - (textual similarity) TF-IDF / BM25
  - (dense encodings - semantic) Neural retriever AKA semantic similarity
- Knowledge Graphs
  -
  -
  -
- W
  - etrieval





We can train the retriever

Grounding with Public Knowledge

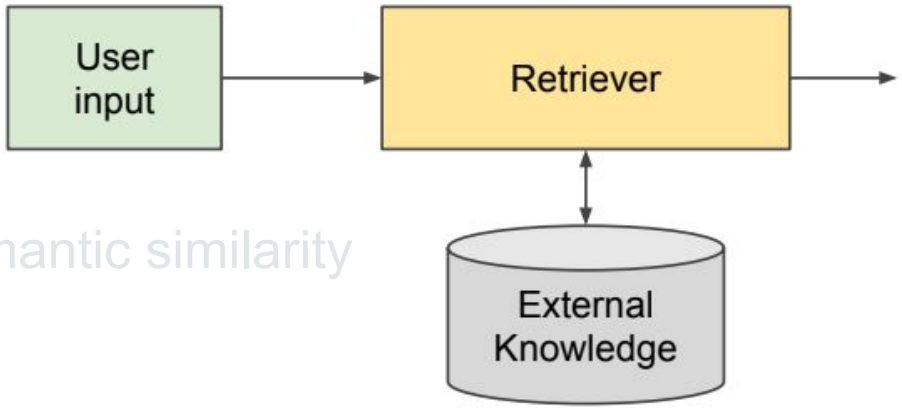


neural network learned from data

$$\text{score}(v_1, v_2) = \frac{1}{1 + e^{-f_\theta(v_1, v_2)}}$$

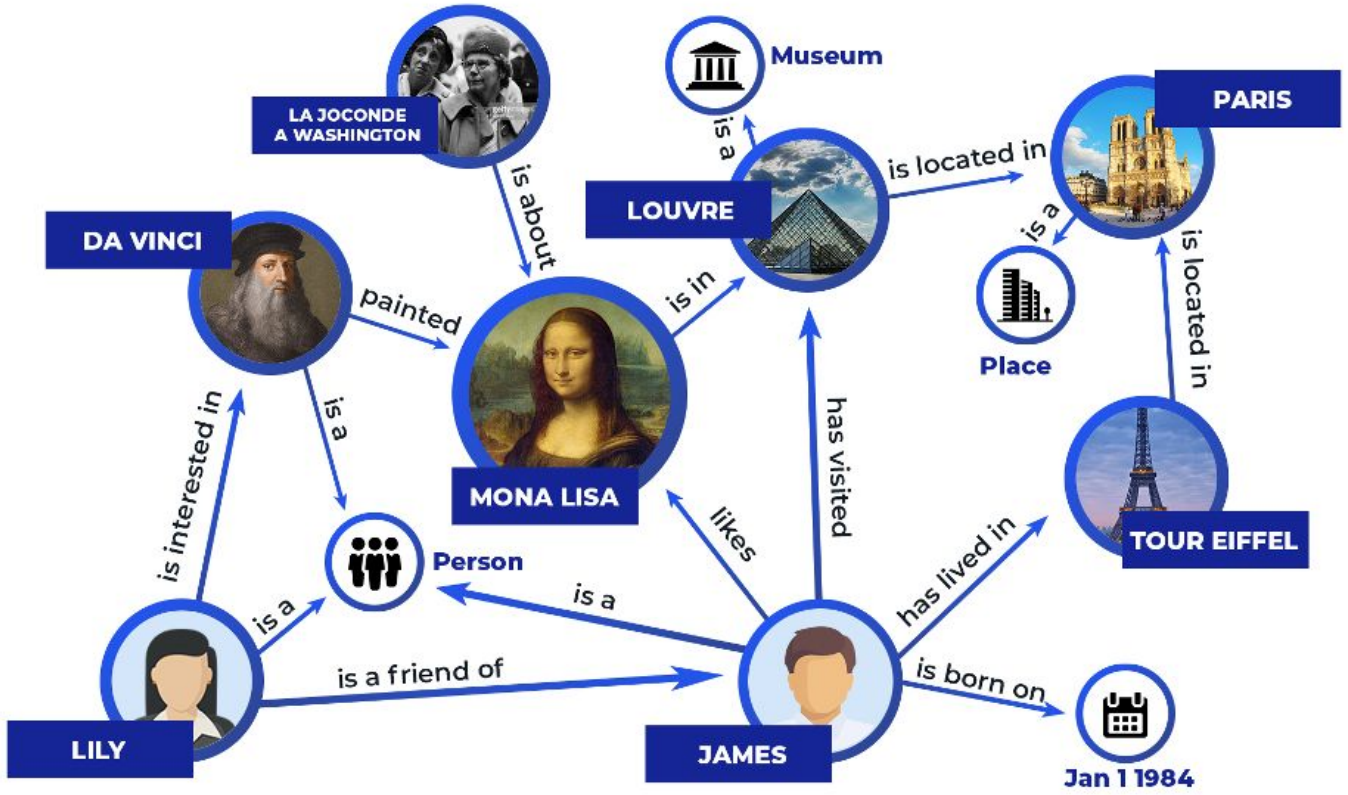
$$(v_1,\ v_2,\ \text{label})$$

Grounding with Public Knowledge



- Documents non structuré:
  - (textual similarity) TF-IDF / BM25
  - (dense encodings - semantic) Neural retriever AKA semantic similarity
- Knowledge Graphs
  - Graph traversal (e.g. n-hop)
  - Autoregressive path finding
  - GraphRag
- Web:
  - Off-the-shelf search engine for retrieval

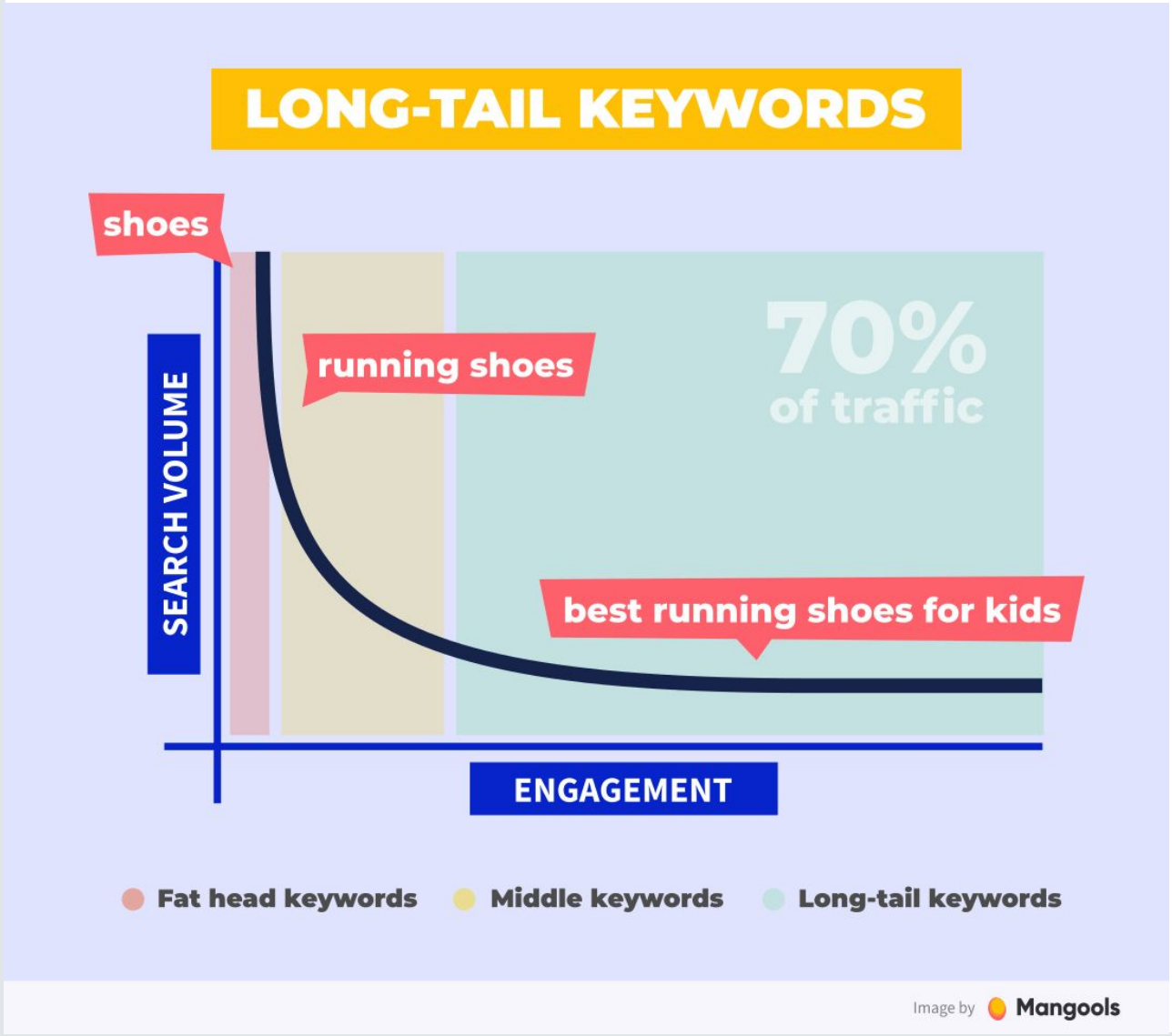
Grounding with Public Knowledge



A knowledge graph is made up of three main components: nodes, edges, and labels. Any object, place, or person can be a node. An edge defines the relationship between the nodes. For example, a node could be a client, like IBM, and an agency like, Ogilvy.

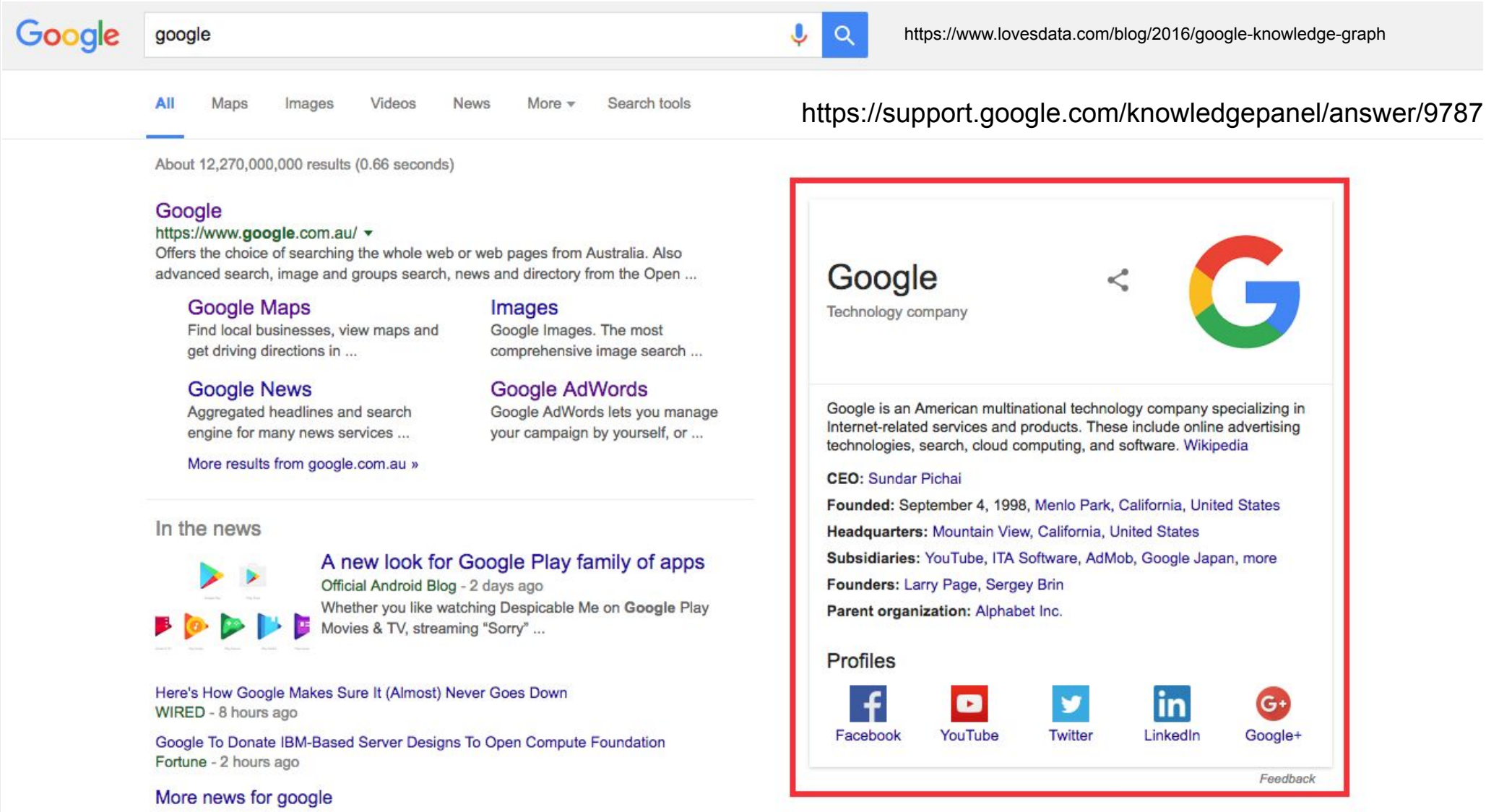
https://www.ibm.com/topics/knowledge-graph

Grounding with Public Knowledge



Long-tail keywords are search queries that get a small number of searches per month. They tend to be longer and more specific than their “head” counterparts and, therefore, often have a higher conversion rate.

Grounding with Public Knowledge



Grounding with Public Knowledge

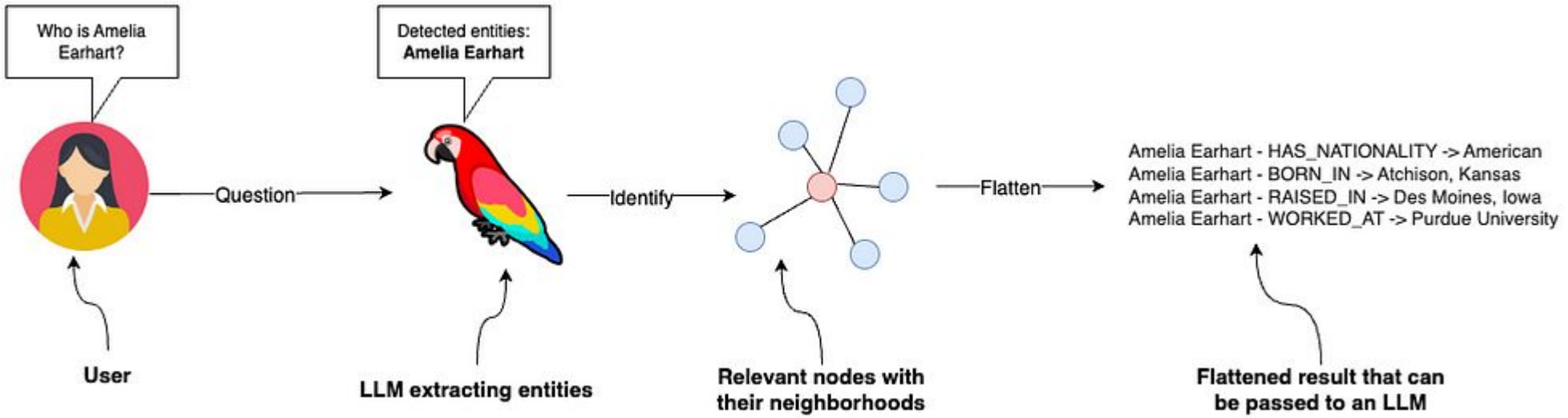
How to query a knowledge graph

Neo4J ←

Cypher	SPARQL
CREATE	INSERT
RETURN	SELECT
WITH	SELECT in a sub query
MATCH	WHERE
WHERE	FILTER
:label	owl:Class or rdfs:Class
[edge]	Predicate <edge> or Triple << <subject> <edge> <object> >>
(node)	a graph node (blank or IRI node)
var:Person	?var a :Person
var.name	?name assuming a match ?var <name> ?name
nodeVar:label {key value, key2 value2}	?nodeVar a label; <key> value;key2 value2.
(ee)-[:KNOWS {since: 2001}]->(js)	<< ?ee <knows> <js> >> <since> 2001.

→ RDF

Grounding with Public Knowledge



Grounding with Public Knowledge

GNOMON DIGITAL

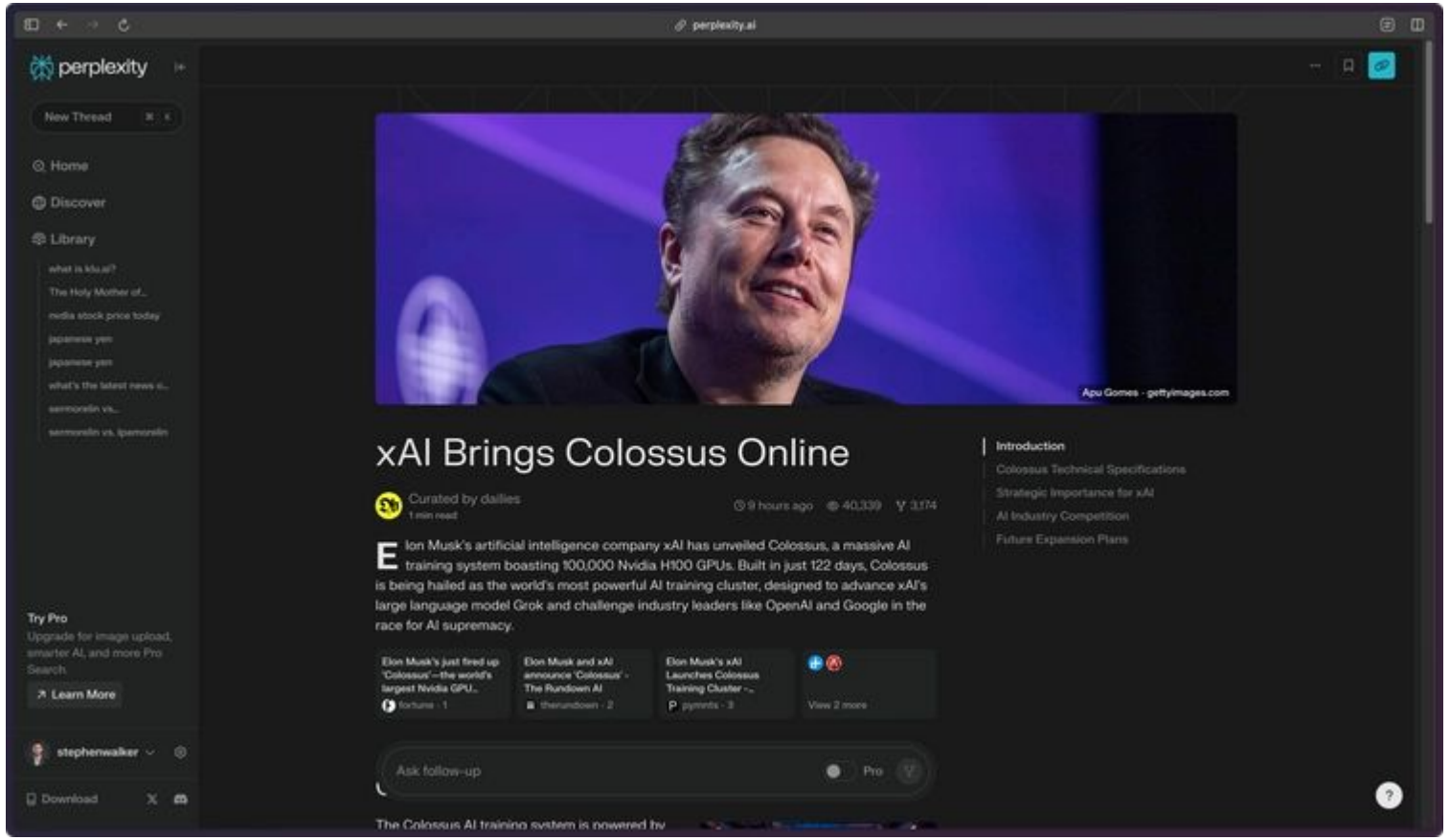
- Documents non structuré:
  - (textual similarity) TF-IDF / BM25
  - (dense encodings - semantic) Neural retriever AKA semantic similarity
- Knowledge Graphs
  - Graph traversal (e.g. n-hop)
  - Autoregressive path finding
  - GraphRag
- Web:
  - Off-the-shelf search engine for retrieval

https://www.perplexity.ai/

Grounding with Public Knowledge

GNOMON DIGITAL

Google search API , Elastic, MongoDB



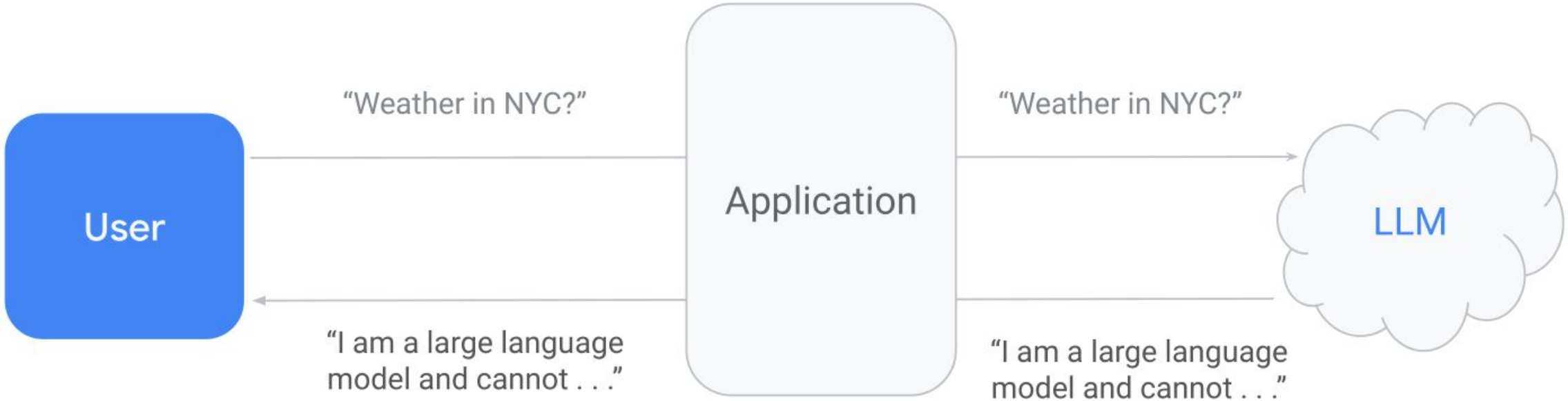
Grounding with Public Knowledge

GNOMON®
DIGITAL

https://github.com/GoogleCloudPlatform/asl-ml-immersion/blob/master/notebooks/vertex_genai/solutions/retrieval_augmented_generation.ipynb

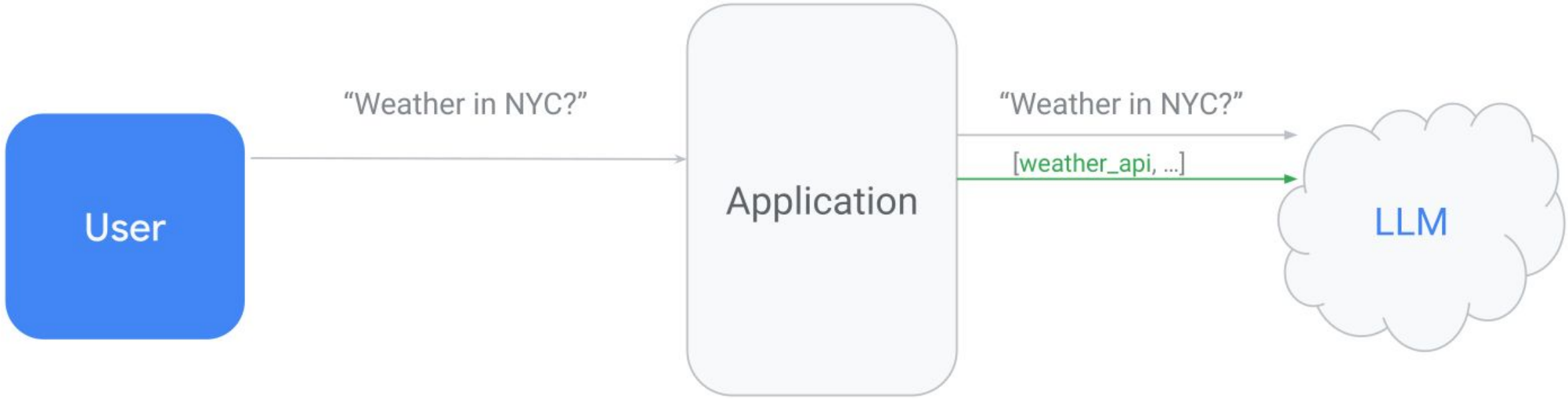
Function Calling

La capacité de connecter de manière fiable les modèles de langage (LLMs) à des outils externes pour permettre une utilisation efficace de ces outils et une interaction avec des API externes.



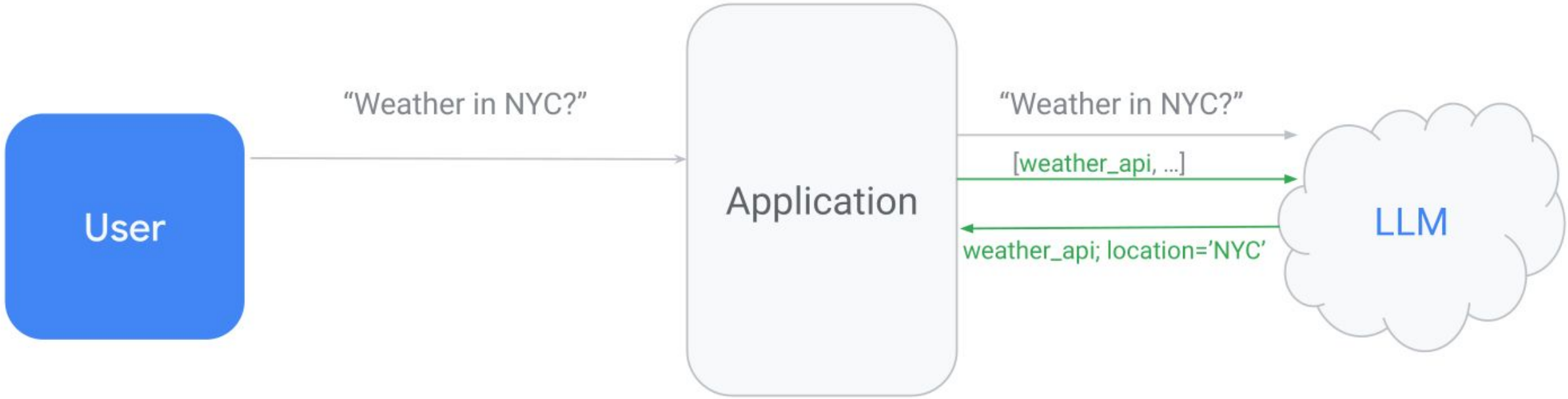
Function Calling

La capacité de connecter de manière fiable les modèles de langage (LLMs) à des outils externes pour permettre une utilisation efficace de ces outils et une interaction avec des API externes.

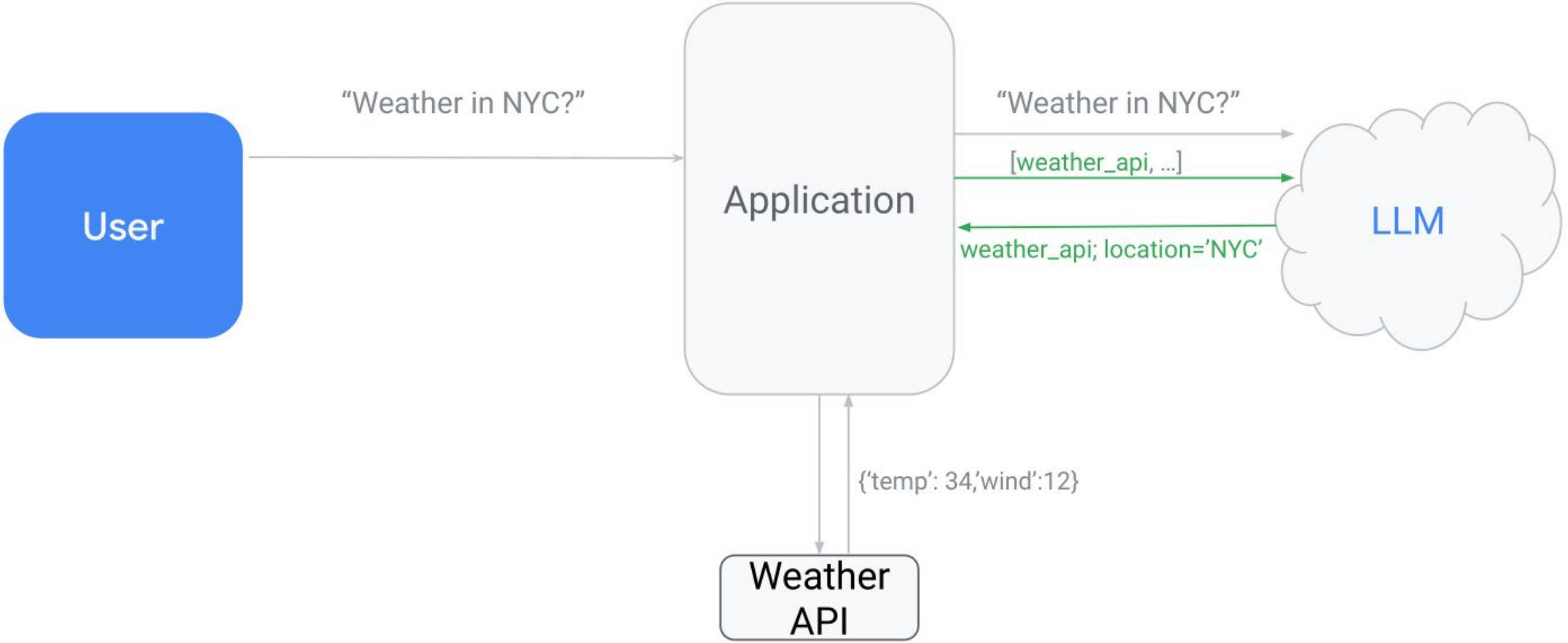


Function Calling

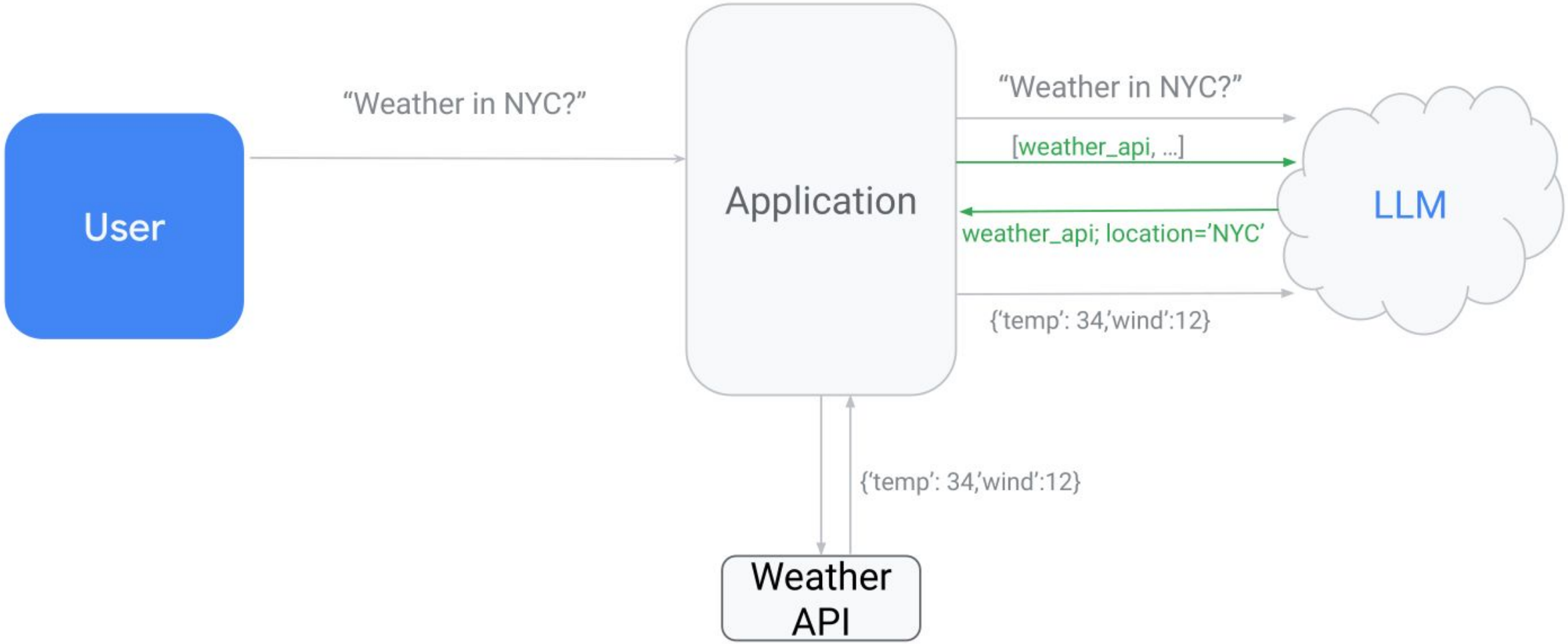
La capacité de connecter de manière fiable les modèles de langage (LLMs) à des outils externes pour permettre une utilisation efficace de ces outils et une interaction avec des API externes.



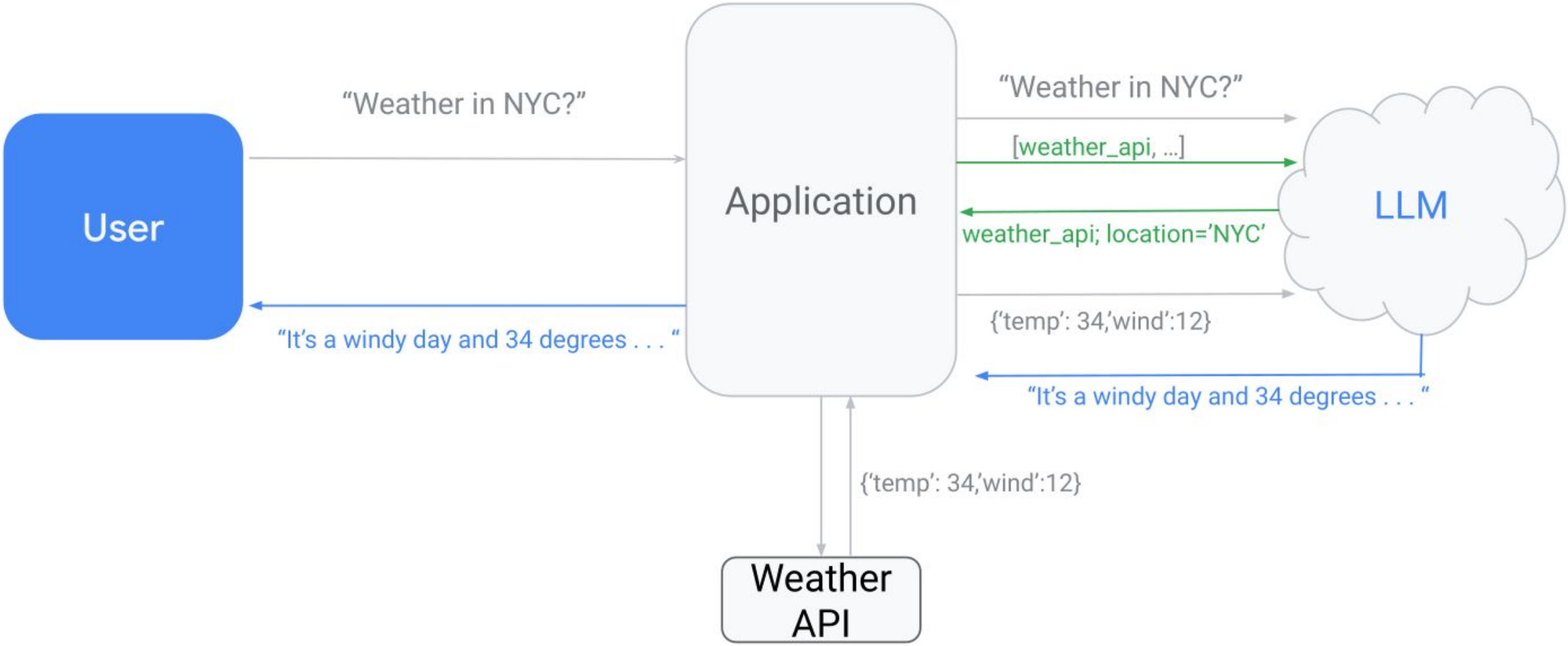
Function Calling



Function Calling



Function Calling



Function Calling

GNOMON DIGITAL

What is the weather like in London?

```
tools = [
    {
        "type": "function",
        "function": {
            "name": "get_current_weather",
            "description": "Get the current weather in a given location",
            "parameters": {
                "type": "object",
                "properties": {
                    "location": {
                        "type": "string",
                        "description": "The city and state, e.g. San Francisco, CA",
                    },
                    "unit": {
                        "type": "string",
                        "enum": ["celsius", "fahrenheit"]},
                },
                "required": ["location"],
            },
        },
    }
]
```

Function Calling

GNOMON DIGITAL

What is the weather like in London?

```
def get_completion(messages, model="gpt-3.5-turbo-1106", temperature=0, max_tokens=300, tools=None):
    response = openai.chat.completions.create(
        model=model,
        messages=messages,
        temperature=temperature,
        max_tokens=max_tokens,
        tools=tools
    )
    return response.choices[0].message
```

Function Calling