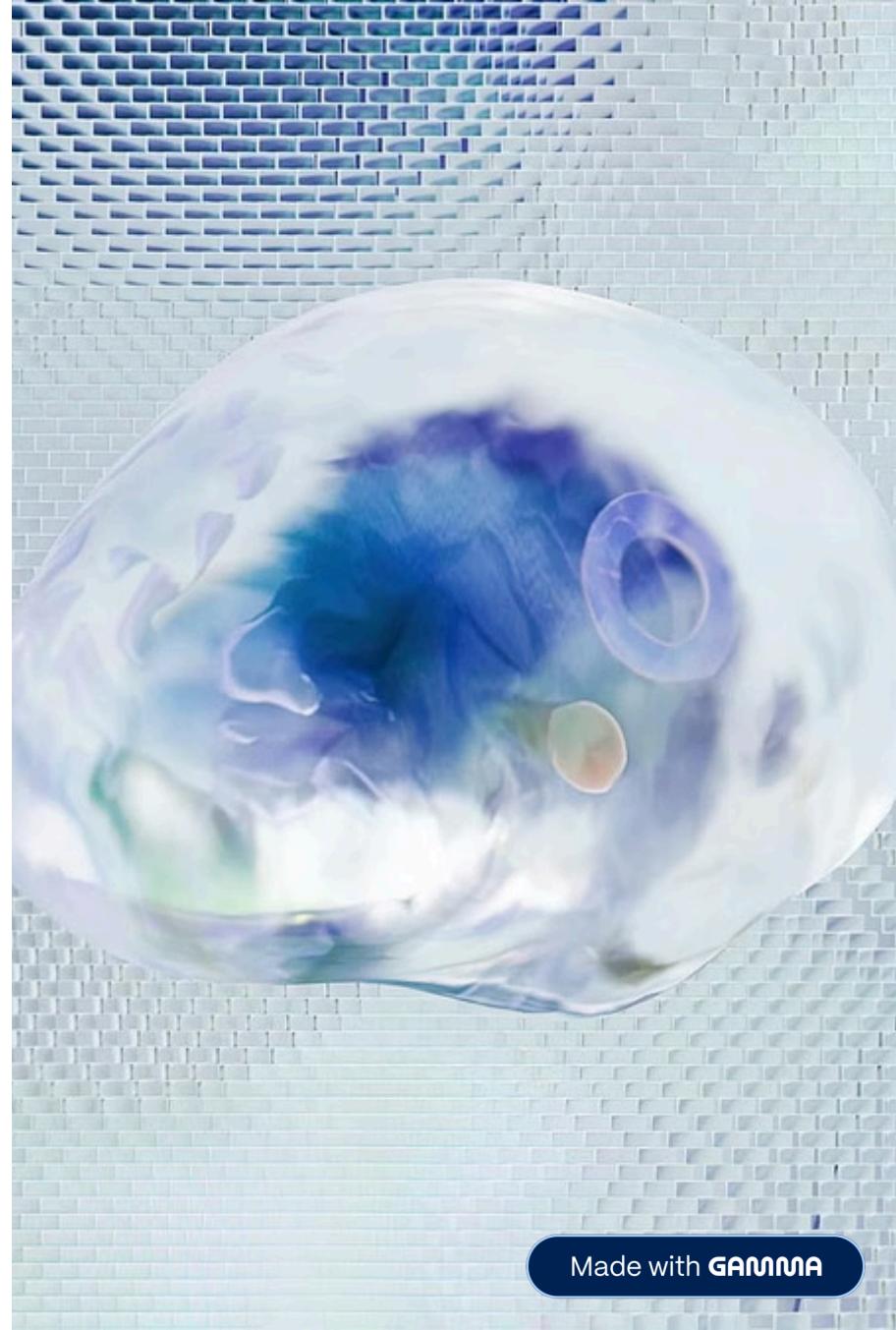


Au-delà de ChatGPT ,Gemini: Comment donner un cerveau et une mémoire à l'IA grâce au **RAG**

IDO EFRAIM

Étudiant à l'École Nationale des Sciences Appliquées de
Berrechid



Introduction : L'IA n'est pas magique, c'est juste des maths

Accroche : Tout le monde utilise ChatGPT, mais qui sait ce qui se passe sous le capot ?

1. L'Analogie du Bébé (L'observation)

Imaginez un bébé qui apprend à parler. Il ne connaît pas vraiment les règles grammaticales ou celles de la conjugaison. Il écoute. Il remarque qu'après "Maman", le mot "vient" apparaît souvent. Il apprend des motifs (*patterns*).

Un LLM (**Large Language Model**), c'est pareil : il a "lu" tout internet pour apprendre quels mots vont ensemble ou comment les agencer pour donner une bonne réponse.

2. L'Analogie du Basketteur (L'entraînement)

Une IA, c'est comme un **basketteur** qui lance un ballon. Il rate (*erreur*). Son coach lui dit : "**trop à gauche.**" Le coach de l'IA, c'est le **GRADIENT**

Le basketteur va ainsi corriger sa posture (Mise à jour des poids). Après des milliards **d'essais**, ce dernier ne rate plus sa cible. C'est la même chose avec l'IA, elle **essaie** de faire des **erreurs**, encore **réessayer**. Après des milliards d'essais, elle finit par être plus performante.



Partie 1 : Le fonctionnement d'un LLM

1. Le cœur : La probabilité conditionnelle

Souvent, on peut entendre le début d'une phrase et connaître déjà la fin. Par exemple, quand quelqu'un dit : « Dans la vie, il y a des hauts. » On sait immédiatement que ce qui suit, c'est : « il y a des bas. » Pourquoi ? Parce qu'on a déjà entendu cette phrase des centaines de fois.

C'est exactement comme ça que fonctionne un Grand Modèle de Langage (LLM) ! Il ne "réfléchit" pas comme nous, il ne comprend pas le sens des mots. Sa seule obsession, c'est de prédire le "token" le plus probable pour compléter une séquence. Un token, c'est un mot ou une partie de mot, comme "s'est" ou "car", un peu comme les atomes du langage.

Le LLM est une gigantesque machine à calculer des probabilités. Il a "lu" une quantité astronomique de textes sur Internet et a mémorisé des milliards de liens statistiques entre ces tokens. Donc, quand tu lui poses une question, il ne fait qu'aligner les tokens les plus probables, un par un, pour former une réponse.

$$P(\text{moose}|\text{Ouedraogo}) = 99.9\%$$

Cette formule mathématique, ça signifie juste : « La probabilité que le mot 'moose' apparaisse après le mot 'Ouedraogo' est de 99.9 %. » Le modèle ne sait pas forcément que le terme "moose" est une ethnie, ni qu'Ouedraogo est un nom de famille. Il a juste vu ces deux mots apparaître ensemble tellement souvent (peut-être dans un document très spécifique et rare sur des communautés au Burkina Faso) qu'il associe l'un à l'autre avec une forte probabilité.

C'est à la fois incroyablement simple et incroyablement puissant, car cette "simple" prédiction de mots permet de générer des textes qui ressemblent étonnamment à ceux écrits par des humains !

2. Le problème : L'hallucination

Conversation

qui est ibrahim traore

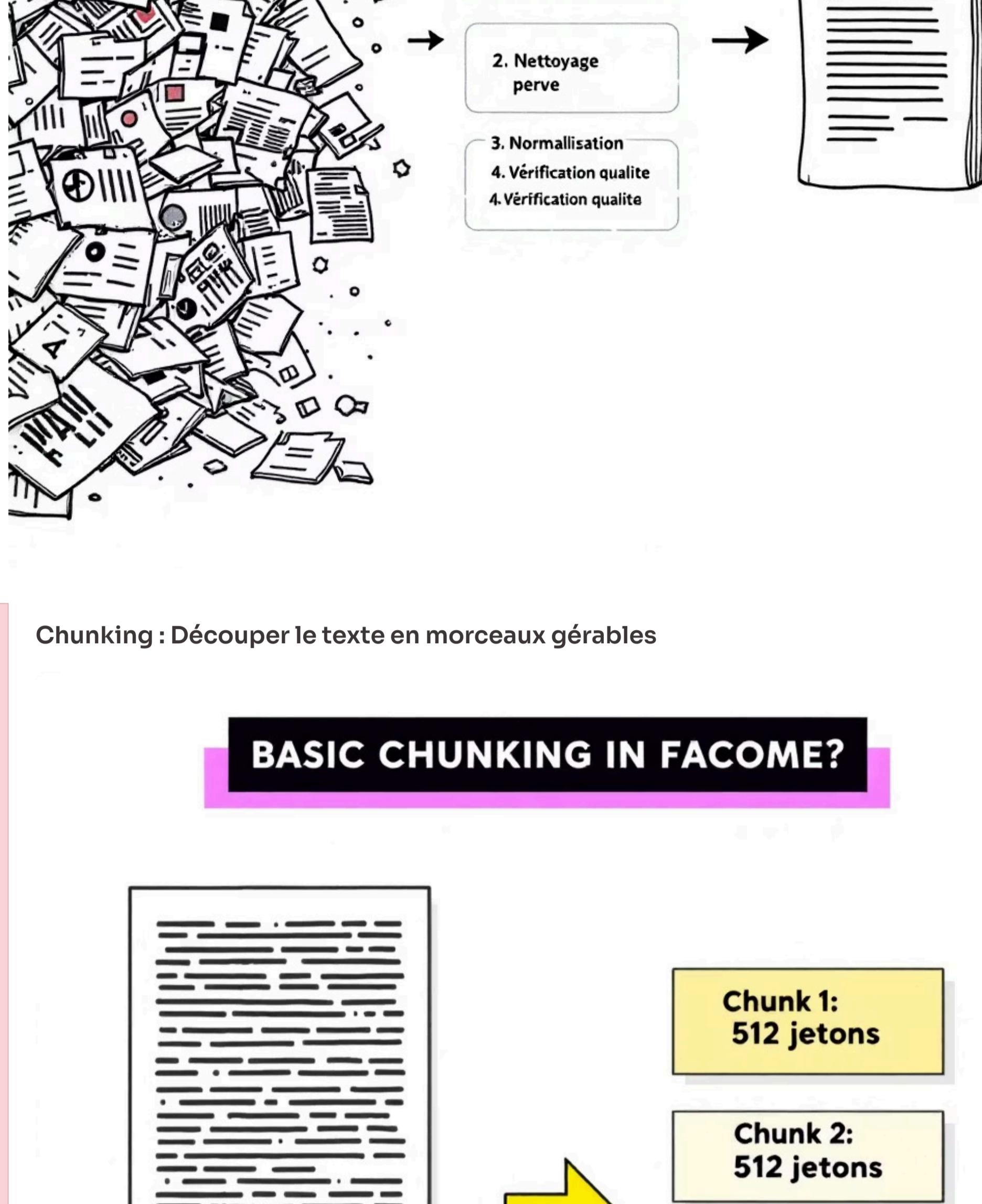
Ibrahim Traoré est un homme politique burkinabè qui a occupé divers postes importants dans son pays, notamment celui de Premier ministre. Il est connu pour avoir inauguré un centre médical d'excellence à Bobo-Dioulasso en décembre 2025, selon l'article "Burkina : Ibrahim Traoré inaugure un centre médical d'excellence à Bobo-Dioulasso (vidéo)" de la Nouvelle Tribune.

Partie 2 : La Solution RAG (Retrieval Augmented Generation)

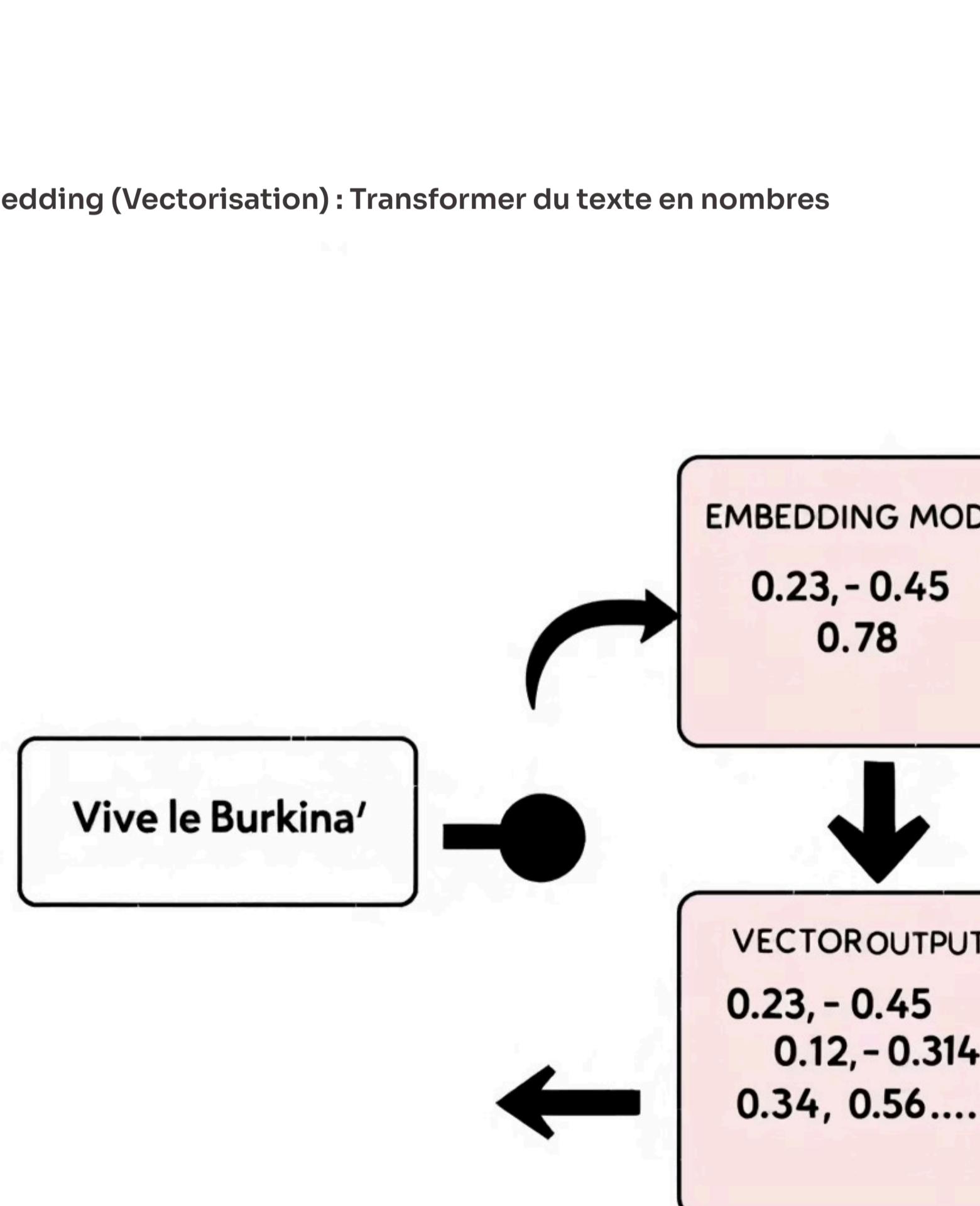
Pour forcer les outils comme CHATGPT à donner des réponses pertinentes, on peut utiliser le RAG. C'est à dire qu'on lui donne des données à jour. On lui donne juste des références.

Etape du RAG

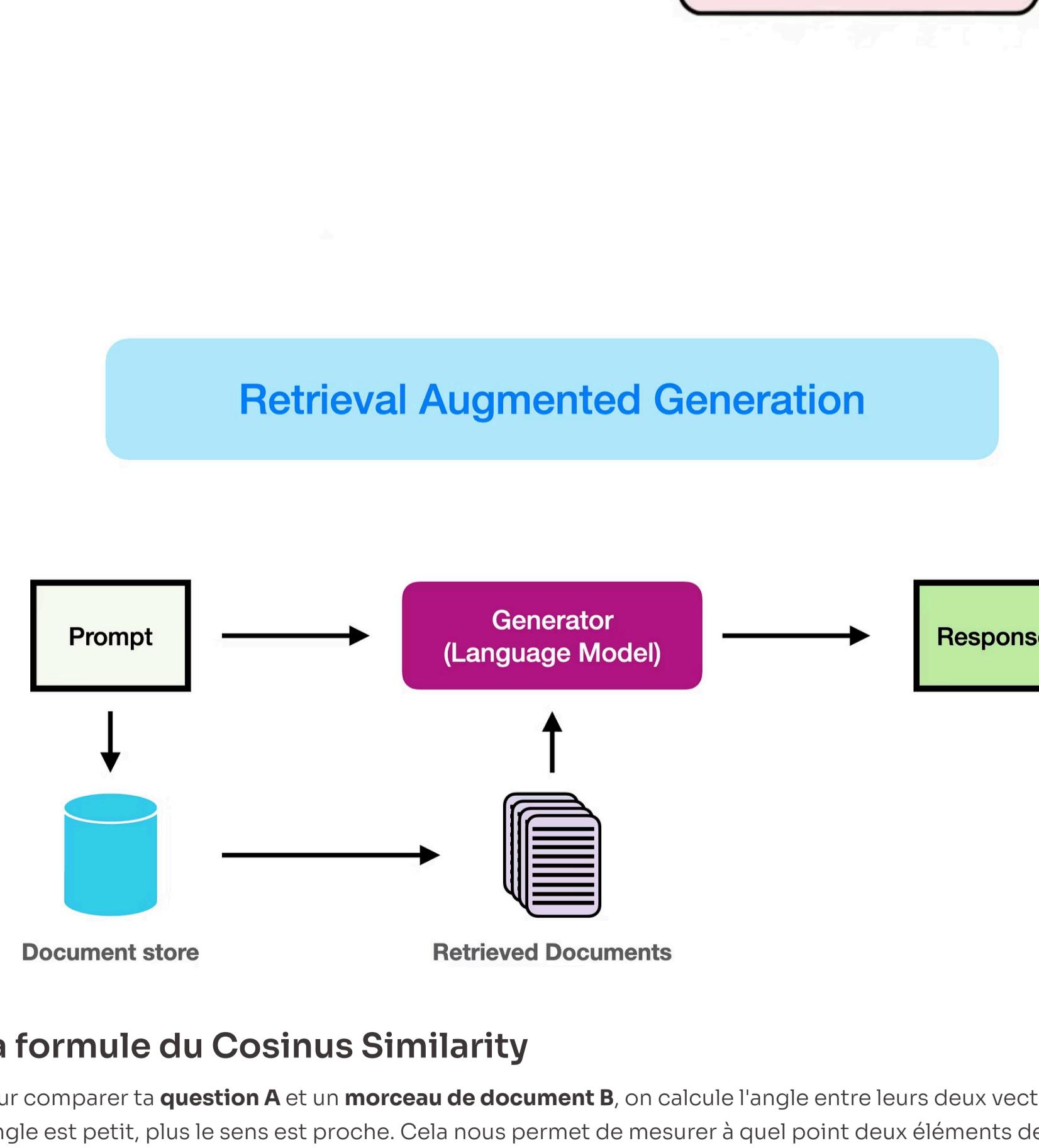
Ingestion : Préparer les données brutes



Chunking : Découper le texte en morceaux gérables



Embedding (Vectorisation) : Transformer le texte en nombres



La formule du Cosinus Similarity

Pour comparer ta **question A** et un **morceau de document B**, on calcule l'angle entre leurs deux vecteurs. Plus l'angle est petit, plus le sens est proche. Cela nous permet de mesurer à quel point deux éléments de texte sont sémantiquement similaires, même si'ils n'utilisent pas les mêmes mots exacts.

$$\text{score} = \frac{A \cdot B}{\|A\| \|B\|}$$

Cette formule calcule le cosinus de l'angle entre les deux vecteurs \$A\$ (votre question) et \$B\$ (le morceau de document). Un score plus élevé indique une plus grande similarité sémantique.

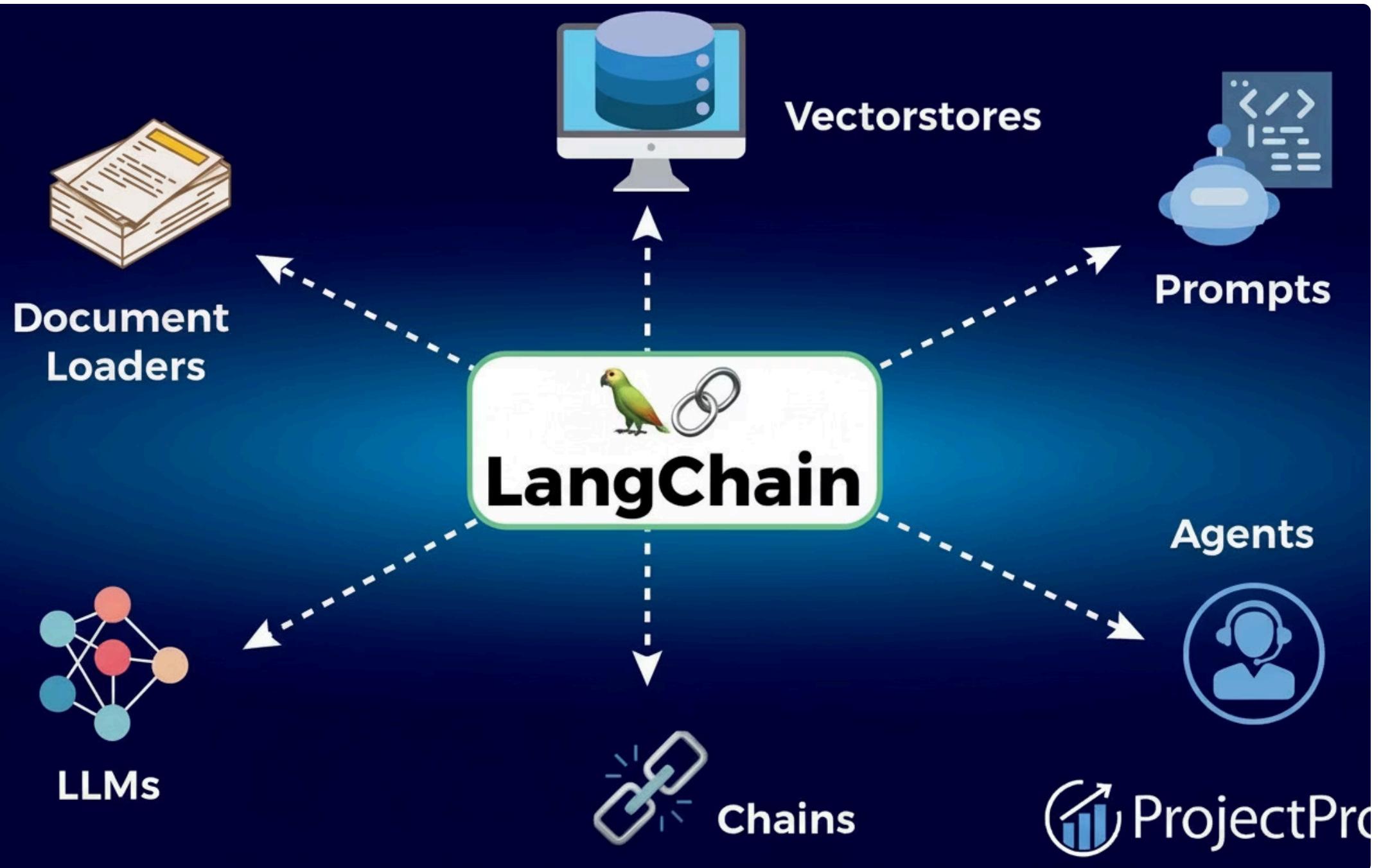
Interprétation des résultats

- **1**: Les deux vecteurs sont identiques (même direction, même sens), indiquant une parfaite similarité.
- **0**: Ils n'ont aucun rapport (ils sont perpendiculaires), signifiant aucune similarité sémantique.
- **-1**: Ils sont opposés, indiquant une forte opposition (contraires).

Pourquoi on l'utilise ?

Dans notre système RAG, on ne cherche pas des mots identiques, on cherche des idées proches.

Si ta question pointe vers le "Nord" et qu'un document pointe aussi vers le "Nord", le cosinus sera proche de 1. C'est ce document que le système va "pecher" (Retrieval) pour donner la réponse.





Partie 3 : La démo avec LangChain

Je veux aussi coder en live pour leur montrer comment ça se passe dans la vraie vie...

Pourquoi LangChain ?

C'est le "framework" qui permet d'assembler ces briques (Lego pour développeurs IA). Il gère la connexion entre le PDF, la base vectorielle et le LLM.

Scénario de Démo

Source : Un document "sérieux" (ex: article scientifique sur les trous noirs).

L'erreur : Question pointue au LLM brut -> Il échoue.

Le RAG : Import, Splitter et interrogation.

```
# Aperçu du code Python
from langchain.vectorstores import Chroma
from langchain.llms import OpenAI

# Chargement et interrogation
doc = loader.load("cours_prep.pdf")
query = "Quelle est la limite de la fonction f?"
response = rag_chain.invoke(query)
print(response)
```



Conclusion : Esprit Critique & Limites

1. Ce n'est pas magique

Garbage In, Garbage Out : Si le PDF source est mauvais, la réponse sera mauvaise.

Fenêtre de contexte : On ne peut pas tout mettre, il faut choisir les bons "chunks".

2. Bonnes pratiques

Toujours demander au modèle de citer ses sources.

Utiliser le "Prompt Engineering" : "Si tu ne trouves pas, dis 'Je ne sais pas', n'invente rien."

3. Ouverture

Le métier d'ingénieur de demain, c'est architecturer ces systèmes. Les maths (algèbre linéaire, probabilités) sont la clé de tout ça.



Conseils pour l'orateur



« C'est de la géométrie dans un espace à 1500 dimensions. Si deux vecteurs ont un angle faible (cosinus proche de 1), alors les idées sont sémantiquement proches. »

Mot de fin

Fin de la présentation – Prêt pour le Live Coding !