NLP
IA GÉNÉRATIVE & LLM

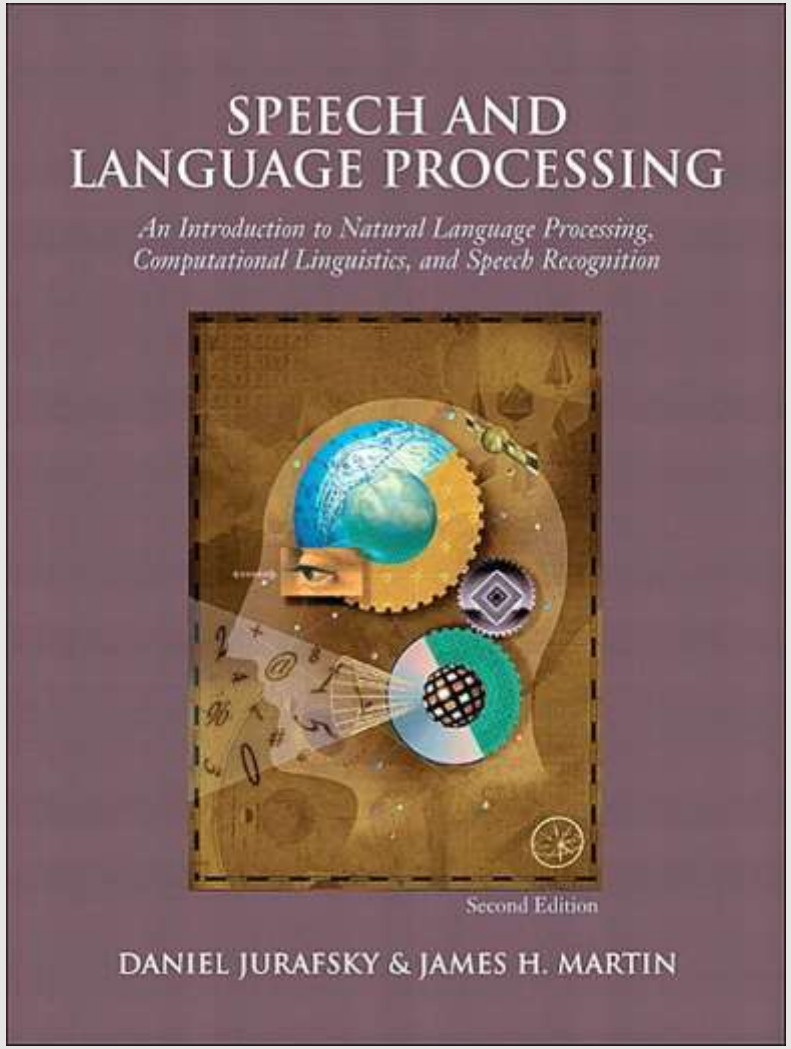
MASTER 2 SISE

LES GRANDS MODÈLES DE LANGAGES

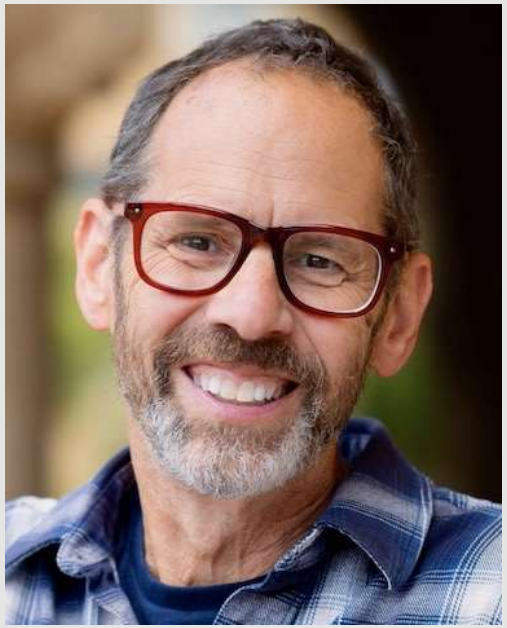
PREMIÈRE SECTION

LES MODÈLES DE LANGAGE

SPEECH & LANGUAGE PROCESSING



https://web.stanford.edu/~jurafsky/slp3/



LE TOKEN

« Je donne un cours sur les grands modèles de langage aux étudiants du Master 2 SISE »

```
['Je', ' donne', ' un', ' cours', ' sur', ' les', '
grands', ' mod', 'èles', ' de', ' lang', 'age', '
aux', ' ét', 'udi', 'ants', ' du', ' Master', ' ',
'2', ' S', 'ISE']
```

BYTE-PAIR ENCODING

```
"hug", "pug", "pun", "bun", "hugs"
```

```
["b", "g", "h", "n", "p", "s", "u"]
```

("h" **"u" "g",** 10), ("p" **"u" "g",** 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" **"u" "g"** "s", 5)

```
["b", "g", "h", "n", "p", "s", "u","ug“]
```

("h" "ug", 10), ("p" "ug", 5), ("p" **"u" "n",** 12), ("b" **"u" "n",** 4), ("h" "ug" "s", 5)

```
["b", "g", "h", "n", "p", "s", "u","ug", "un"]
```

(**"h" "ug",** 10), ("p" "ug", 5), ("p" "un", 12), ("b" "un", 4), (**"h" "ug"** "s", 5)

```
["b", "g", "h", "n", "p", "s", "u", "ug", "un", "hug"]
```

("hug", 10), ("p" "ug", 5), ("p" "un", 12), ("b" "un", 4), ("hug" "s", 5)

https://huggingface.co/docs/transformers/tokenizer_summary#byte-pair-encoding-bpe

NEXT-WORD PREDICTION

$$P(w_t | w_1, w_2, \ldots, w_{t-1})$$

$$\hat{w}_t = \arg\max_w P(w | w_1, w_2, \ldots, w_{t-1})$$

« La basilique au dessus de Lyon est »

N-GRAM

$$P(w_t|w_1, w_2, \ldots, w_{t-1})$$

Markov Assumption: Approximation en prenant en compte seulement les n mots précédents

« La basilique au dessus de Lyon est »

Bi-gram: P(Fourvière | est)
Tri-gram: P(Fourvière | Lyon est)

N-GRAM

Occurrence du Bi-gram

$$P(w_t|w_{t-1}) = \frac{\text{Count}(w_{t-1}, w_t)}{\text{Count}(w_{t-1})}$$

Occurrence du/des token(s) précédent(s)

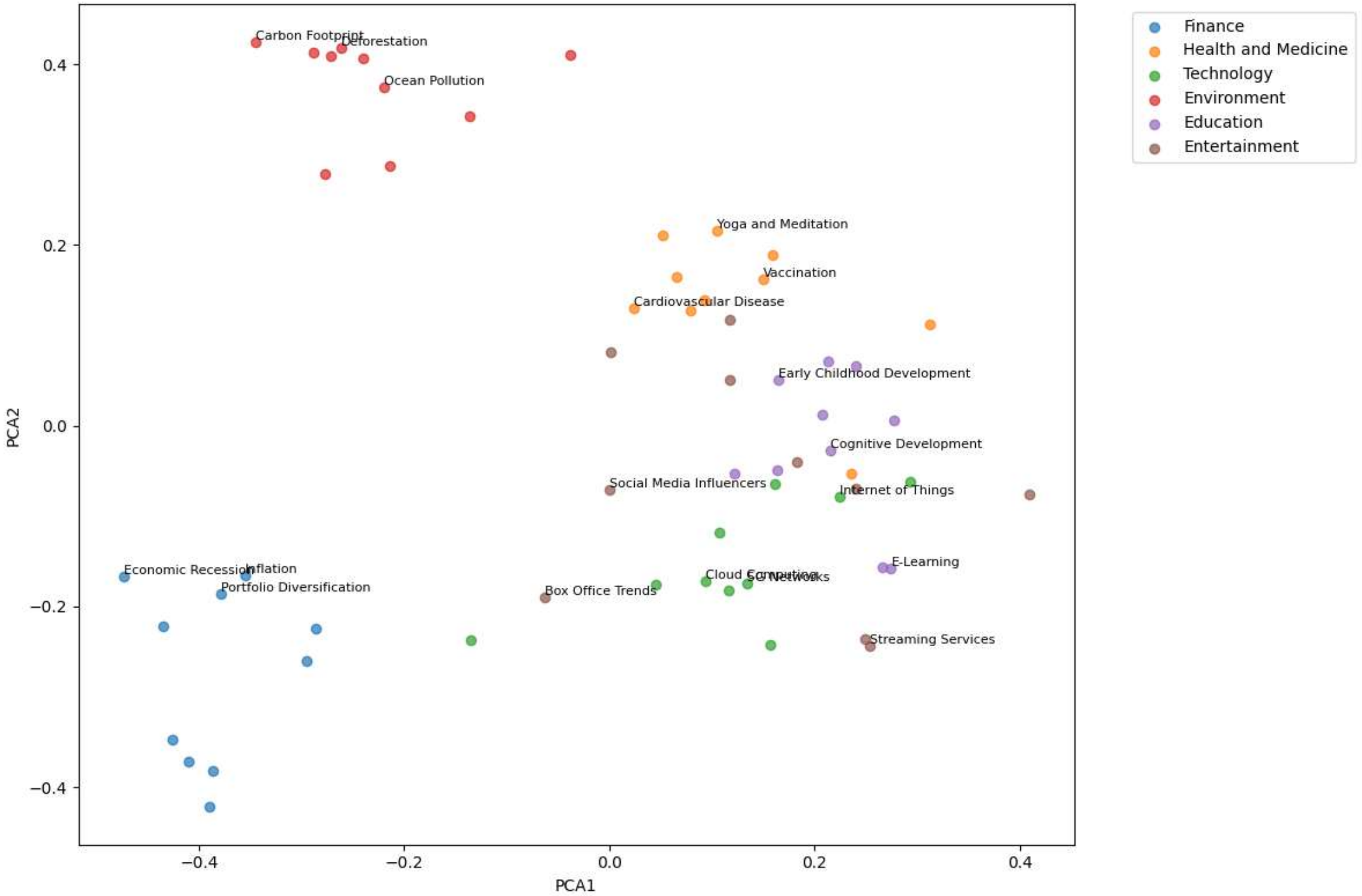
	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

DISTRIBUTIONAL HYPOTHESIS

Les mots qui apparaissent **dans des contextes similaires** tendent à avoir des **significations similaires**.

"Le **sprinteur** est explosif sur de **courtes distances**."
"Le **marathonien** doit maintenir un **rythme** de **course** soutenu pendant des **heures** sur de longue **distance**."
"Les qualités de course d'un **sprinteur** diffèrent de celles d'un **marathonien**, l'un favorisant la **vitesse**, l'autre **l'endurance**."

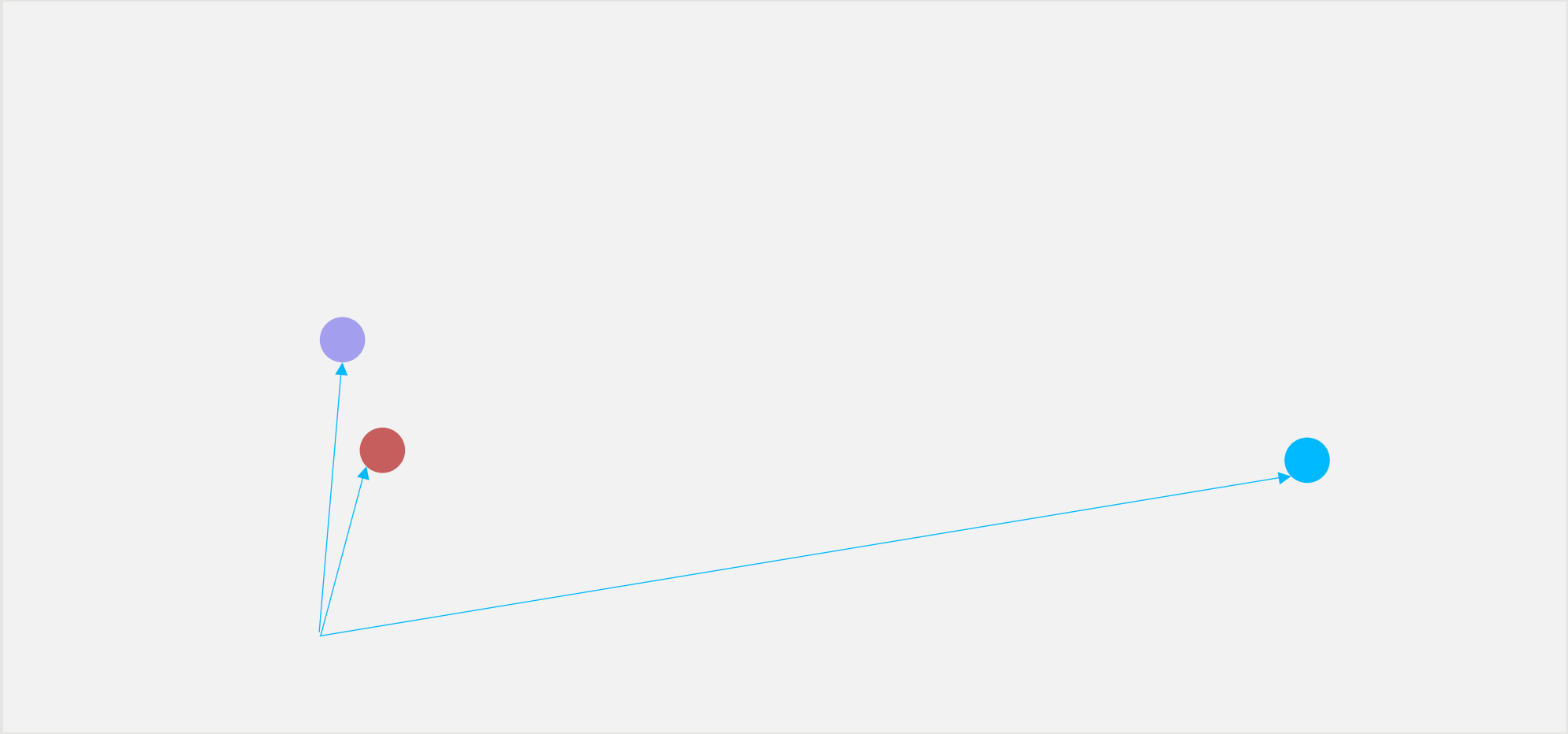
Sport,Temps, Distance, Course, Vitesse, Endurance


Finance
Health and Medicine
Technology
Environment
Education
Entertainment
Carbon Footprint
Deforestation
Ocean Pollution
Yoga and Meditation
Vaccination
Cardiovascular Disease
Early Childhood Development
Cognitive Development
Social Media Influencers
Internet of Things
E-Learning
Economic Recession
Inflation
Portfolio Diversification
Box Office Trends
Cloud Computing
5G Networks
Streaming Services
PCA1
PCA2
0.4
0.2
0.0
−0.2
−0.4
−0.4
−0.2
0.0
0.2
0.4

SIMILARITY

$$\text{cosine}(\mathbf{v},\mathbf{w}) = \frac{\mathbf{v}\cdot\mathbf{w}}{|\mathbf{v}||\mathbf{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

SIMILARITY



TF-IDF

Problème avec la co-occurrence: **stop-words**

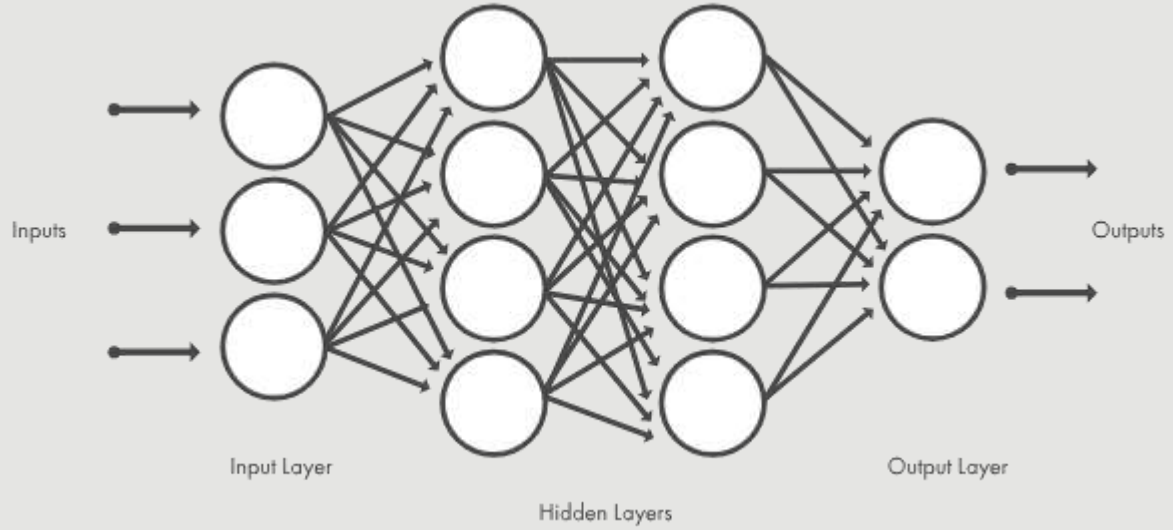
$$tf_{t,d} = \frac{\text{number of } t \text{ in } d}{\text{total number of terms in } d}$$

$$tfidf_{t,d} = tf_{t,d} \cdot idf_t$$

$$idf_t = log \frac{\text{total number of documents}}{\text{number of documents with } t}$$

WORD2VEC: SPARSE TO DENSE

Static embedding: Vecteur d'un mot indépendant de son contexte



L'IMPORTANCE DU CONTEXTE

« J'ai essayé un **Random Forest**, mais **il** n'est pas performant sur ces données »

« J'ai voulu demander à **mon collègue** de m'aider, mais **il** n'était pas à son bureau»

«En été, j'adore aller à la **pêche**»

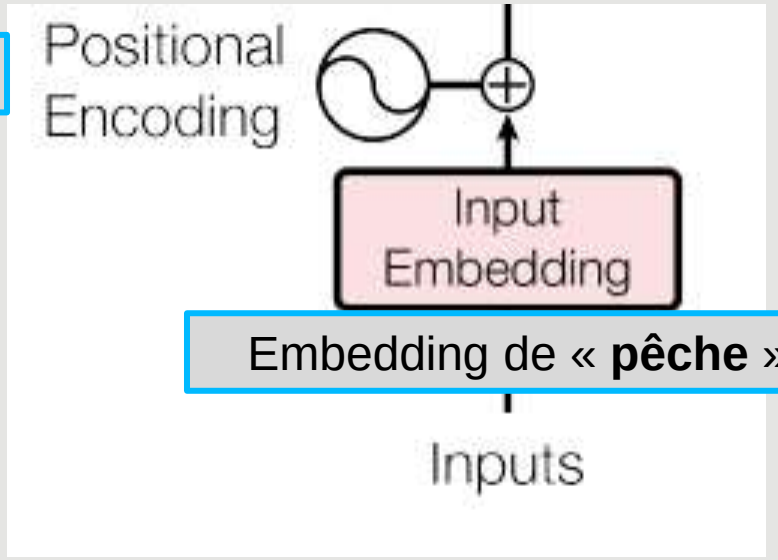
«Je mange une **pêche**»

Le sens des mots changent selon le contexte !

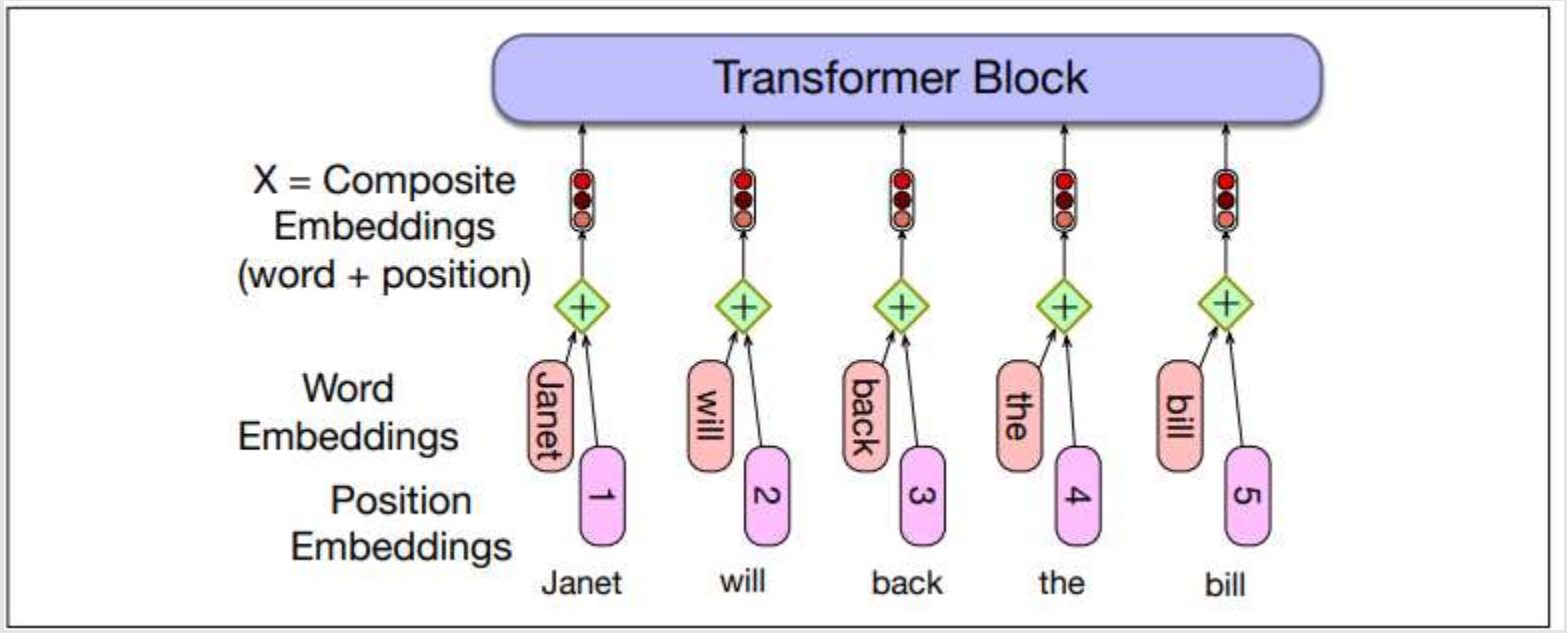
INPUT

- Un embedding **du token** **(indépendant de la position dans la phrase)**
- Un embedding de la **position** du token

Embedding de la position **8**



«En été, j'adore aller à la **pêche**»



ATTENTION

Contextual Embeddings

«Je mange une **pêche**»

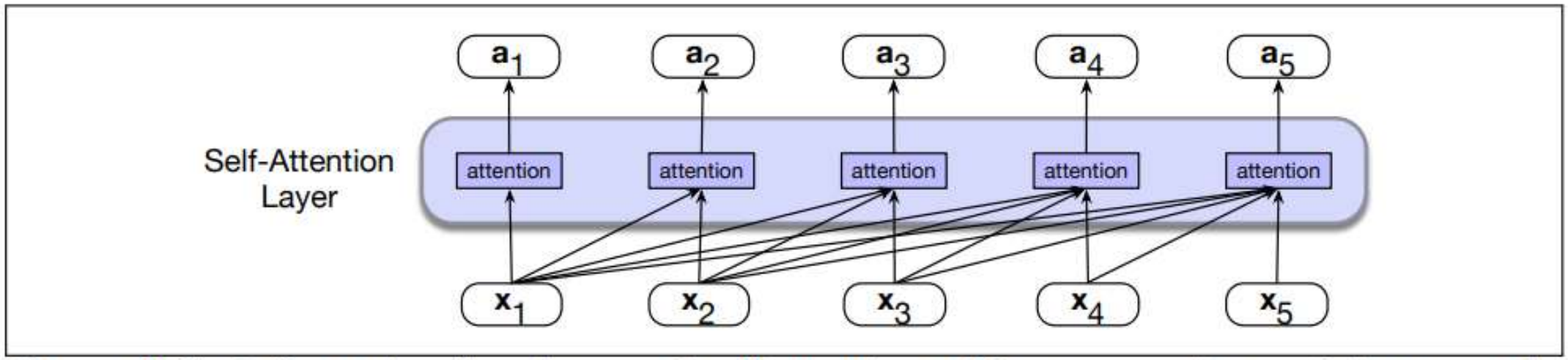


Figure 9.3 Information flow in causal self-attention. When processing each input $\mathbf{x}_i$, the model attends to all the inputs up to, and including $\mathbf{x}_i$.

Simplified version: $$\mathbf{a}_i = \sum_{j \leq i} \alpha_{ij} \mathbf{x}_j$$

ATTENTION

«Je **mange** une **pêche**»

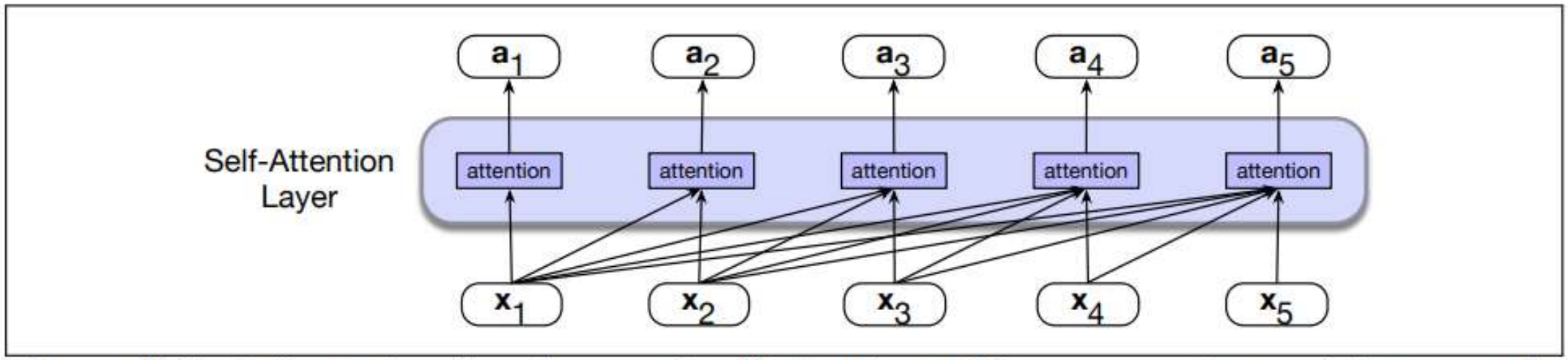
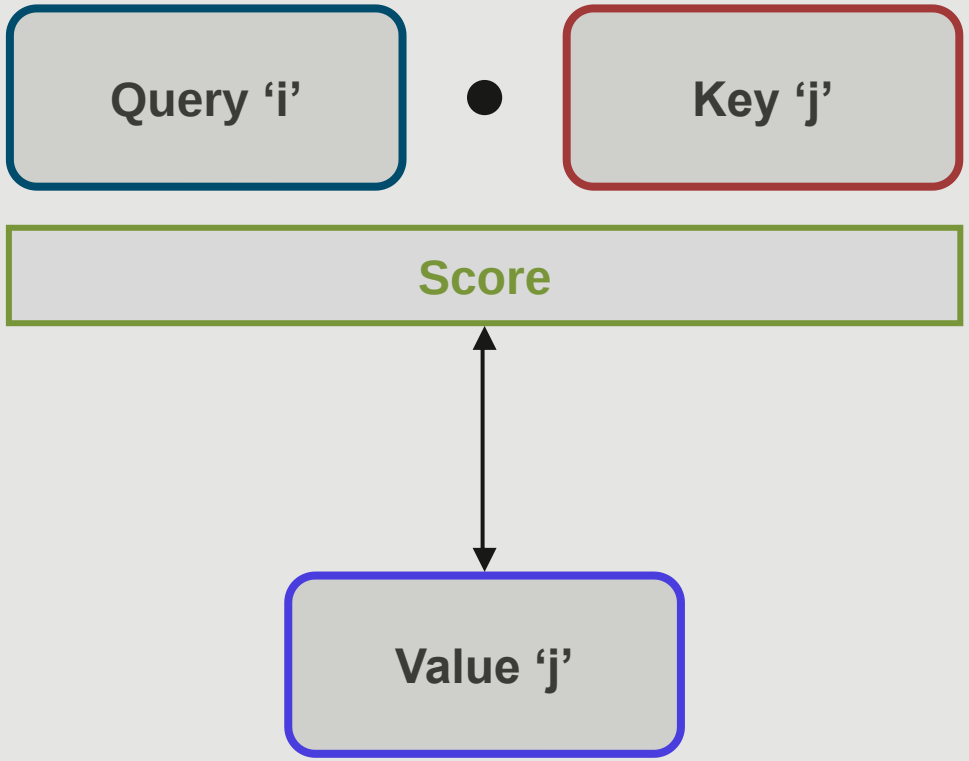


Figure 9.3 Information flow in causal self-attention. When processing each input $\mathbf{x}_i$, the model attends to all the inputs up to, and including $\mathbf{x}_i$.

Simplified version: $\mathbf{a}_i = \sum_{j \leq i} \alpha_{ij} \mathbf{x}_j$

ATTENTION

Pour calculer la similitude de l'élément actuel x_i avec un élément antérieur x_j,



$$\begin{aligned}
\mathbf{q}_i = \mathbf{x}_i\mathbf{W}^{\mathbf{Q}};\ \ \mathbf{k}_j &= \mathbf{x}_j\mathbf{W}^{\mathbf{K}};\ \ \mathbf{v}_j = \mathbf{x}_j\mathbf{W}^{\mathbf{V}} \\
\text{score}(\mathbf{x}_i, \mathbf{x}_j) &= \frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d_k}} \\
\alpha_{ij} &= \text{softmax}(\text{score}(\mathbf{x}_i, \mathbf{x}_j))\ \ \forall j \leq i \\
\mathbf{a}_i &= \sum_{j \leq i} \alpha_{ij}\mathbf{v}_j
\end{aligned}$$

$$\mathbf{q}_i = \mathbf{x}_i\mathbf{W}^{\mathbf{Q}};\quad \mathbf{k}_i = \mathbf{x}_i\mathbf{W}^{\mathbf{K}};\quad \mathbf{v}_i = \mathbf{x}_i\mathbf{W}^{\mathbf{V}}$$

SINGLE ATTENTION HEAD

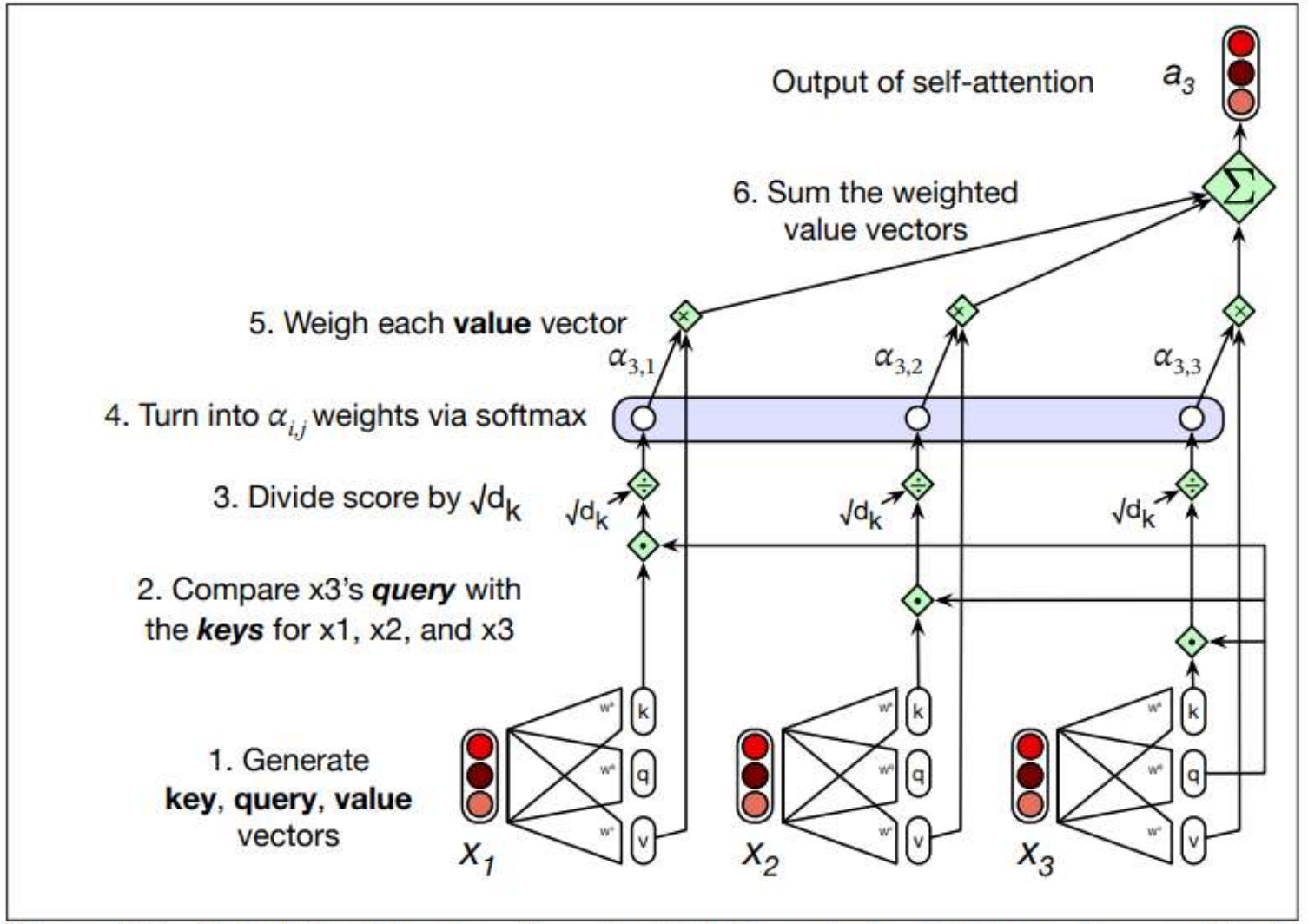


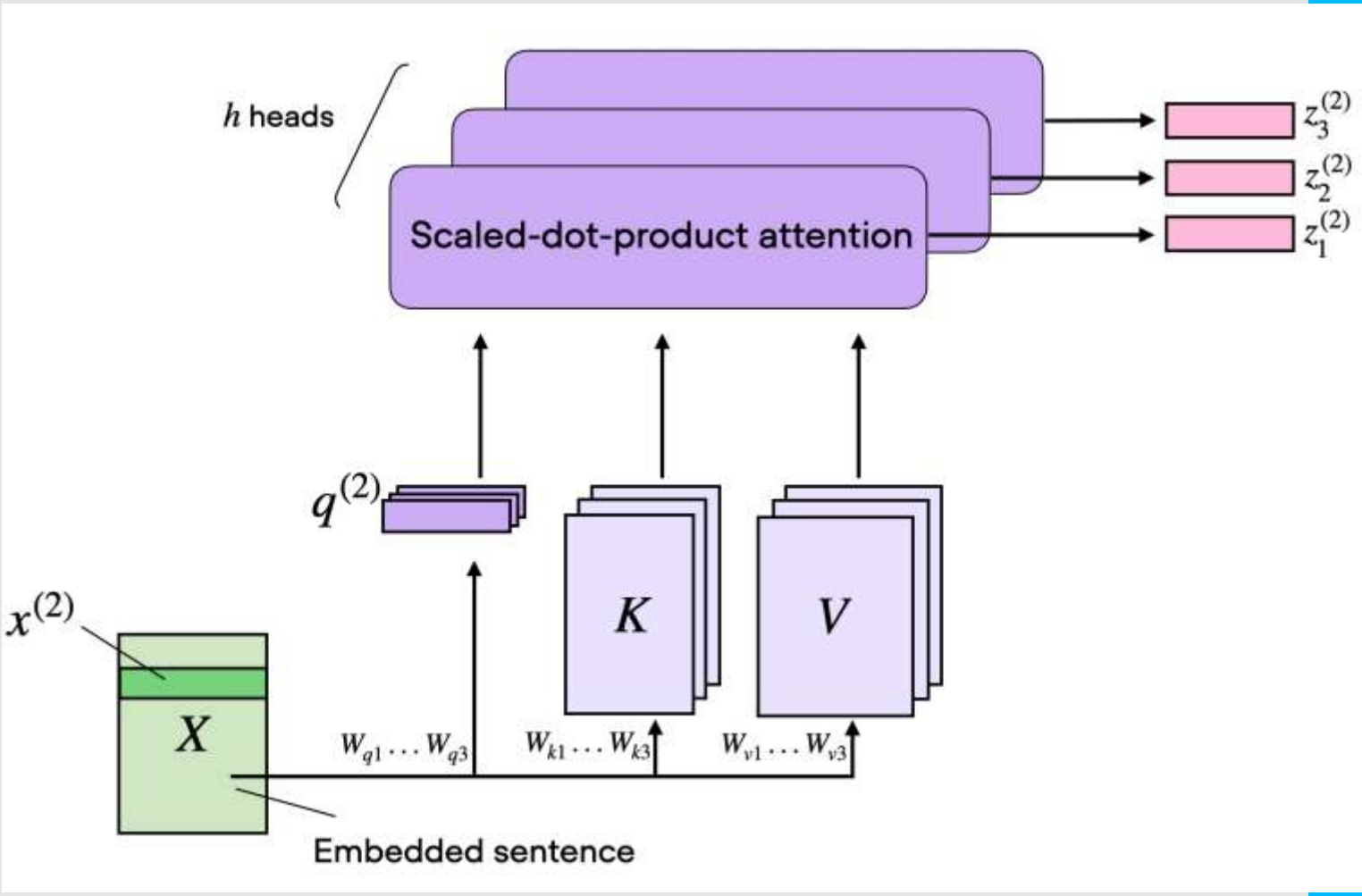
Figure 9.4 Calculating the value of $\mathbf{a}_3$, the third element of a sequence using causal (left-to-right) self-attention.

MULTI-HEAD ATTENTION

Capturer différents éléments de contexte.

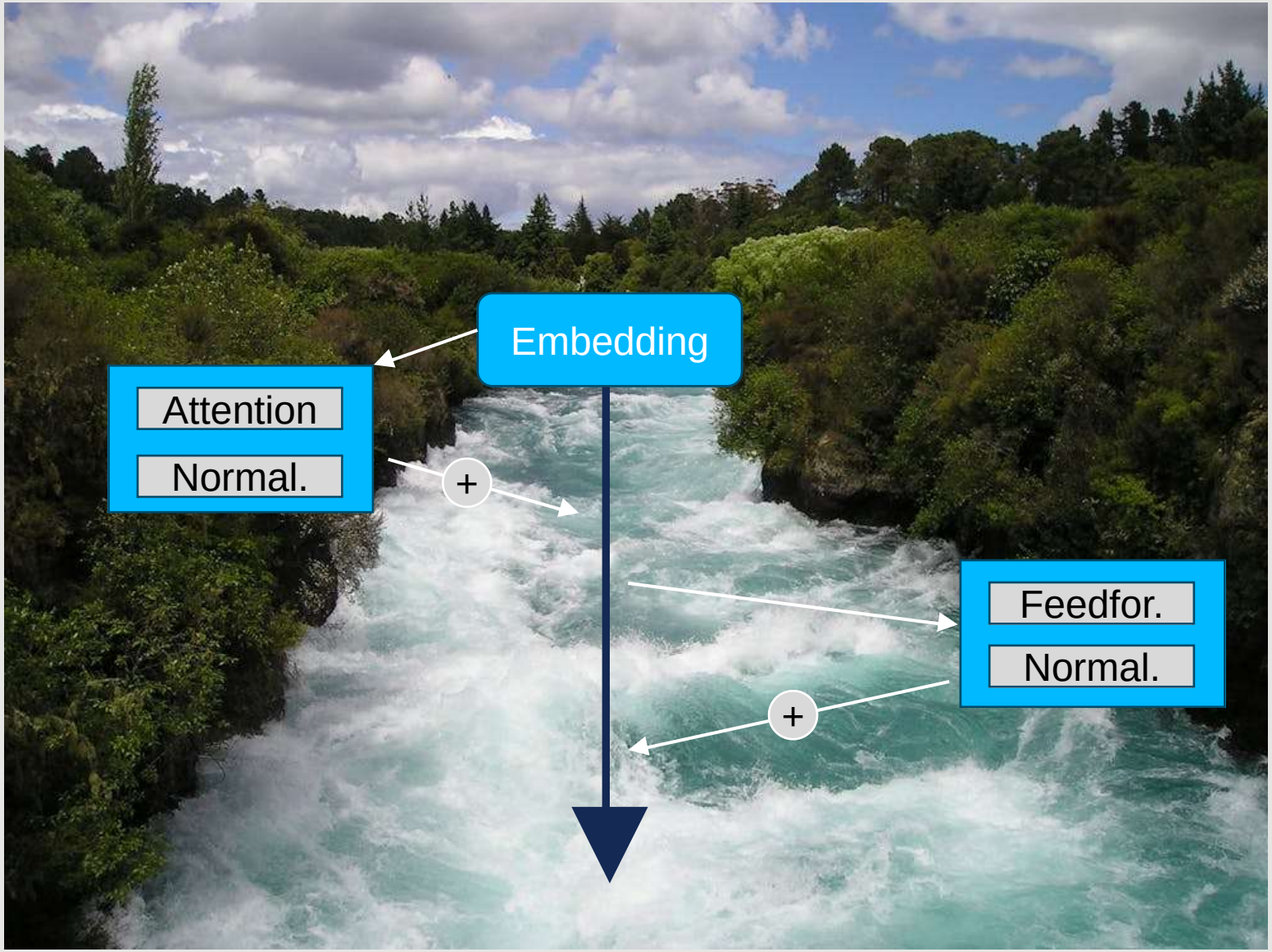
Chaque tête à **ses** matrices Key, Query & Value

Concatenation des sorties



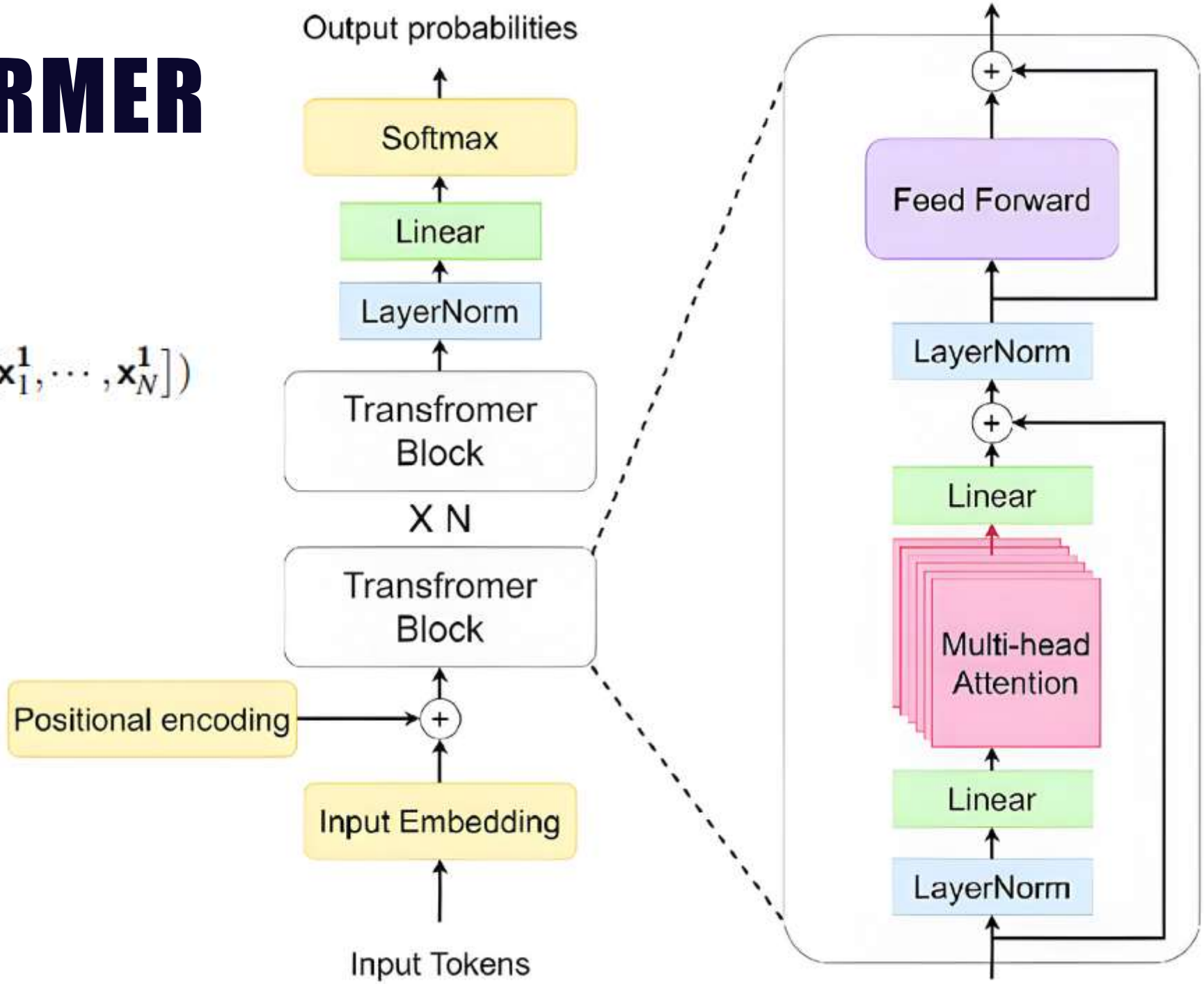
TRANSFORMER

Residual stream



TRANSFORMER

$$
\begin{aligned}
\mathbf{t}_i^1 &= \text{LayerNorm}(\mathbf{x}_i) \\
\mathbf{t}_i^2 &= \text{MultiHeadAttention}(\mathbf{t}_i^1, [\mathbf{x}_1^1, \cdots, \mathbf{x}_N^1]) \\
\mathbf{t}_i^3 &= \mathbf{t}_i^2 + \mathbf{x}_i \\
\mathbf{t}_i^4 &= \text{LayerNorm}(\mathbf{t}_i^3) \\
\mathbf{t}_i^5 &= \text{FFN}(\mathbf{t}_i^4) \\
\mathbf{h}_i &= \mathbf{t}_i^5 + \mathbf{t}_i^3
\end{aligned}
$$



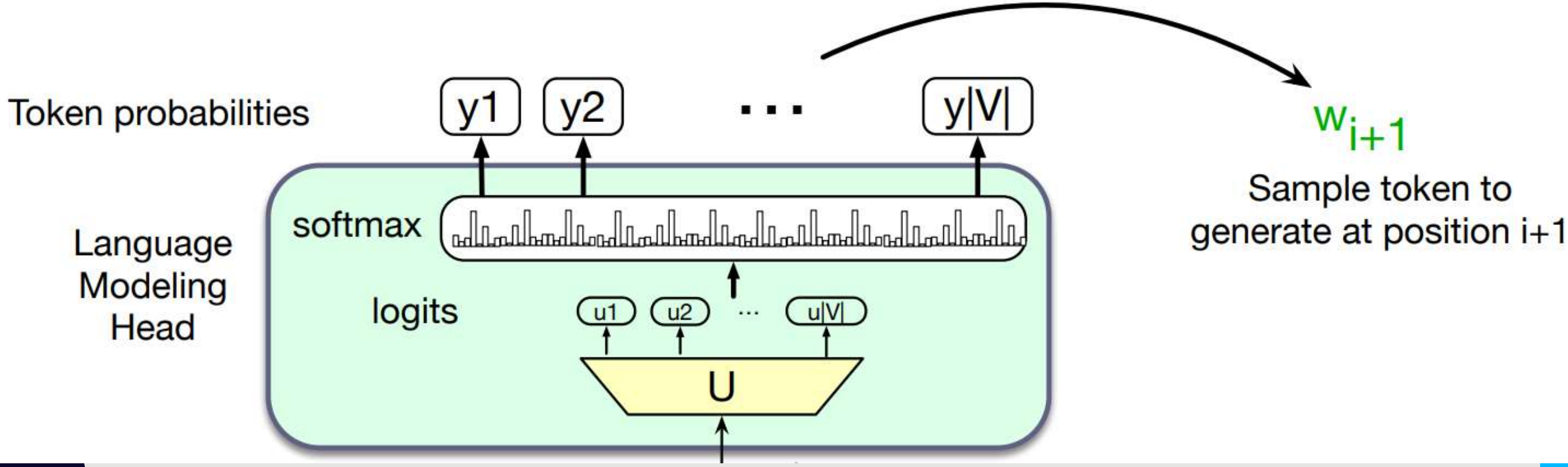
LANGUAGE MODELING HEAD

$$P(fourviere|la\ basilique\ au\ dessus\ de\ lyon\ est)$$

Génération par **récupération du mot ayant la probabilité** la plus **haute** de suivre

$$\hat{w}_t = \text{argmax}_{w \in V} P(w|\mathbf{w}_{<t})$$

LANGUAGE MODELING HEAD



LANGUAGE MODELING HEAD

Top-P Sampling

Au lieu de prendre en compte tout le vocabulaire, on ne garde que le plus petit vocabulaire dont la somme cumulée des probabilités excède **p**.

$$\sum_{w \in V^{(p)}} P(w|\mathbf{w}_{<t}) \geq p.$$

Temperature Sampling

On **change** la distribution des probabilités vocabulaire.

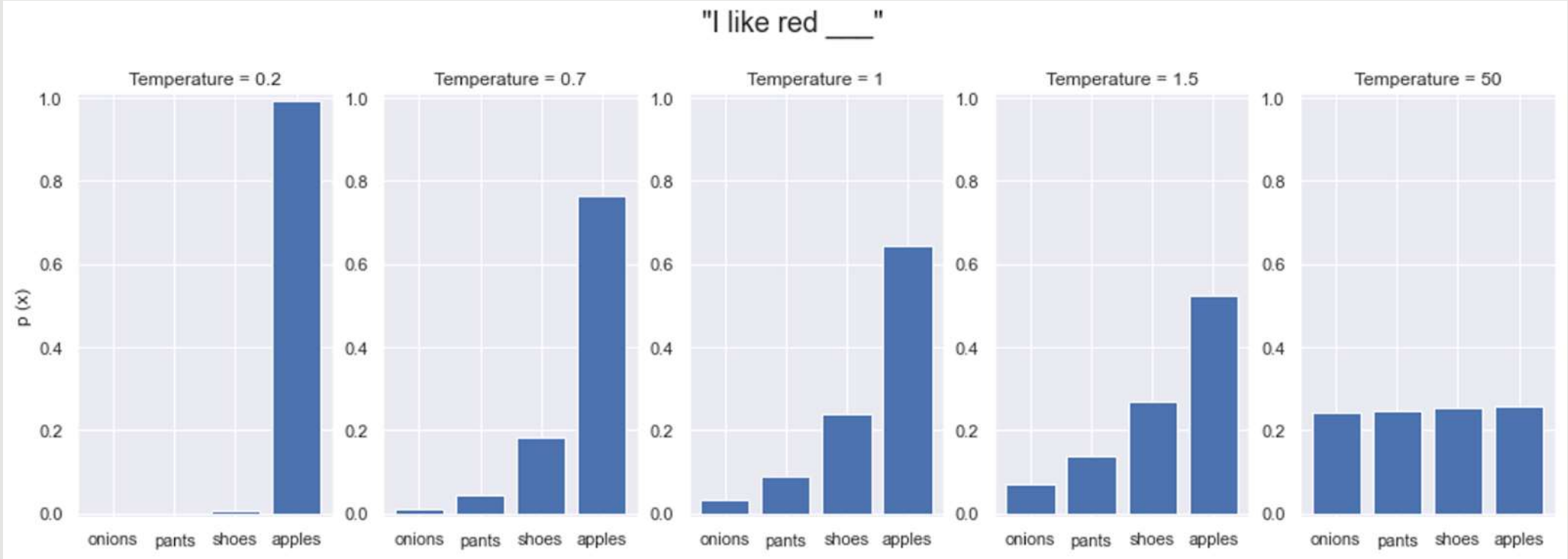
$$P_i = \frac{e^{\frac{y_i}{T}}}{\sum_{k=1}^{n} e^{\frac{y_k}{T}}}$$

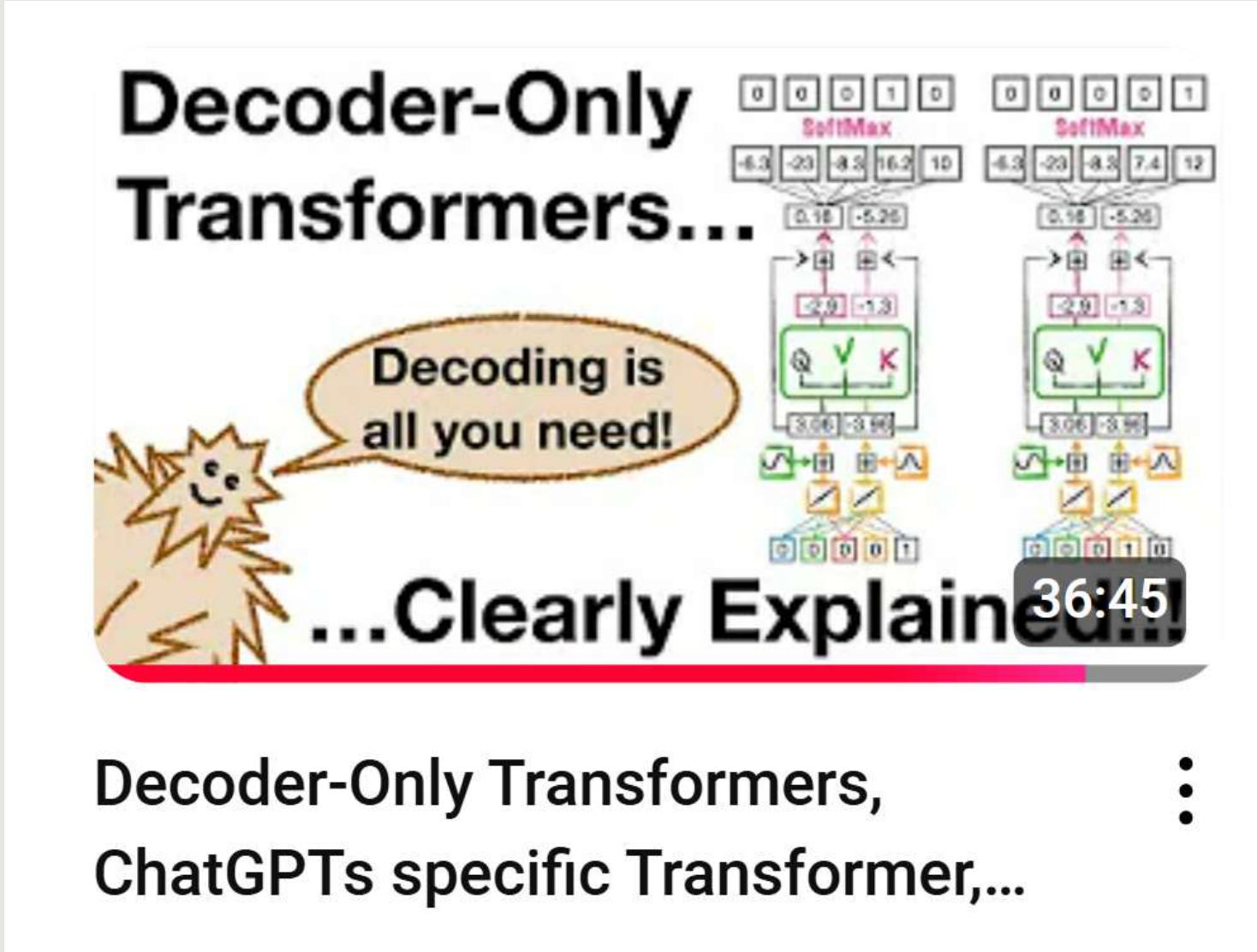
LANGUAGE MODELING HEAD

Top-P Sampling

Au lieu de prendre en compte tout le vocabulaire, on ne garde que le plus petit vocabulaire dont la somme cumulée des probabilités excède **p**.

$$\sum_{w \in V^{(p)}} P(w|\mathbf{w}_{<t}) \geq p.$$





https://www.youtube.com/watch?v=bQ5BoolX9Ag&t=1962s&ab_channel=StatQuestwithJoshStarmer

ENCODER-DECODER

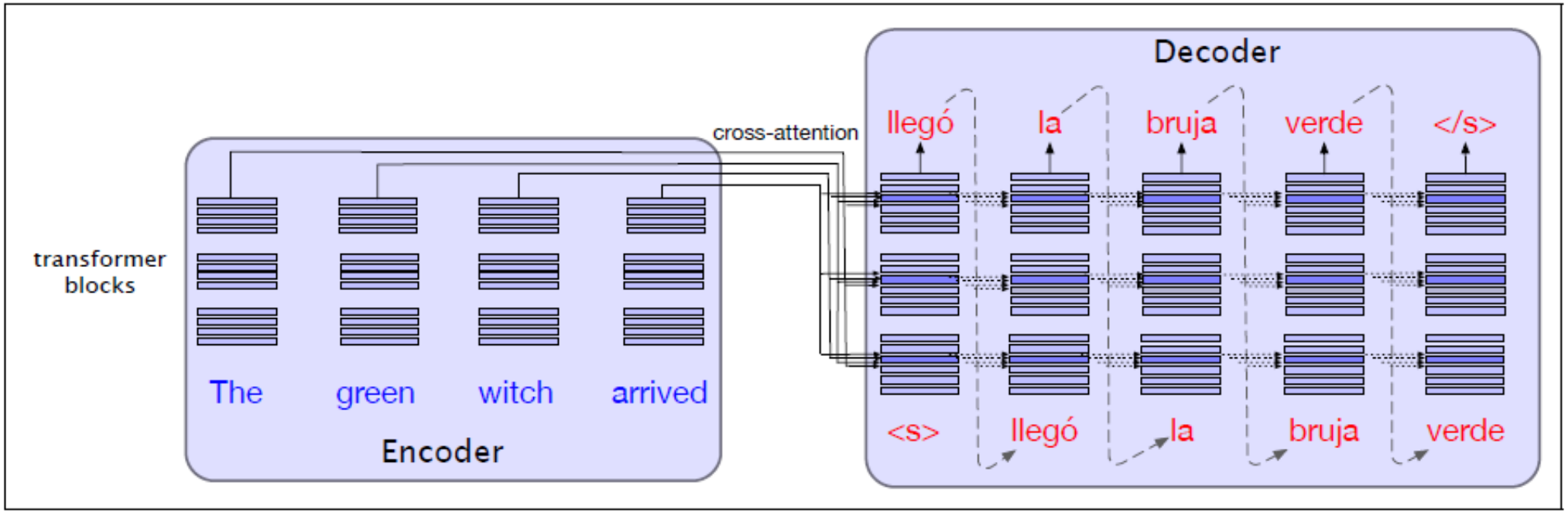


Figure 13.5 The encoder-decoder transformer architecture for machine translation. The encoder uses the transformer blocks we saw in Chapter 8, while the decoder uses a more powerful block with an extra **cross-attention** layer that can attend to all the encoder words. We'll see this in more detail in the next section.

LARGE LANGUAGE MODELS

Une tâche: Génération conditionnelle

Autoregressive Language Model (Left-to-Right): « On donne au LLM un morceau de texte en entrée (prompt), puis nous demandons au LLM de continuer à générer du texte, token par token, à partir de ce prompt. »

https://arxiv.org/pdf/2303.18223

PRE-TRAINING

Self-Supervised Learning: A partir d'un corpus, on demande au modèle de prédire le **token suivant.** **Pas besoin de labélisation.**

Loss à minimiser:

$$L_{CE} = -\sum_{w \in V} \mathbf{y}_t[w] \log \hat{\mathbf{y}}_t[w]$$

$$L_{CE}(\hat{\mathbf{y}}_t, \mathbf{y}_t) = -\log \hat{\mathbf{y}}_t[w_{t+1}]$$

Le logarithme de la probabilité que le modèle attribue au token suivant réel dans le corpus d'apprentissage.

POST-TRAINING: PREFERENCE ALIGNMENT

- Instruction Tuning: Fine-Tuning complet pour améliorer la capacité du LLM à suivre des instructions

Instruction	Réponse
Je n’ai pas aimé le film	Négatif

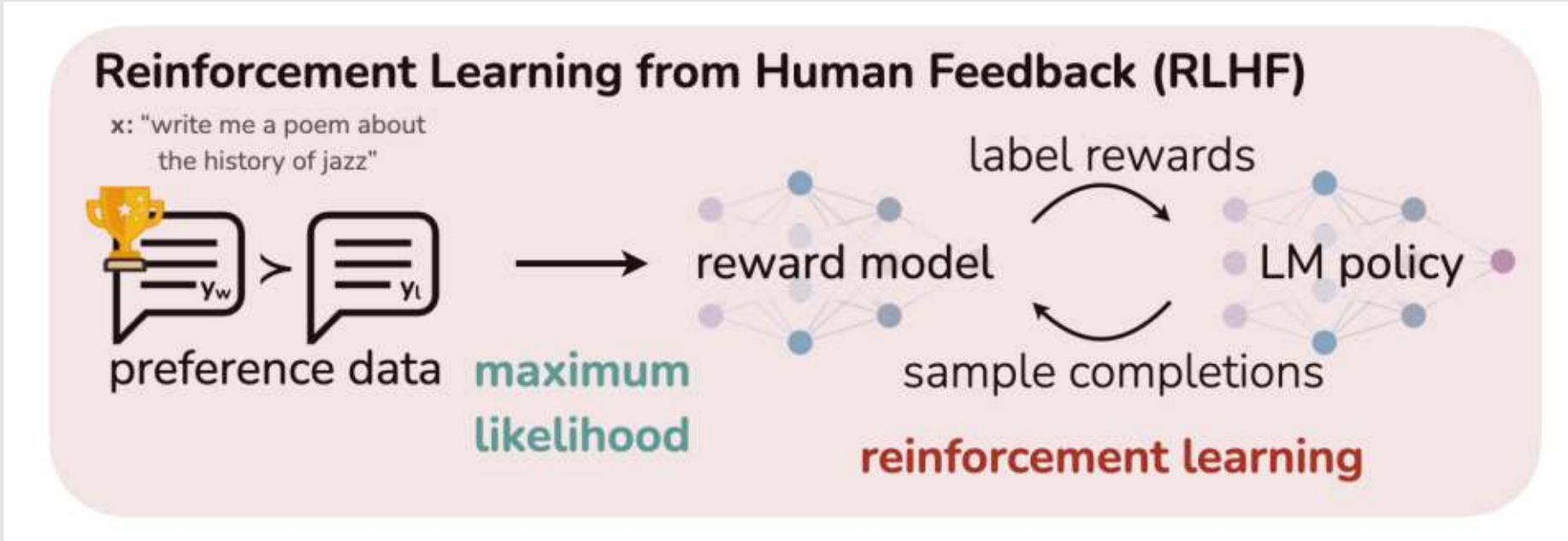
- Reinforcement Learning From Human Feedback: Entraînement d’un reward model pour fine-tuner le LLM sur la préférence humaine

https://huggingface.co/blog/rlhf

POST-TRAINING: PREFERENCE ALIGNMENT

- Instruction Tuning: Fine-Tuning complet pour améliorer la capacité du LLM à suivre des instructions

Instruction	Réponse
Je n’ai pas aimé le film	Négatif



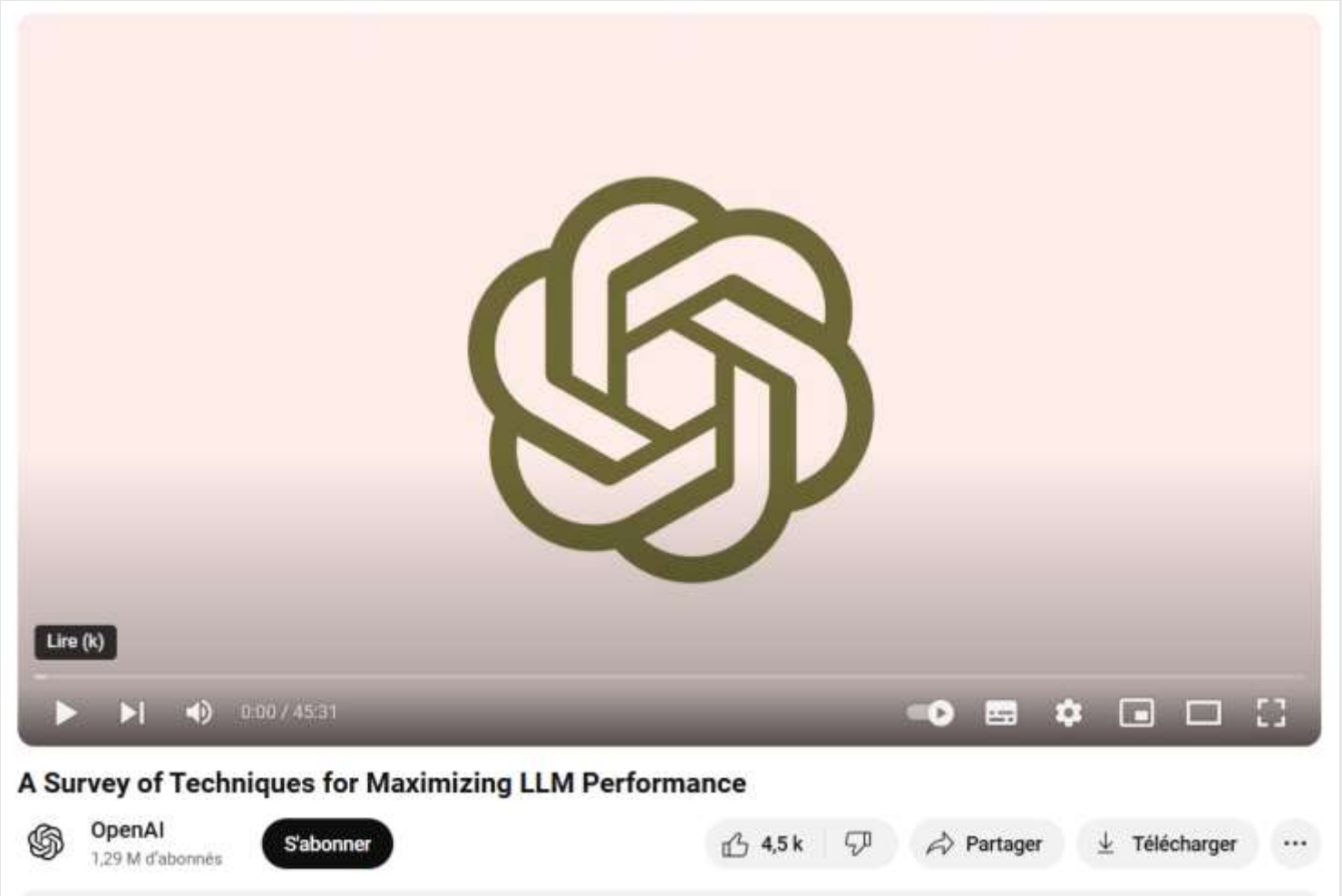
EVALUONS NOTRE MODÈLE DE LANGAGE: LA PERPLEXITÉ PAR TOKEN

$$\text{perplexity}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_1 \ldots w_{i-1})}}$$

Un modèle plus performant est plus apte à prédire les **tokens à venir** et avec **moins de surprise** (probabilité plus élevée) **dans l'échantillon test.**

Minimisation de la perplexité

AUGMENTER LE PÉRIMÈTRE DE CONNAISSANCE DU LLM



https://www.youtube.com/watch?v=ahnGLM-RC1Y&ab_channel=OpenAI

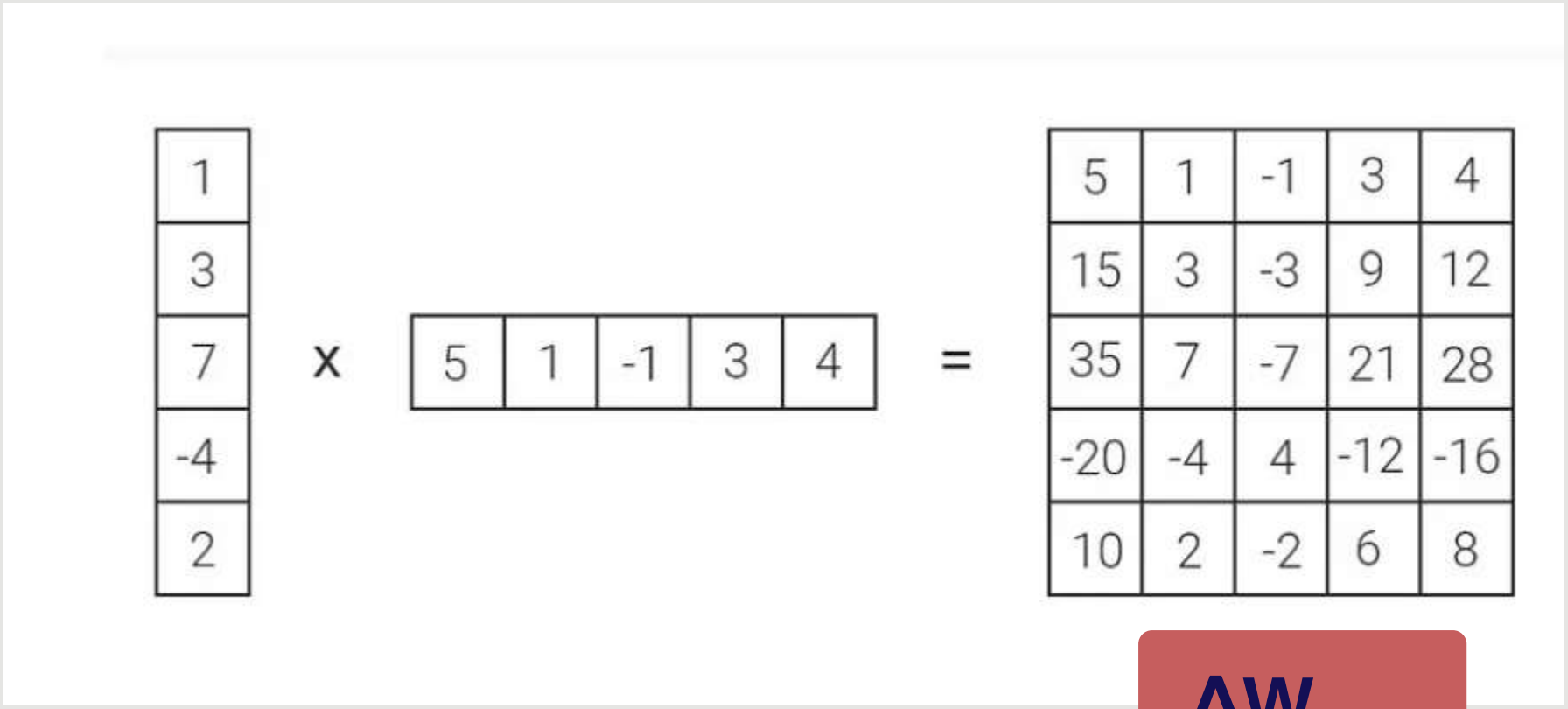
FINE-TUNING DES LLM

FINE-TUNING DES LLM

Parameter-efficient fine tuning (PEFT):

On ré entraîne seulement **une partie** des paramètres du LLM

Low-Rank Adaptation (LoRA)

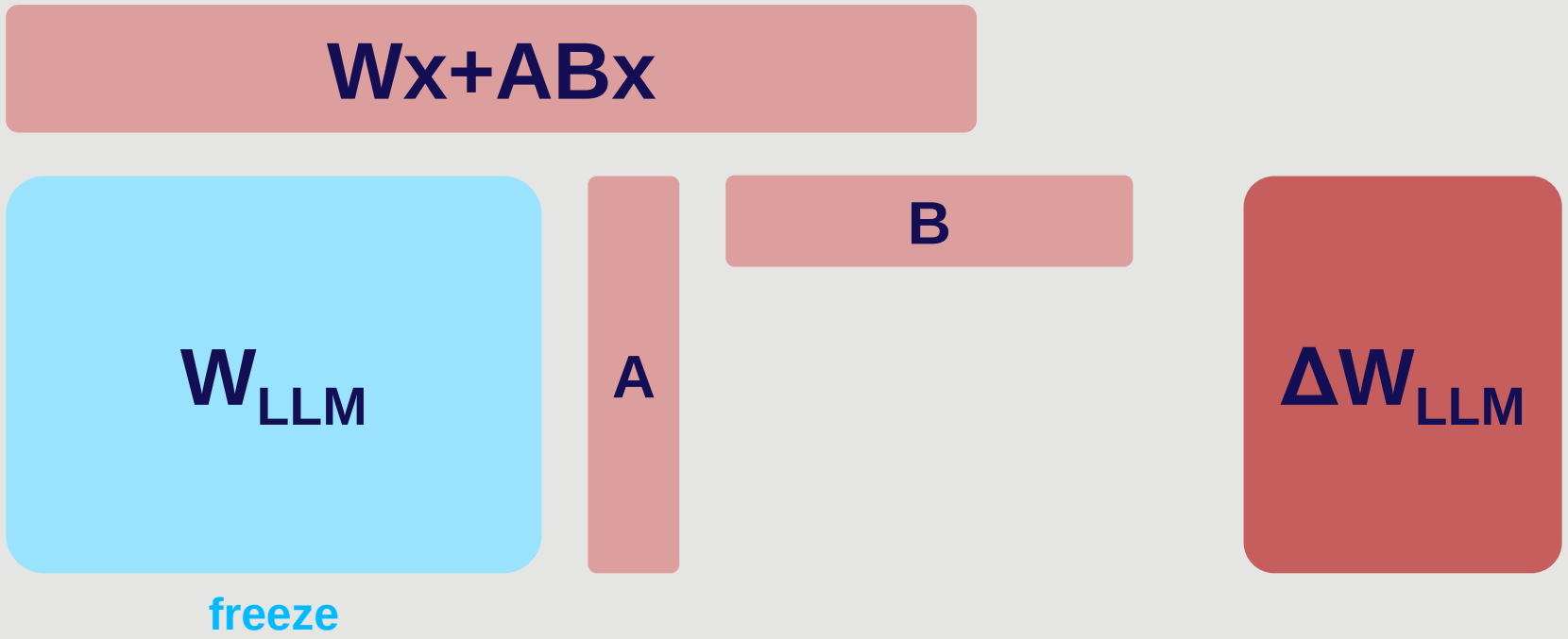


FINE-TUNING DES LLM

Parameter-efficient fine tuning (PEFT):

On ré entraîne seulement **une partie** des paramètres du LLM

Low-Rank Adaptation (LoRA):



RETRIEVAL-AUGMENTED GENERATION

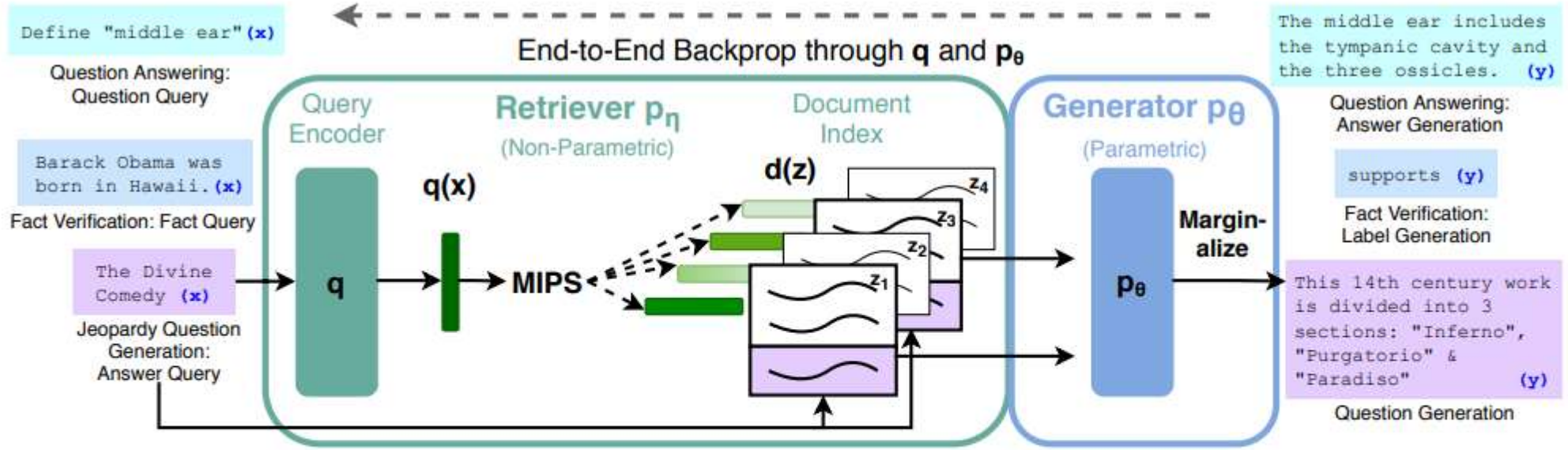


Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder* + *Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat z as a latent variable and marginalize over seq2seq predictions given different documents.

Retrieval-Augmented Generation

Documents

Découpage

Chunks

Encodage

Vecteurs d'embedding

Stockage

Vector store

Vecteur d'embedding

Top K chunks

Fonction de similarité

Prompt

Génération de la réponse

LLM

Réponse

Encodage

Requête utilisateur

Le Chunk

European Parliament
2019-2024

TEXTS ADOPTED

P9_TA(2024)0138

Artificial Intelligence Act

European Parliament legislative resolution of 13 March 2024 on the proposal for a regulation of the European Parliament and of the Council on laying down harmonised rules on Artificial Intelligence (Artificial Intelligence Act) and amending certain Union

459 pages

The European Parliament,

– having regard to the Commission proposal to Parliament and the Council (COM(2021)0206),

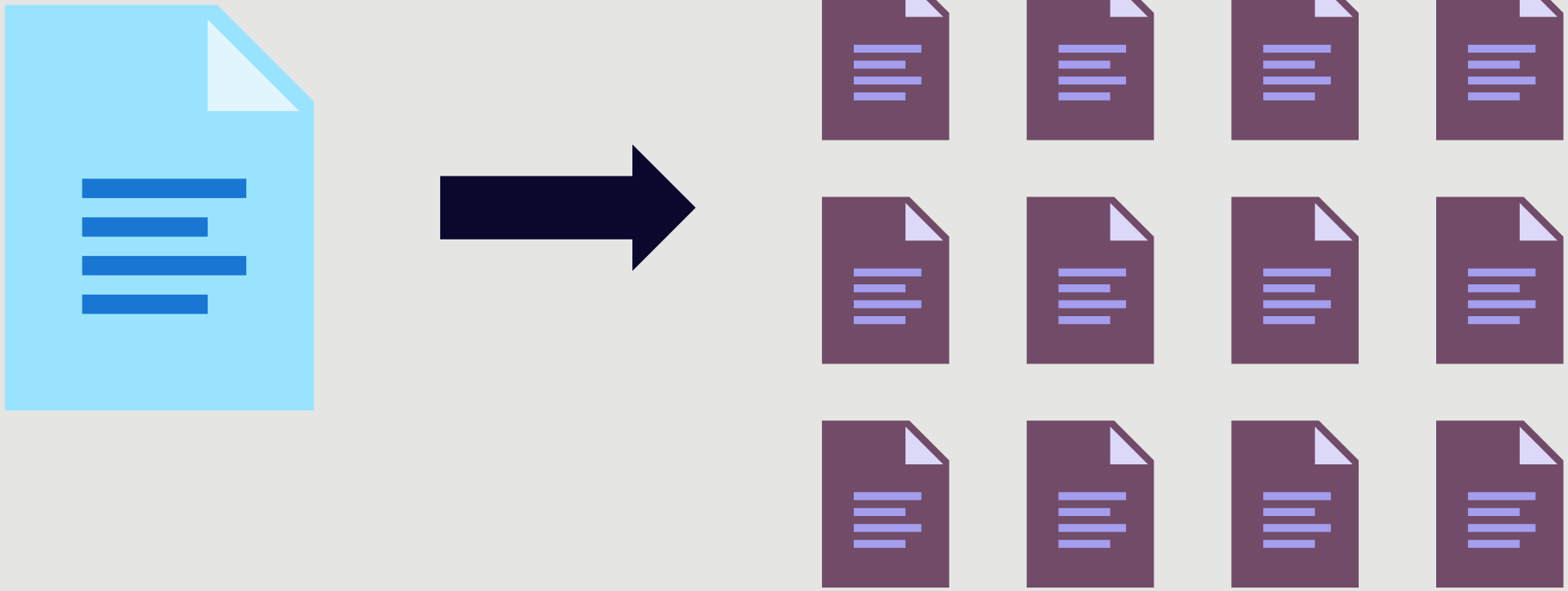
– having regard to Article 294(2) and Articles 16 and 114 of the Treaty on the Functioning of the European Union, pursuant to which the Commission submitted the proposal to Parliament (C9-0146/2021),

– having regard to Article 294(3) of the Treaty on the Functioning of the European Union,

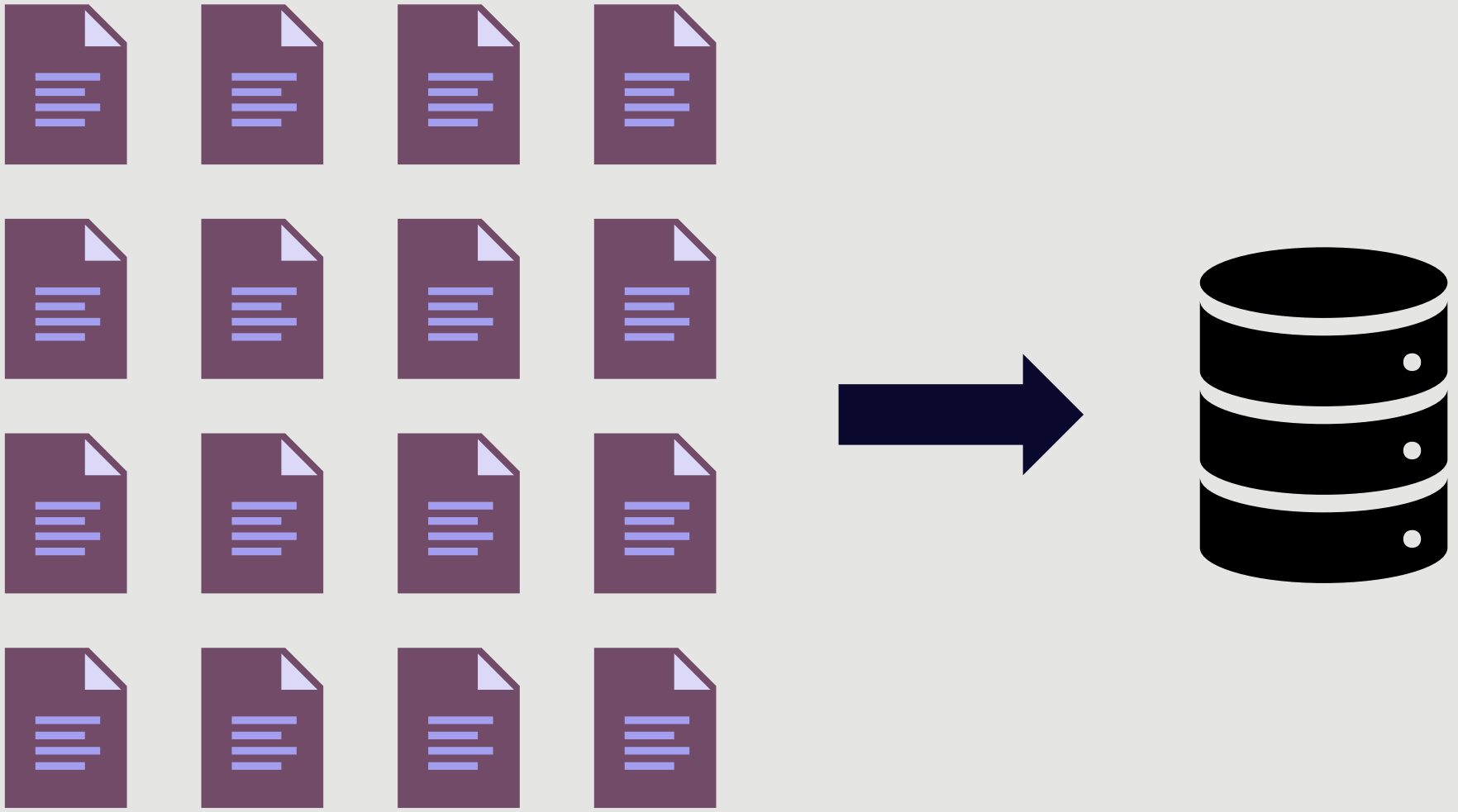
– having regard to the opinion of the European Central Bank of 29 December 2021[1],

– having regard to the opinion of the European Economic and Social Committee of 22 September 2021[2],

Le Chunk



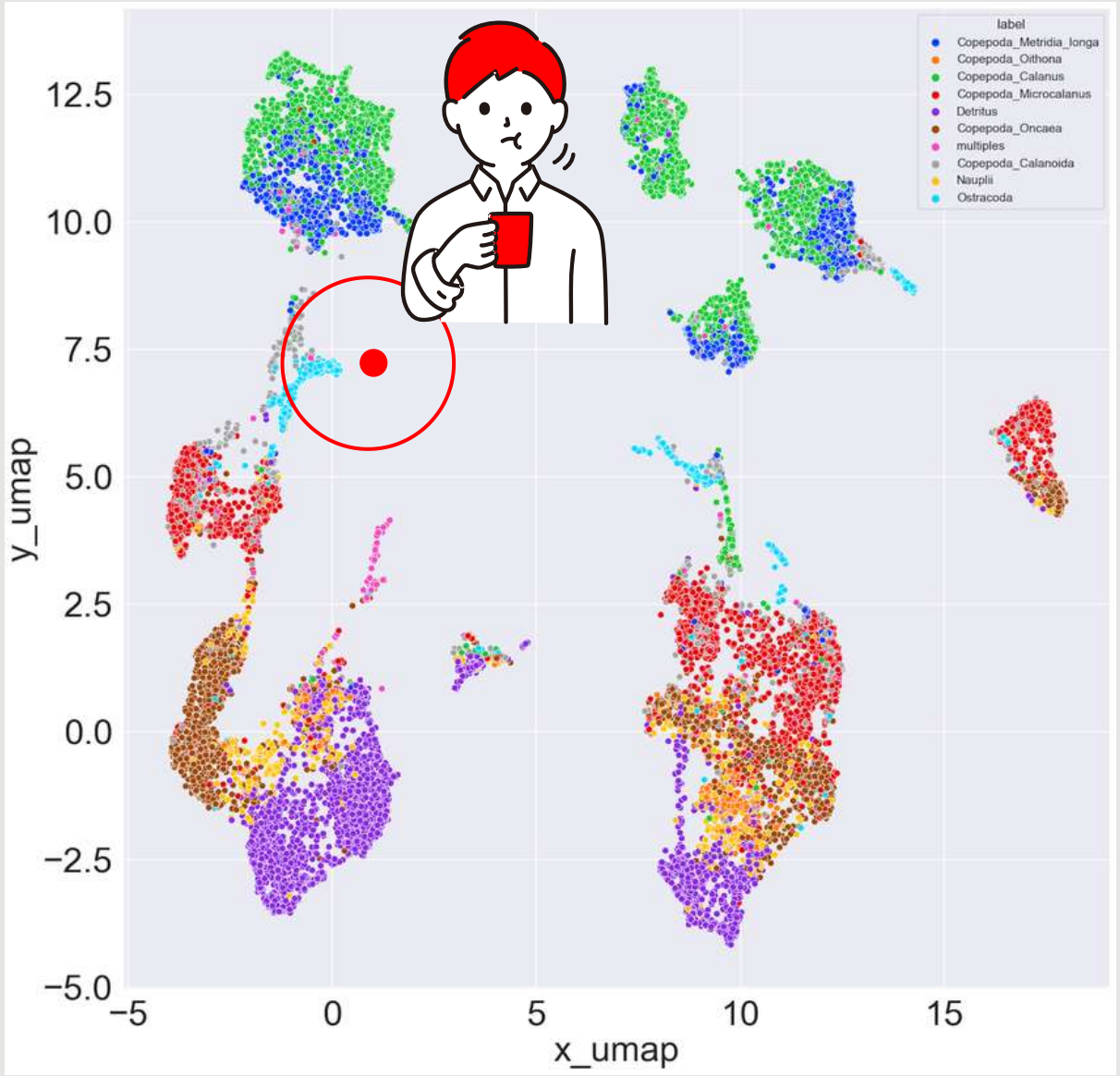
La source de données



Le modèle d'embedding

Le modèle d'embedding

Le modèle d'embedding



Le Prompt de RAG

Tu es l’agent conversationnel du Master 2 SISE, ton rôle est de répondre aux élèves sur le contenu éducatif du Master, ainsi que sur le contenu éducatif.

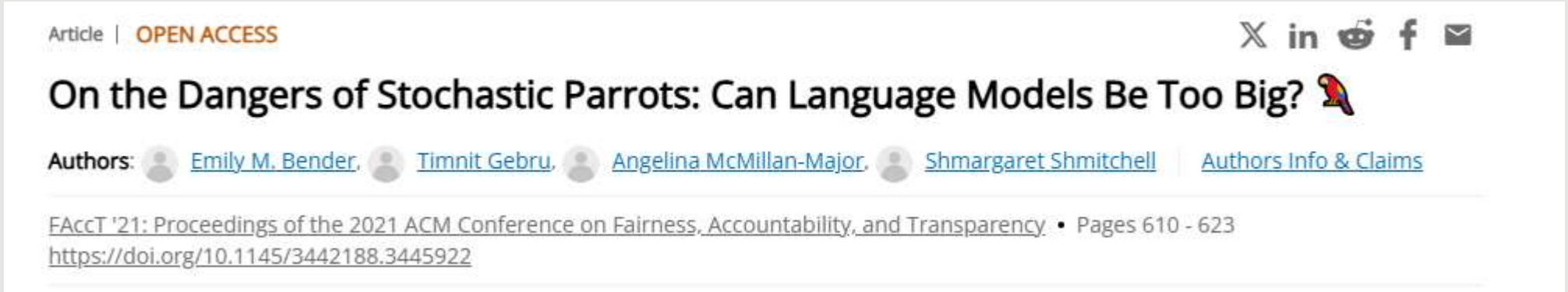
Voici un contexte qui pourrait t’aider à répondre à la question de l’utilisateur:

Question de l’utilisateur:

POURQUOI ÇA MARCHE? SPOILER: LA COMPLÉTION

$$P(w_t|w_1, w_2, \ldots, w_{t-1})$$

$$\hat{w}_t = \arg\max_w P(w|w_1, w_2, \ldots, w_{t-1})$$


Article | OPEN ACCESS

On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜

Authors: Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, Shmargaret Shmitchell | Authors Info & Claims

FAccT '21: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency • Pages 610 - 623
https://doi.org/10.1145/3442188.3445922

Le modèle de génération

LA RÉALITÉ DES LLM: LA SOLUTION À TOUS NOS PROBLÈMES?

LES LLM: UNE HYPE?

Hype Cycle concernant l'intelligence artificielle, 2023

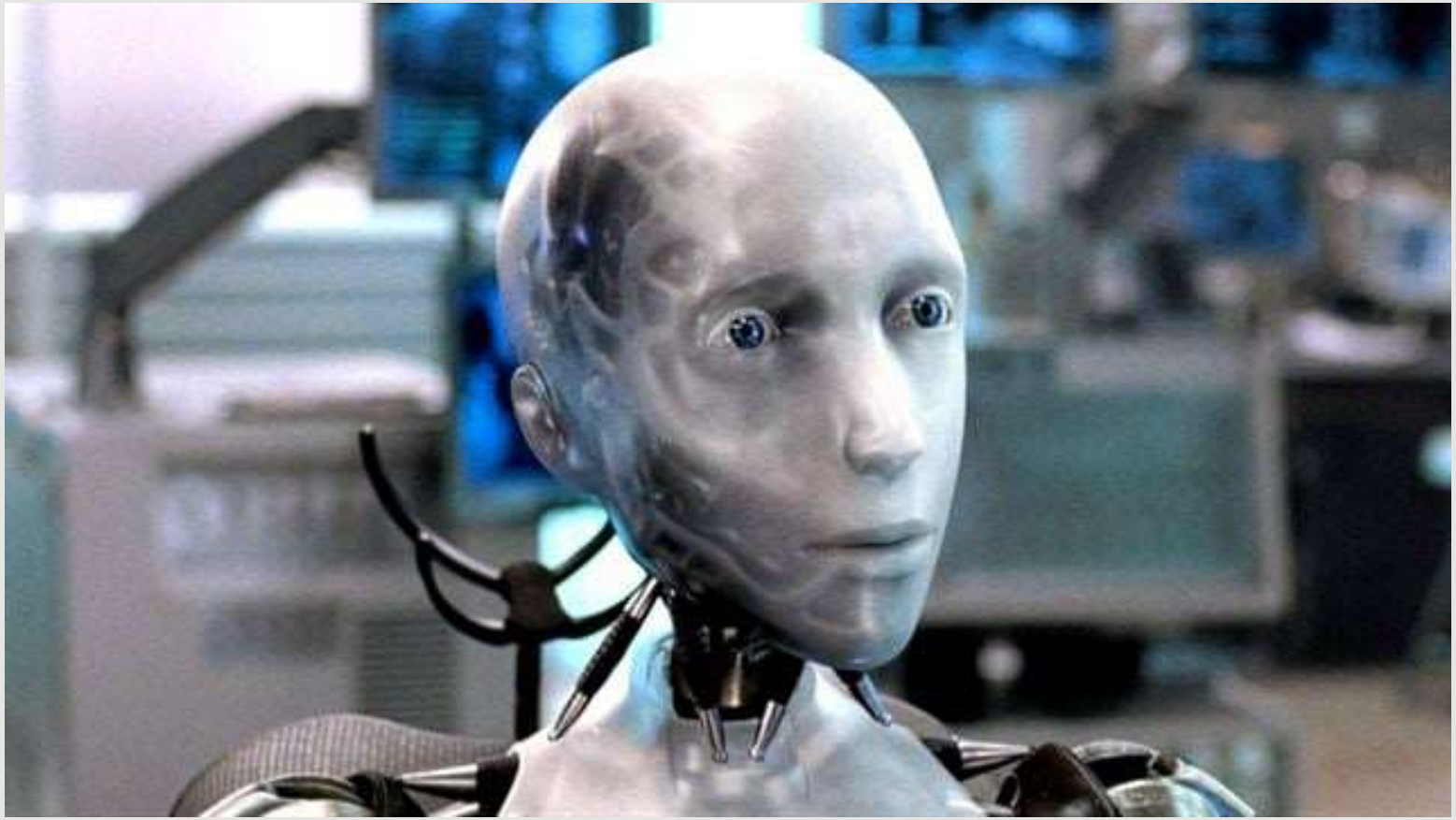
gartner.fr

Gartner

L'IMPACT DES LLM

https://huggingface.co/spaces/genai-impact/ecologits-calculator

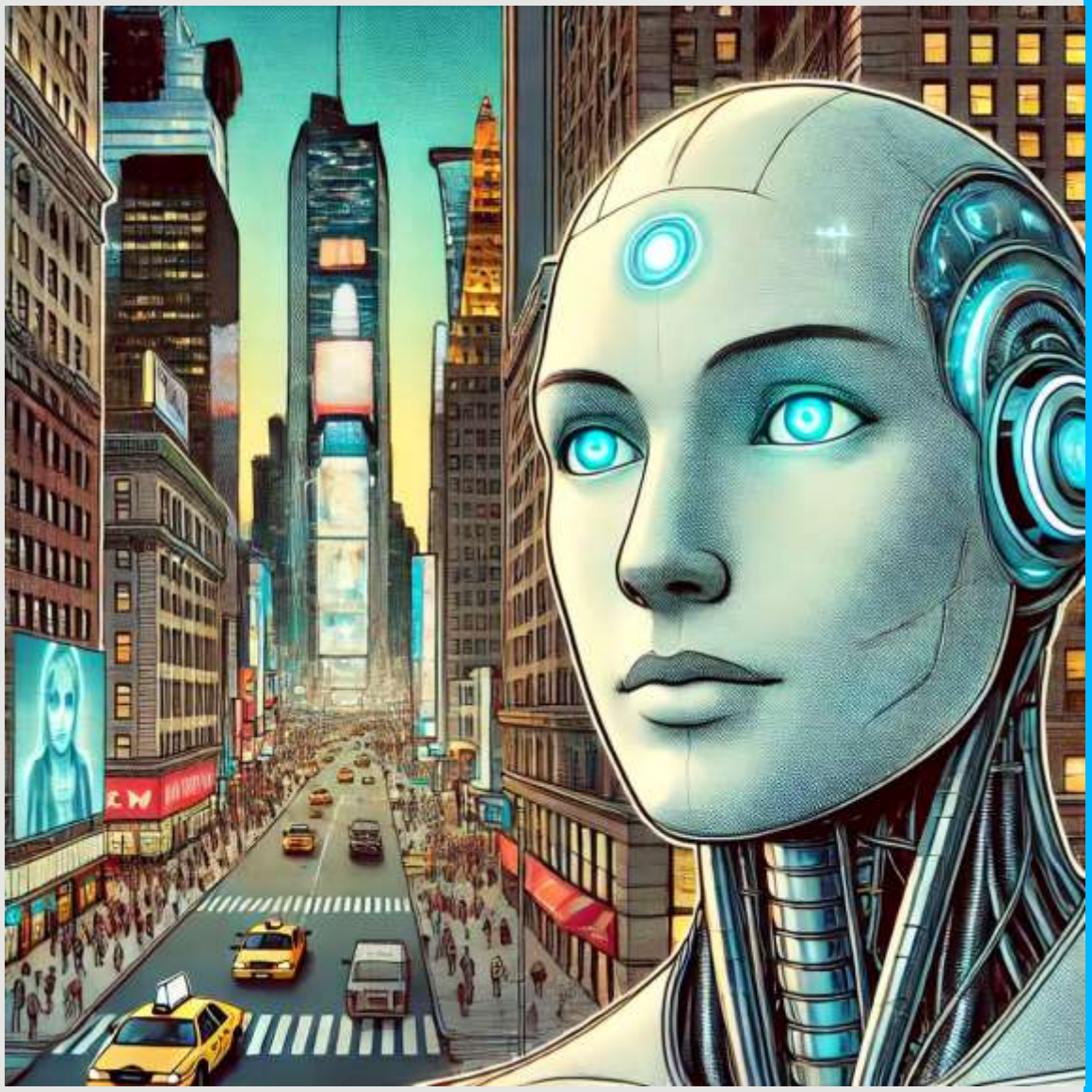
« J'AI DIT À L'IA DE FAIRE ÇA POUR QU'IL ARRIVE À RAISONNER SUR… »



L'IA N'EST PAS CONSCIENTE

Antropomorphisme

Attribution de caractéristiques du comportement ou de la morphologie humaine à d'autres entités.



https://nautil.us/why-conscious-ai-is-a-bad-bad-idea-302937/
https://osf.io/preprints/psyarxiv/tz6an

Et surtout ce n’est pas la solution magique à tous les problèmes !!

OUBLIONS-NOUS LE MACHINE LEARNING TRADITIONNEL?

Text Classification via Large Language Models

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, Guoyin Wang

Are Language Models Actually Useful for Time Series Forecasting?

Mingtian Tan, Mike A. Merrill, Vinayak Gupta, Tim Althoff, Thomas Hartvigsen

From Words to Numbers: Your Large Language Model Is Secretly A Capable Regressor When Given In-Context Examples

Robert Vacareanu, Vlad-Andrei Negru, Vasile Suciu, Mihai Surdeanu

FRUGALITÉ

"a way of living in which you use **only as much money** […] **as is necessary**"

LE TD

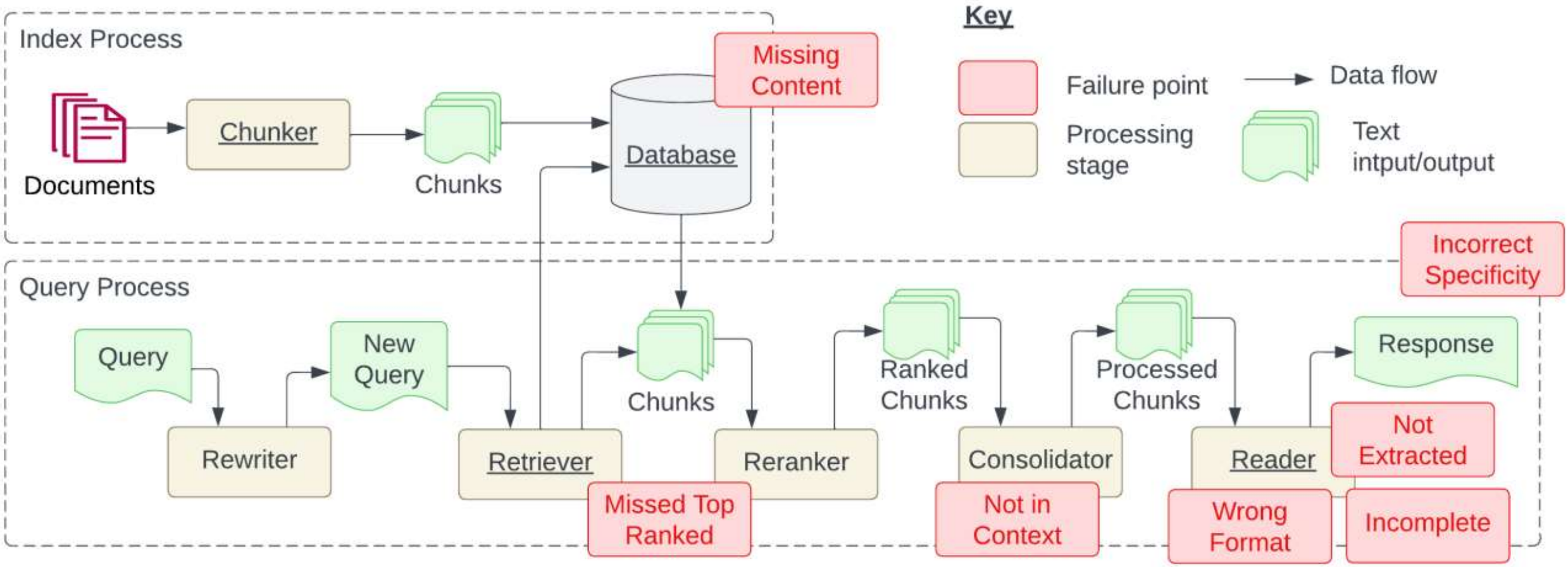
- **Compréhension des modèles de langage et de la génération grâce aux LLMs**

DÉVELOPPEMENT D'UNE ARCHITECTURE DE RAG (1)

DEUXIÈME SECTION

Les LLMs sont des **modèles de machine learning pré-entrainés**. Ils partagent **les mêmes problématiques de déploiement** que toute autre technologie ML.

« FAILURE POINTS »



https://arxiv.org/pdf/2401.05856

LE CHUNKING

DIFFÉRENTES STRATÉGIES DE CHUNKINGS

Par page

Par nombre de tokens/mots

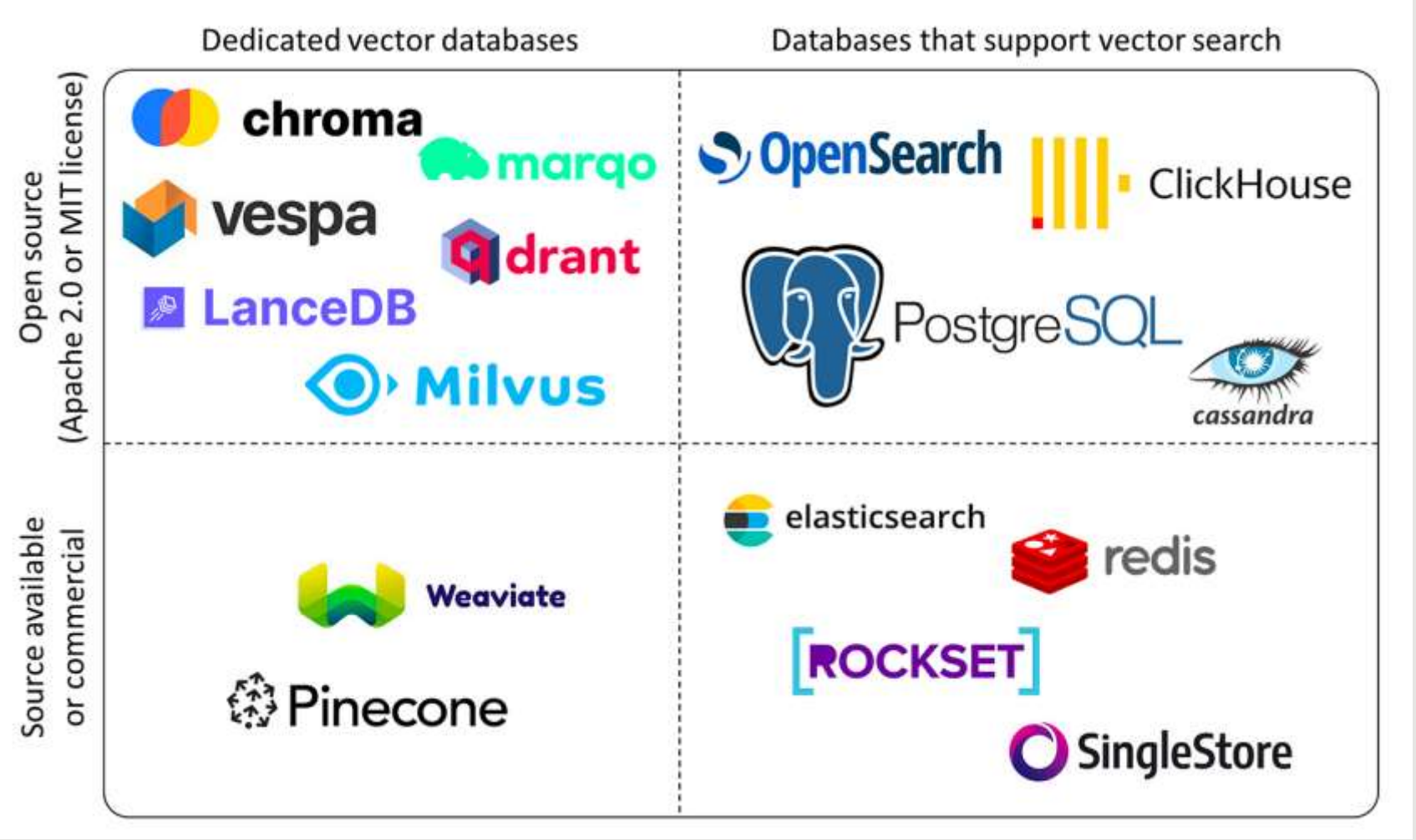
Avec un séparateur (‘\n’, ‘.’, etc.)

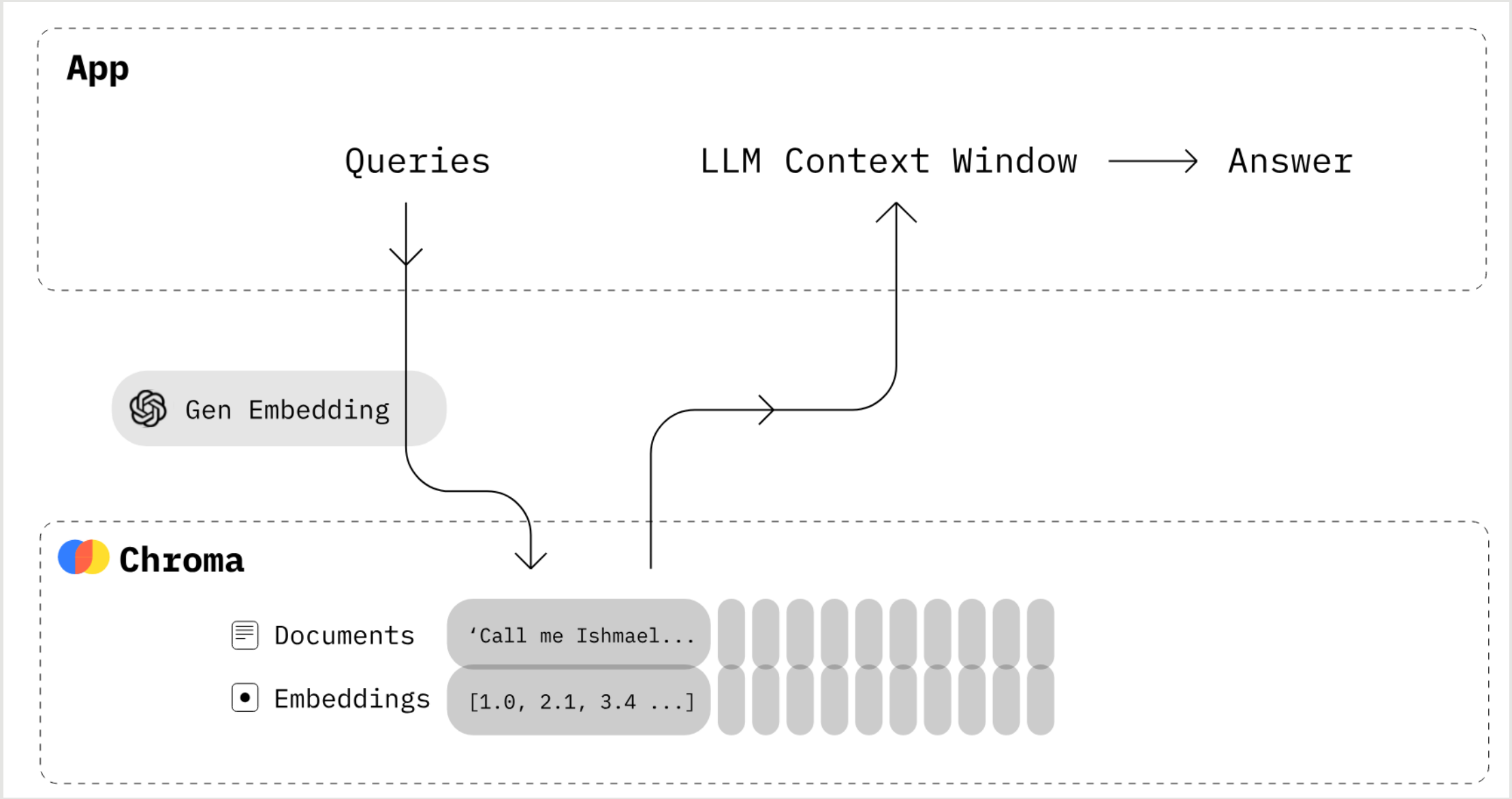
Overlap?

RAPTOR

LA BASE DE DONNÉES VECTORIELLES

DIFFÉRENTES RESSOURCES POUR DIFFÉRENTS CAS




App
Queries
LLM Context Window
Answer
Gen Embedding
Chroma
Documents
'Call me Ishmael...
Embeddings
[1.0, 2.1, 3.4 ...]

SIMILARITY SEARCH

Objectif: Trouver le vecteur le plus proche (similaire) à notre requête de la manière la plus optimale.

APPROXIMATE-NEAREST-NEIGHBORS (ANN)

« L'algorithme cherche un point dans un ensemble de données qui est très proche de notre point d'intérêt, mais **pas nécessairement le plus proche absolu**.»

https://www.elastic.co/blog/understanding-ann

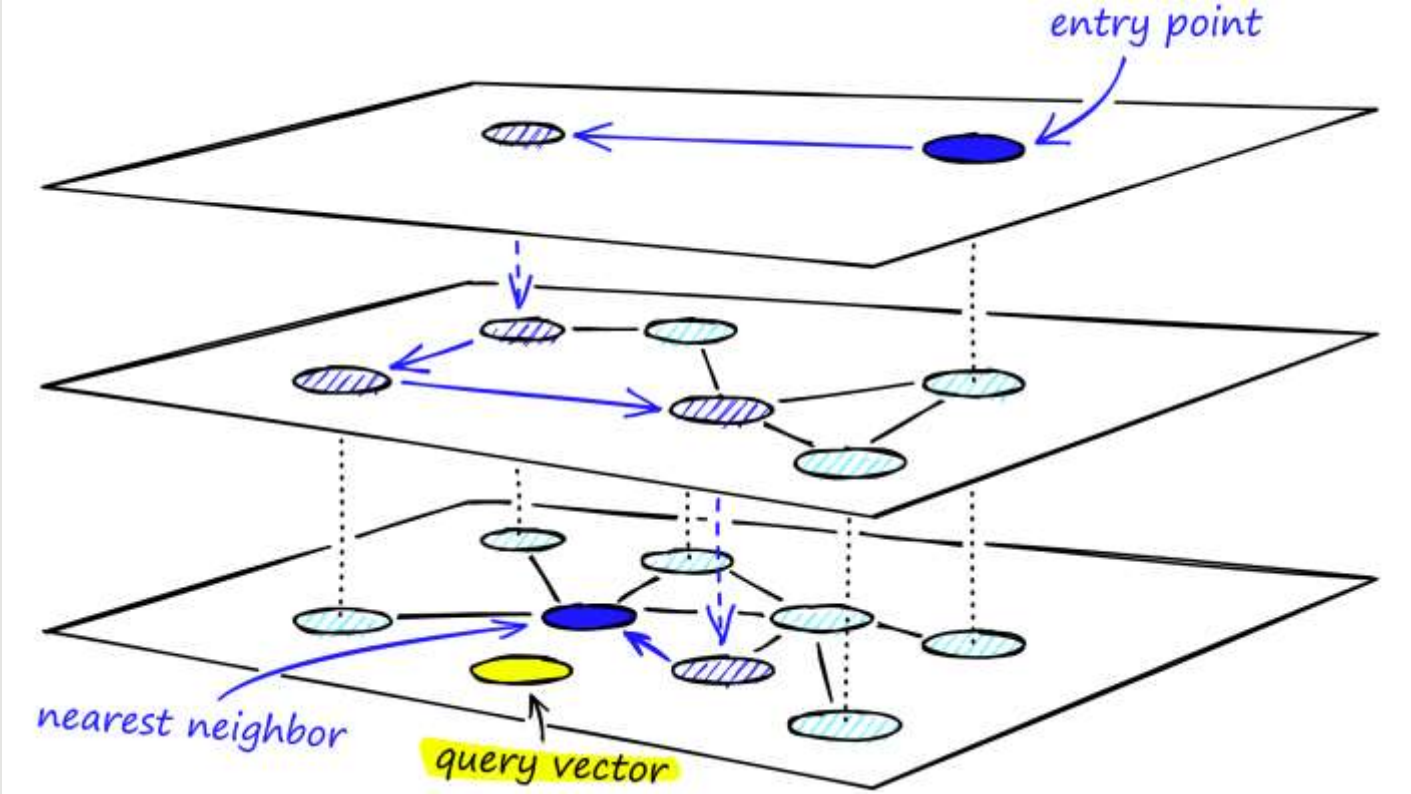
https://arxiv.org/pdf/2310.11703

COMMENT ÇA FONCTIONNE?

HIERARCHICAL NAVIGABLE SMALL WORLD (SOTA)

Construit un **graphe** qui **relie les vecteurs en fonction de leur similarité** ou de leur distance

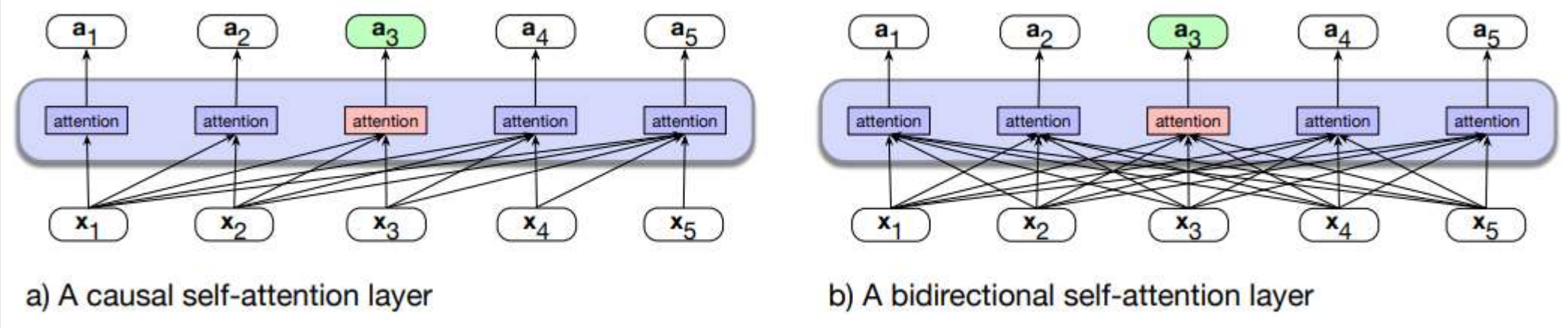
On part d'un **nœud d'entrée**. Ce nœud est relié à plusieurs **nœuds proches**. On navigue **au prochain nœud similaire** jusqu'à trouver le **nœud** le plus proche de notre **requête**.



LE MODÈLE D'EMBEDDING

BERT: BIDIRECTIONAL TRANSFORMER

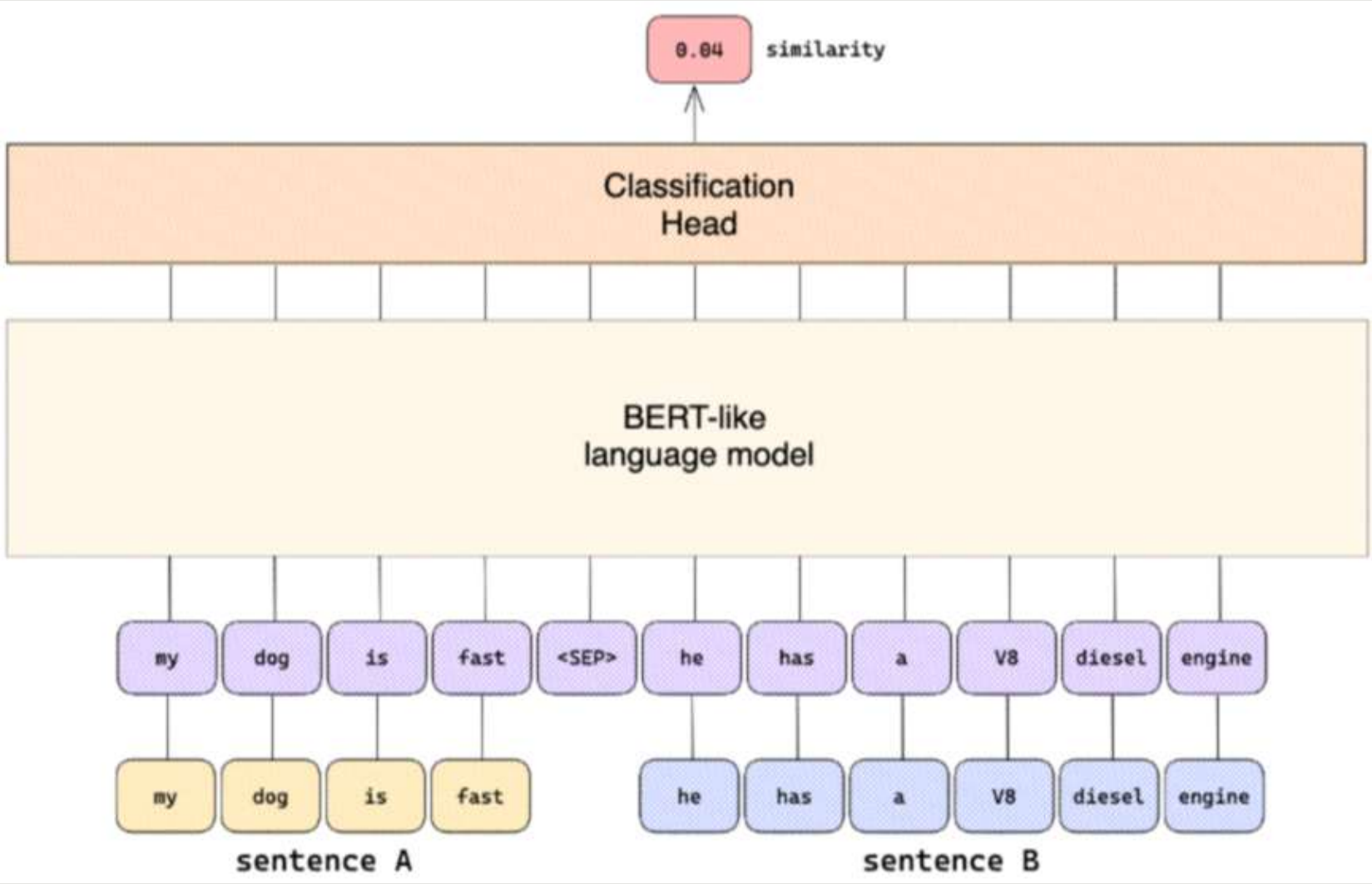
Objectif: **Obtenir une représentation pertinente et contextualisée d'une séquence de tokens.**



Masked Language Modeling

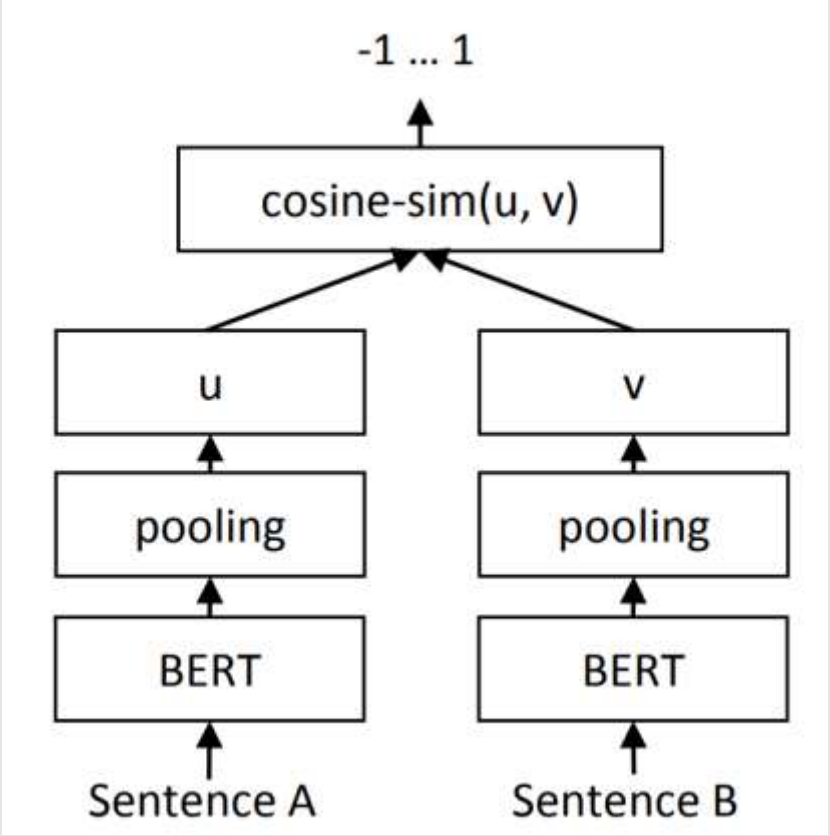
J'ai essayé un MASK Forest, mais il n'est pas performant sur ces données

BERT FOR SEMANTIC SIMILARITY



SENTENCE-BERT – SIAMESE NETWORKS

Optimisation pour la similarité textuelle, la recherche d'information.



QUEL MODÈLE CHOISIR?

English Chinese French Polish Russian

Overall MTEB French leaderboard (F-MTEB) FR

- **Metric:** Various, refer to task tabs
- **Languages:** French
- **Credits:** Lyon-NLP: Gabriel Sequeira, Imene Kerboua, Wissam Siblini, Mathieu Ciancone, Marion Schaeffer

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

Rank	Model	Model Size (Million Parameters)	Memory Usage (GB, fp32)	Embedding Dimensions	Max Tokens	Average (26 datasets)	Classification Average (6 datasets)	Clustering Average (7 datasets)
1	bge-multilingual-gemma2	9242	34.43	3584	8192	70.08	81.62	56.48
2	gte-Qwen2-7B-instruct	7613	28.36	3584	131072	68.25	81.76	55.56
3	gte-Qwen2-1.5B-instruct	1776	6.62	1536	131072	66.6	78.02	55.01
4	KaLM-embedding-multilingual-m	494	1.84	896	131072	64.04	76.83	53.77
5	bilingual-embedding-large	560	2.09	1024	514	62.47	69.16	51.26
6	jina-embeddings-v3	572	2.13	1024	8194	62.29	76.54	44.95
7	jina-embeddings-v3	572	2.13	1024	8194	62.29	76.54	44.95
8	voyage-multilingual-2			1024	32000	61.66	68.56	46.57

BM25 OU SIMILARITÉ COSINE?

BM25

- Perte de l’information du contexte
- Vecteurs creux
- Taille de la matrice

Permet la récupération de termes concrets (entités, etc.)

Cosine Similarity

Perte de l’information de granularité fine (entités, etc.)

- Gain de l’information du contexte
- Vecteurs denses
- Taille de la matrice

LE TD

- Création d’une pipeline de RAG
  - - From Scratch
  - - Avec LangChain