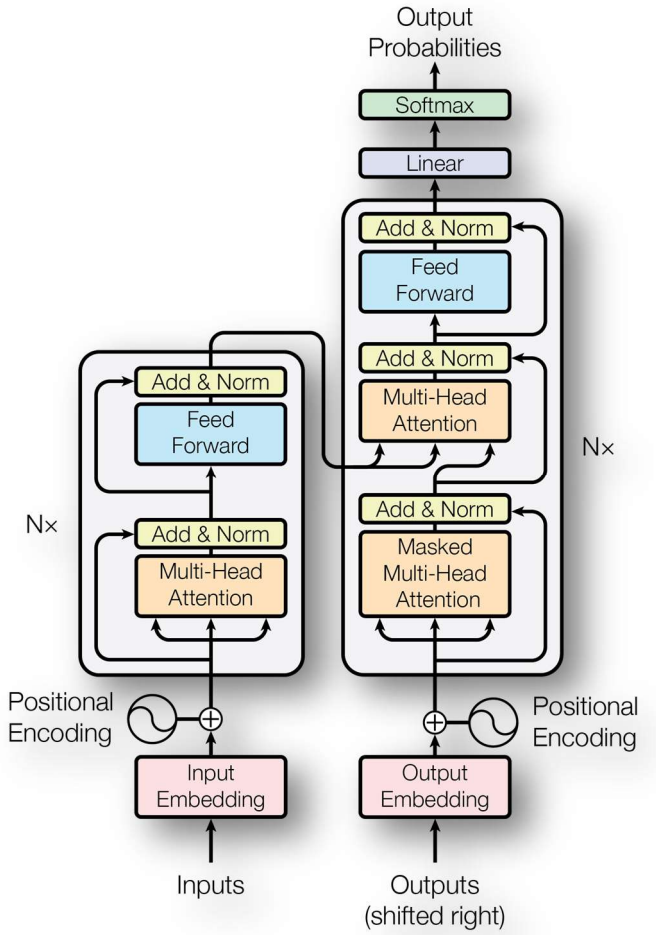
IA génératives à l'Etat de Genève

Mai 2025

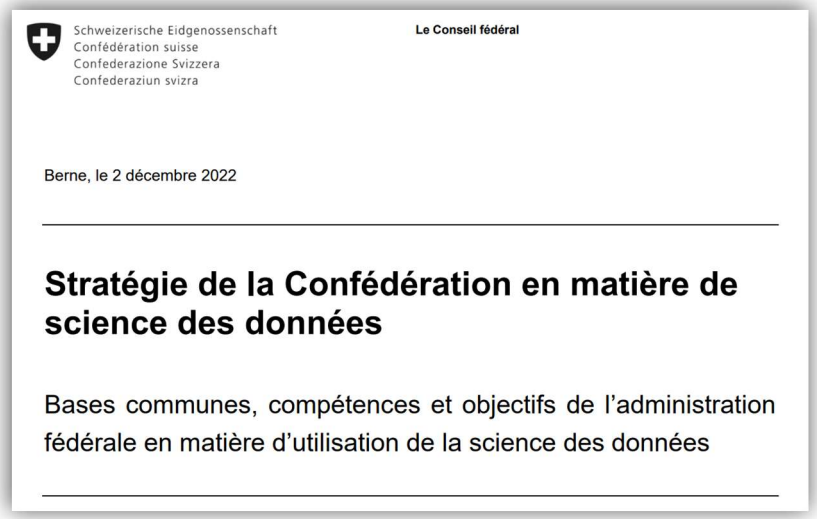


Source: arxiv.org/1706.03762

Sommaire

Organisation des projets d'IA génératives

Le cadre légal fédéral



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Le Conseil fédéral

Berne, le 2 décembre 2022

Stratégie de la Confédération en matière de science des données

Bases communes, compétences et objectifs de l'administration fédérale en matière d'utilisation de la science des données

4.1 Vision

Elle se définit comme suit:

«Une science des données centrée sur l'humain et digne de confiance qui soutient le bien commun ainsi que les politiques publiques»

12 principes fondamentaux:

- ❑ sécurité de l'information
- ❑ protection des données et de l'information
- ❑ sécurité des données
- ❑ gouvernance des données
- ❑ non-discrimination
- ❑ explicabilité
- ❑ traçabilité
- ❑ transparence
- ❑ reproductibilité
- ❑ neutralité
- ❑ objectivité
- ❑ traitement éthique des données et des résultats

Organisation des projets d'IA génératives

La convention du conseil de l'Europe sur l'IA

La Suisse signe la convention du Conseil de l'Europe sur l'IA

Berne, 26.3.2025 - Le 27 mars 2025 à Strasbourg, le conseiller fédéral Albert Rösti signera, au nom de la Suisse, la Convention-cadre du Conseil de l'Europe sur l'intelligence artificielle et les droits de l'homme, la démocratie et l'État de droit. Par sa signature, la Suisse réaffirme son engagement en faveur d'une utilisation des technologies de l'IA à la fois responsable et conforme aux droits fondamentaux.

Principes fondamentaux

- ❑ Dignité humaine et autonomie personnelle
- ❑ Égalité et non-discrimination
- ❑ Respect de la vie privée et protection des données à caractère personnel
- ❑ Transparence et contrôle
- ❑ Obligation de rendre des comptes et responsabilité
- ❑ Fiabilité
- ❑ Innovation sûre

https://rm.coe.int/1680afae3d

https://www.news.admin.ch/fr/nsb?id=101063

Organisation des projets d'IA génératives

Le cadre légal cantonal

Loi constitutionnelle modifiant la constitution de la République et canton de Genève (Cst-GE) **A 2 00**
(Pour une protection forte de l'individu dans l'espace numérique)
(12945)

du 22 septembre 2022

Loi modifiant la loi sur l'information du public, l'accès aux documents et la protection des données personnelles (LIPAD) (13347) **A 2 08**

du 3 mai 2024

Organisation des projets d'IA génératives

Les droits humains dans l'IA

AI & EQUALITY

WOMEN AT THE TABLE

Les droits humains :

- ❑ Corps juridique international (et national) **reconnu**
- ❑ Centré "impact" sur les **êtres humains**
- ❑ Les évaluations d'impact sur les droits humains (**HRIA**) font partie de la **régulation de l'IA** dans le cadre du **règlement européen sur l'IA** (EU AI Act)

Pour garantir :

- ❑ Dignité humaine
- ❑ Liberté et autonomie humaines
- ❑ Prévention des préjudices
- ❑ Équité, non-discrimination, égalité, diversité et inclusion
- ❑ Protection des données et droit à la vie privée
- ❑ État de droit

Organisation des projets d'IA génératives

Les points d'attention

AI & EQUALITY

WOMEN AT THE TABLE

1. Définition des objectifs et composition de l'équipe

- Absence d'objectif clair
- Objectif unique et négatif
- Préjudices sous estimés

2. Définition des exigences du système

- Simplification excessive
- Priorisation des intérêts de l'entreprise
- Précision excessive
- Absence d'identification des biais

3. Exploration des données

- Biais statistiques
- Biais sociétaux

4. Sélection et développement du modèle

- Modèle inadapté
- Variables "nuisibles"

5. Test et interprétation des résultats

- Choix inapproprié des métriques d'évaluation de l'équité
- Focalisation sur la précision

6. Déploiement et évaluation des impacts

- Déploiement précipité
- Conflit d'intérêts sur les impacts
- Peu de monitoring

https://aiequalitytoolbox.com

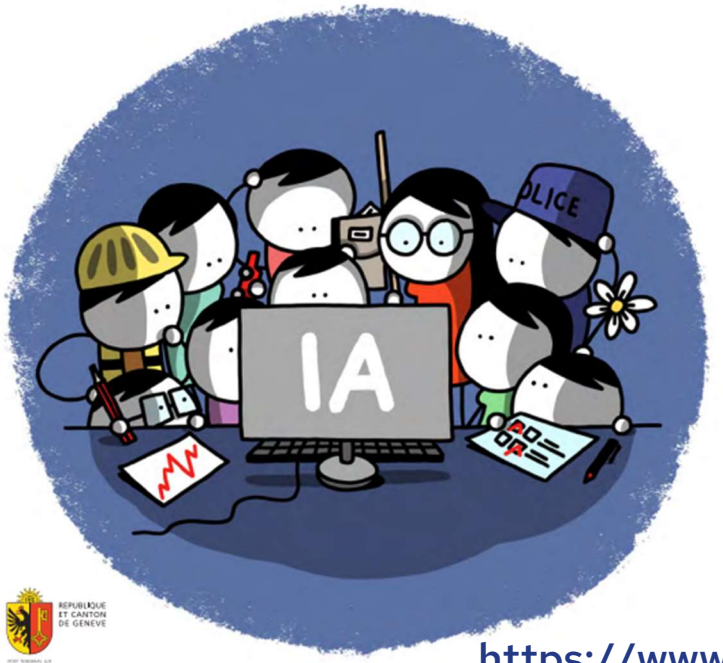
https://community.aiequalitytoolbox.com

Organisation des projets d'IA génératives

Les lignes directrices

L'INTELLIGENCE ARTIFICIELLE À L'ÉTAT

Lignes directrices des services et logiciels numériques en ligne utilisant l'intelligence artificielle (IA) pour les membres du personnel de l'État de Genève



Finalité du traitement	Type de Données à caractère personnel traitées, en fonction de l'activité de traitement
Pour fournir, analyser et maintenir nos Services	• Données liées au Compte • Contenu Utilisateur • Données de Communication • Autres Informations que Vous Fournissez • Données de Connexion • Données d'Utilisation • Informations sur l'Appareil • Données de Localisation • Cookies et Technologies Similaires
Pour améliorer et développer nos Services et mener des recherches	• Données liées au Compte • Contenu Utilisateur • Données de Communication • Autres Informations que Vous Fournissez • Données que nous recevons d'autres sources • Données de Connexion • Données d'Utilisation • Informations sur l'Appareil • Cookies et Technologies Similaires

https://openai.com/policies/privacy-policy/

https://www.ge.ch/document/37368/telecharger

Fondement des IA génératives: transformers

En première approximation, un modèle de langage (LM) est simplement une distribution de probabilité:

Etant donné une séquence de mots $w_{1:t-1} = (w_1, \dots, w_{t-1})$, un LM assigne une probabilité à chaque mot du vocabulaire V pour compléter la séquence:

$$P(w_t | w_{1:t-1}), \quad w_1, \dots, w_{t-1}, w_t \in V.$$

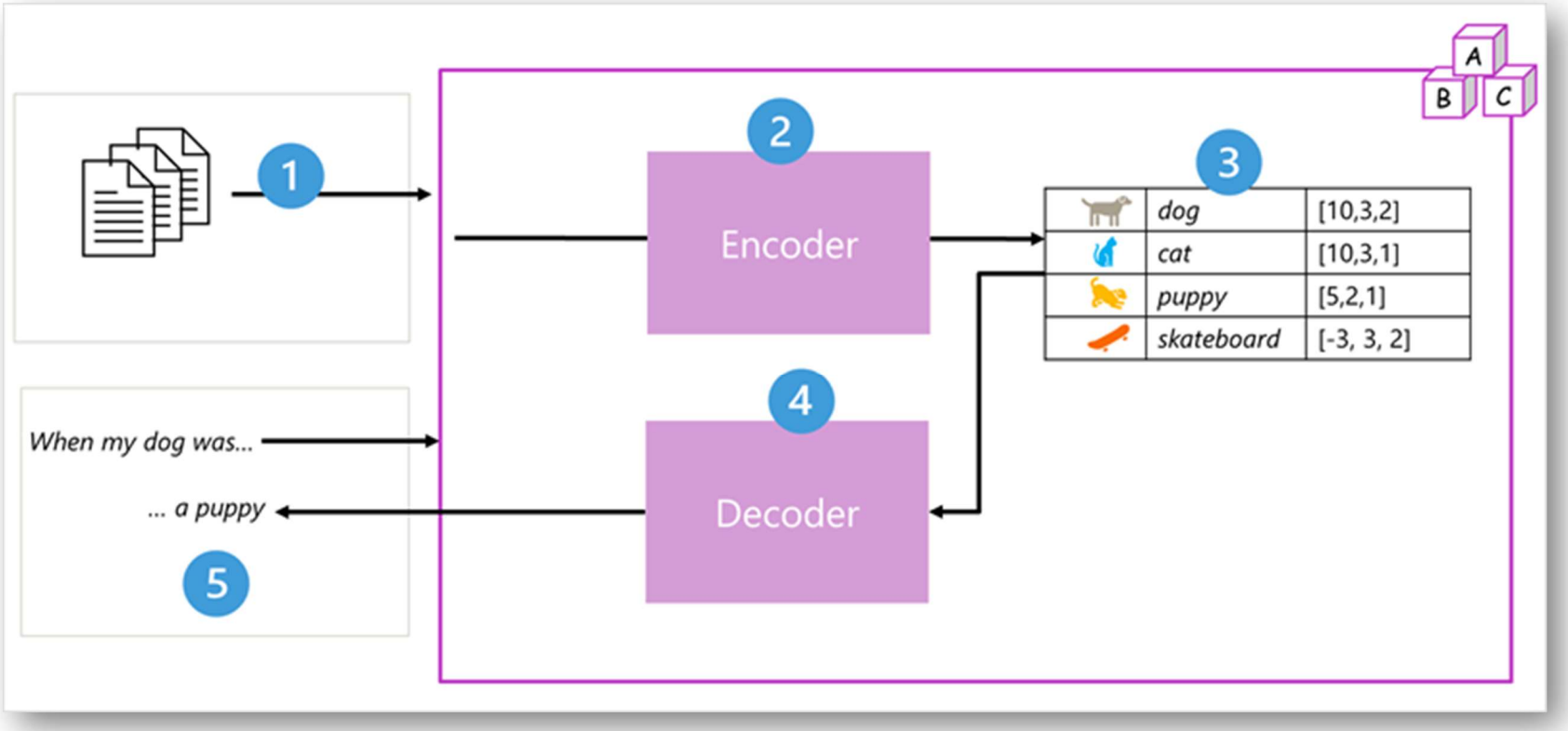
Cette idée est appliquée de manière récursive pour générer des nouveaux textes. Les LM's les plus performants sont basés sur l'architecture deep learning «transformer» introduite en 2017 par Google:

https://arxiv.org/pdf/1706.03762 (Attention Is All You Need)

Fondement des IA génératives: transformers

L'architecture transformer originale consiste en deux blocs:

- ❑ un bloc encodeur qui génère les représentations sémantiques du vocabulaire d'entraînement
- ❑ un bloc décodeur qui génère les nouvelles séquences de texte



Source:
learn.microsoft.com

Fondement des IA génératives: transformers

1. Le modèle est entraîné sur une grande quantité de textes provenant de sources publiques.

2. Les séquences de texte sont décomposées en tokens (par ex. mots) et l'encodeur traite ces séquences de tokens avec la méthode «attention» pour déterminer les relations entre tokens (par ex. quels tokens influencent la présence d'autres tokens dans une séquence).

3. L'output de l'encodeur sont des vecteurs dont les composantes représentent des attributs sémantiques des tokens. Ces vecteurs sont appelés «embeddings» (car le résultat d'un «plongement» en géométrie différentielle).

4. Le décodeur utilise les embeddings produits par l'encodeur pour générer un output en langage naturel.

5. Par exemple, étant donné une séquence input «When my dog was», le modèle emploie l'attention pour analyser les tokens et les attributs sémantiques encodés dans les embeddings pour prédire une complétion appropriée de la phrase, telle que «a puppy».

Fondement des IA génératives: transformers

Tokenisation

La 1ère étape de l'entraînement d'un transformer consiste à décomposer le texte d'entraînement en tokens. Pour simplifier, on considère chaque mot du texte comme un token (en réalité, certains tokens correspondent à des mots partiels, des combinaisons de mots, des ponctuations, etc.)

Pour tokeniser la phrase «I heard a dog bark loudly at a cat», on attribue un token ID à chaque mot. La phrase est ainsi représentée par les tokens {1, 2, 3, 4, 5, 6, 7, 3, 8}.

```
- I (1)
- heard (2)
- a (3)
- dog (4)
- bark (5)
- loudly (6)
- at (7)
- ("a" is already tokenized as 3)
- cat (8)
```

Au fur et à mesure de l'entraînement, on ajoute chaque nouveau token du texte d'entraînement au vocabulaire avec un ID approprié.

→ vocabulaire avec des milliers d'ID's

```
- meow (9)
- skateboard (10)
- *and so on...*
```

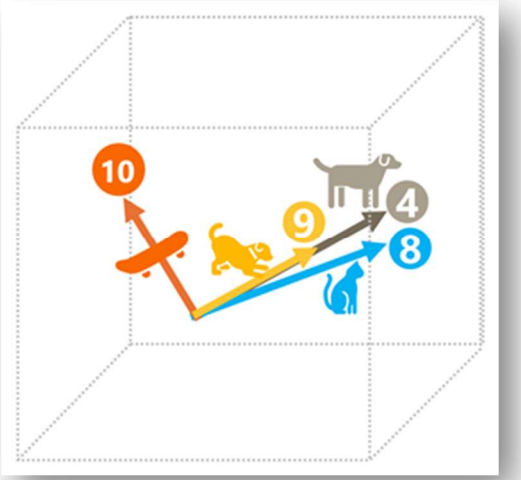
Fondement des IA génératives: transformers

Embeddings

Pour créer un vocabulaire qui encode les relations sémantiques entre tokens, on leur associe des vecteurs (embeddings). Chaque composante de vecteur représente un attribut sémantique de token. Ces attributs sont déterminés lors de l'entraînement, en fonction de la cooccurrence des tokens.

Deux tokens sémantiquement similaires doivent induire deux vecteurs v_1, v_2 pointant dans une même direction. La «cosine similarity» quantifie cette similarité:

$$\textbf{cosine similarity}(v_1, v_2) = \textbf{cosinus de l'angle entre } v_1 \textbf{ et } v_2 = \frac{v_1 \cdot v_2}{||v_1||\,||v_2||}$$




```
- 4 ("dog"): [10,3,2]
- 8 ("cat"): [10,3,1]
- 9 ("puppy"): [5,2,1]
- 10 ("skateboard"): [-3,3,2]
```

Source: learn.microsoft.com

Fondement des IA génératives: transformers

Attention

L'attention est une technique utilisée pour quantifier l'intensité des relations entre les tokens d'une séquence. En particulier, l'auto-attention consiste à déterminer quels autres tokens sont les plus influents pour un token donné lors du traitement d'une séquence.

Dans l'encodeur, chaque token est examiné en contexte, et un encodage approprié est déterminé pour sa vectorisation.

Dans le décodeur, les couches d'attention servent à prédire le token suivant d'une séquence. Pour chaque token généré, le modèle a une couche d'attention qui prend en compte la séquence des tokens jusqu'à ce point. Le modèle identifie les tokens les plus influents pour déterminer le token suivant.

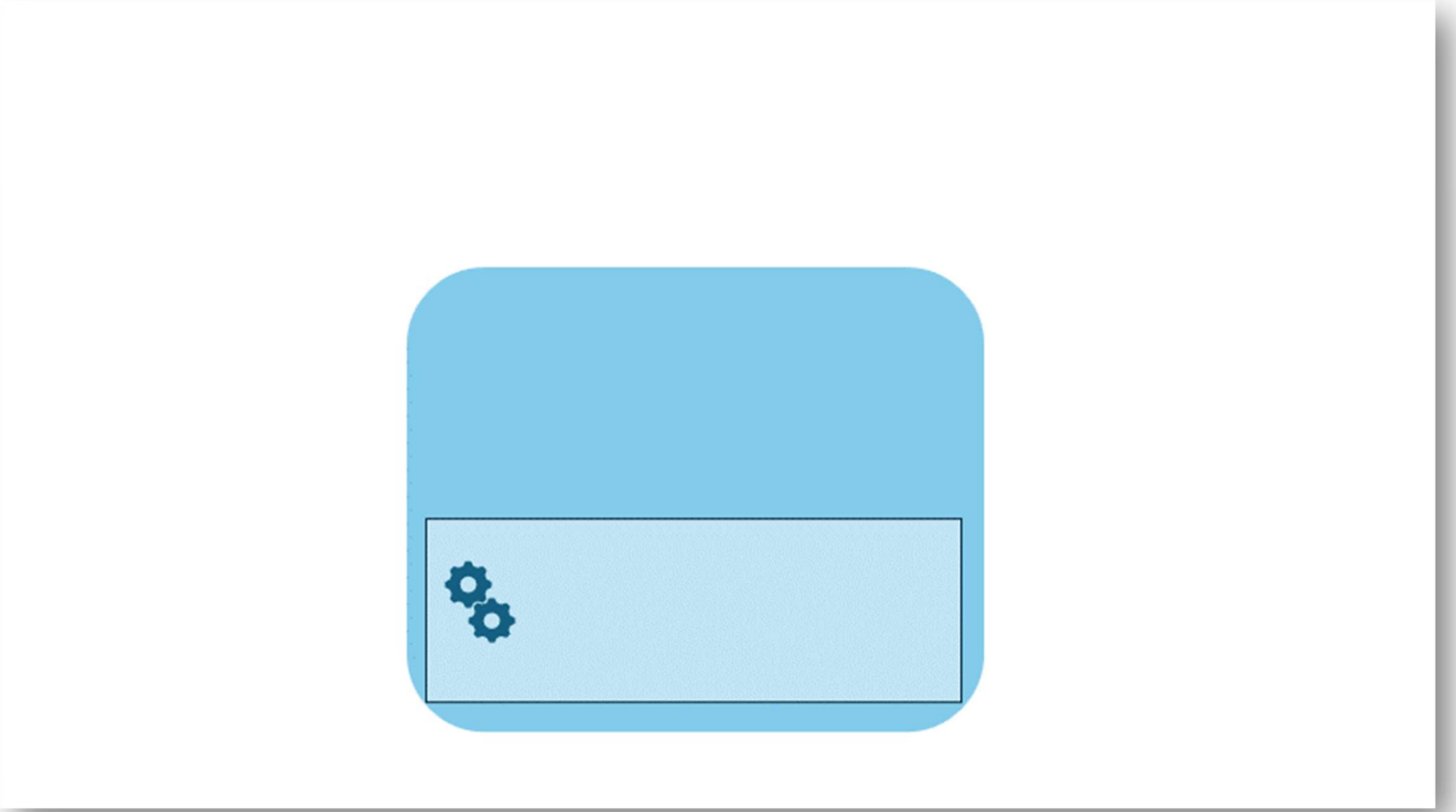
Par exemple, pour la séquence «I heard a dog», la couche d'attention attribue un poids plus important aux tokens «heard » et «dog» pour déterminer le mot suivant de la séquence:

I *heard* a *dog* {*bark*}

Fondement des IA génératives: transformers

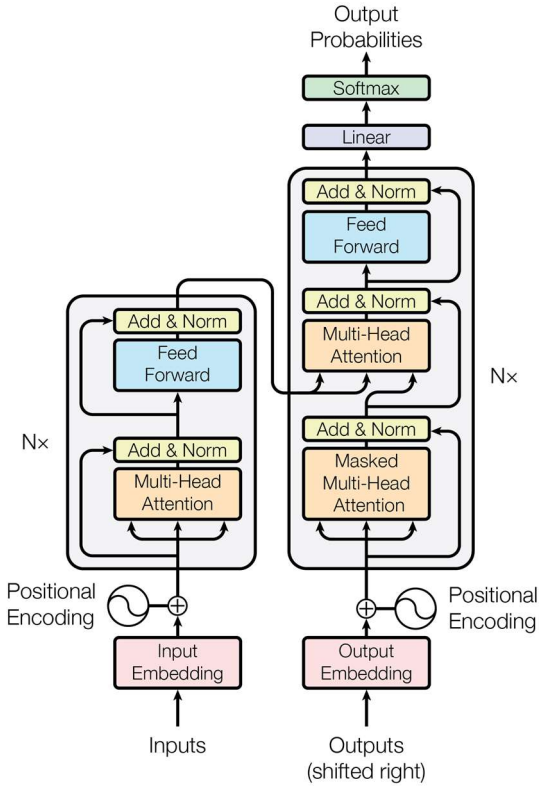
Attention (suite)

L'animation illustre de façon simplifiée ce processus itératif de prédiction du mot suivant:



Fondement des IA génératives: transformers

Dans la pratique, le processus complet est sensiblement plus complexe:



Source: arxiv.org/1706.03762

Pour une description plus détaillée: https://e2eml.school/transformers.html

Pour les mathématiciens: https://arxiv.org/pdf/2410.19370

Architecture des IA génératives: RAG

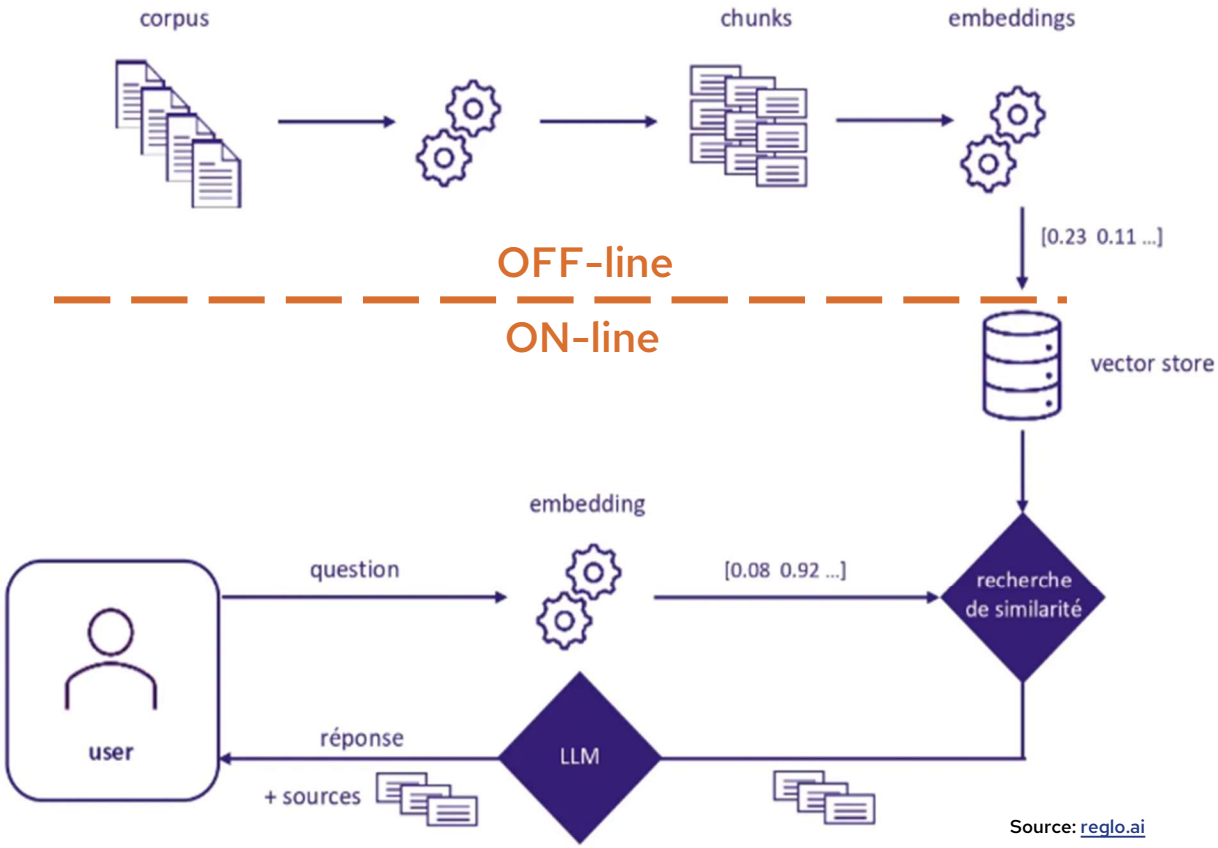
basique RAG

Le fine tunning a des désavantages:

- Coût
- Complexité
- "Hallucination"
- Réponses/informations fournies obsoletes

RAG permet aux modèles de langage:

- d'extraire des informations à partir de sources externes.
- de se spécialiser dans un domaine sans réentrainement.
- d'être actualisés rapidement aux nouvelles données.

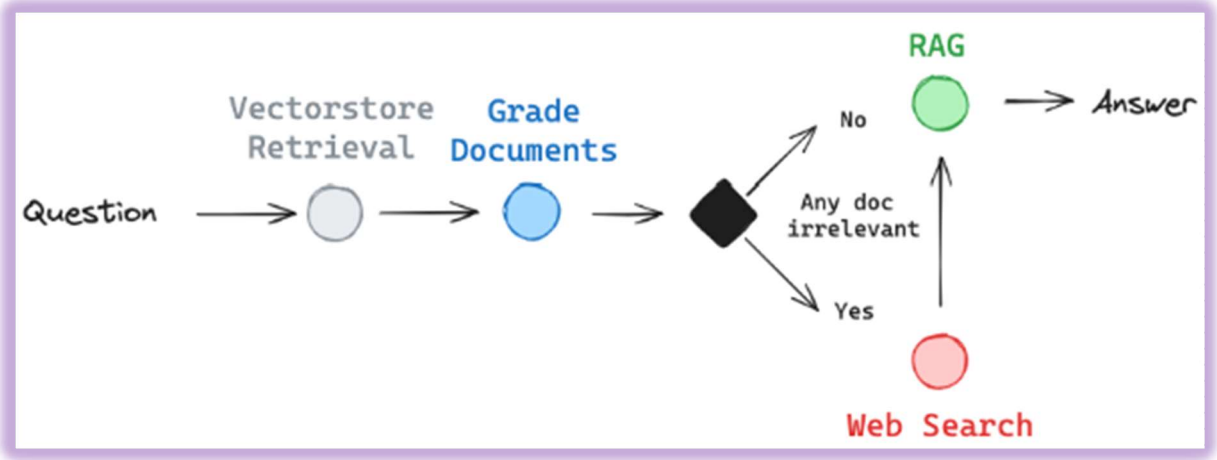


Source: reglo.ai

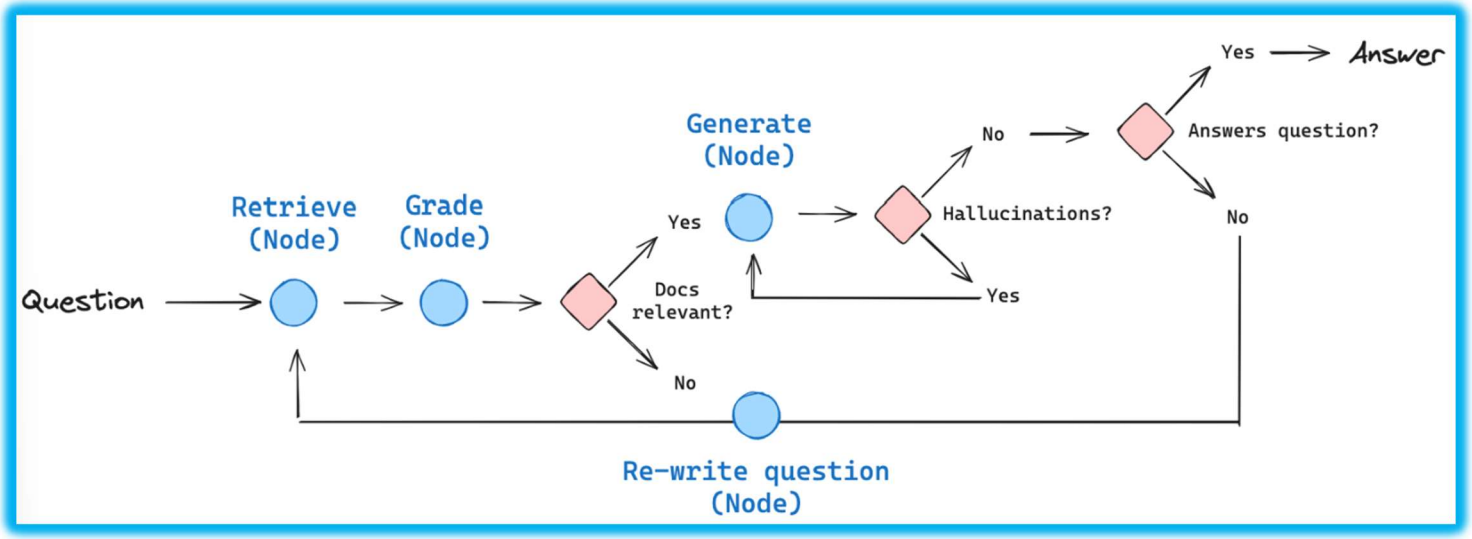
Architecture des IA génératives: RAG

complexe RAG

CRAG

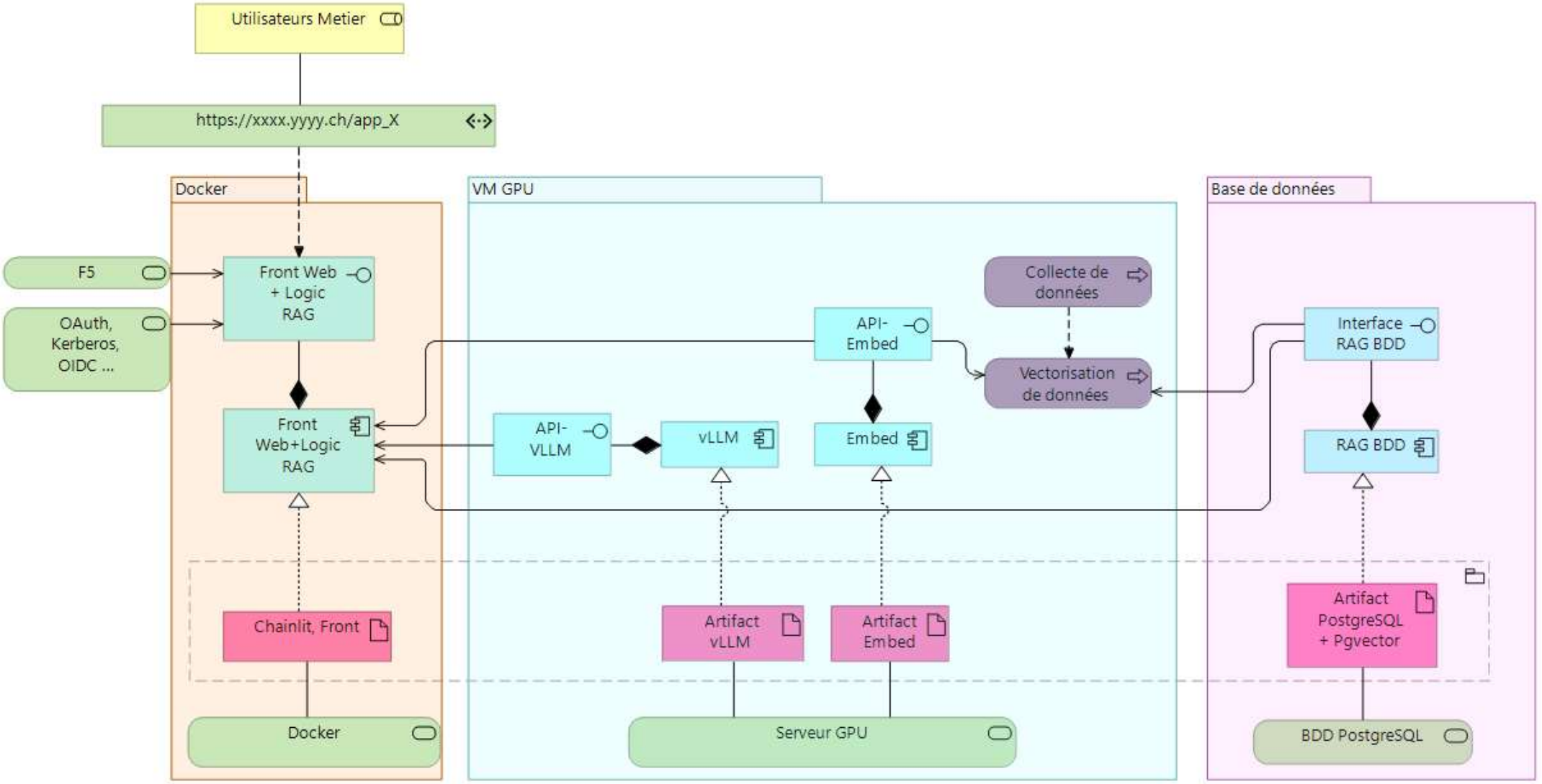


Self-RAG



Source: langchain

Architecture des IA génératives: on-premise

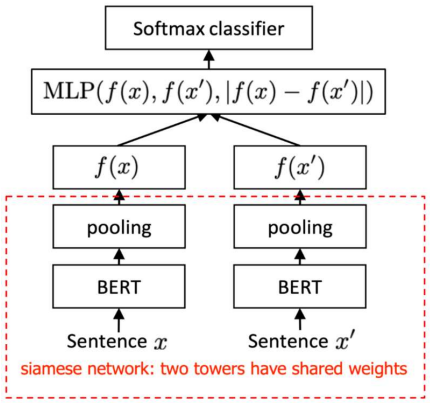


Architecture des IA génératives: Embedding

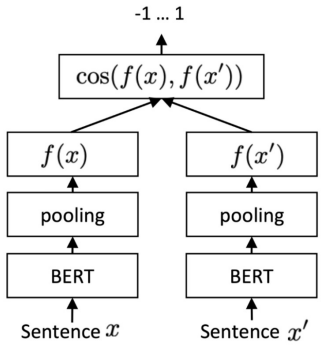
Sentence Transformers

- L' entraînement de BERT sur NSP ne suffit pas à capturer la similarité sémantique entre les phrases.
- Nils Reimers and Iryna Gurevych (article arXiv:1908.10084, 2019) proposent Sentence-BERT (SBERT) basé sur du Siamese et Triplet network (librairie sentence transformers https://sbert.net/)

Classification objective



Regression objective



Triplet objective

$$\max(0, |f(\mathbf{x}) - f(\mathbf{x}^+)| - |f(\mathbf{x}) - f(\mathbf{x}^-)| + \epsilon)$$

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

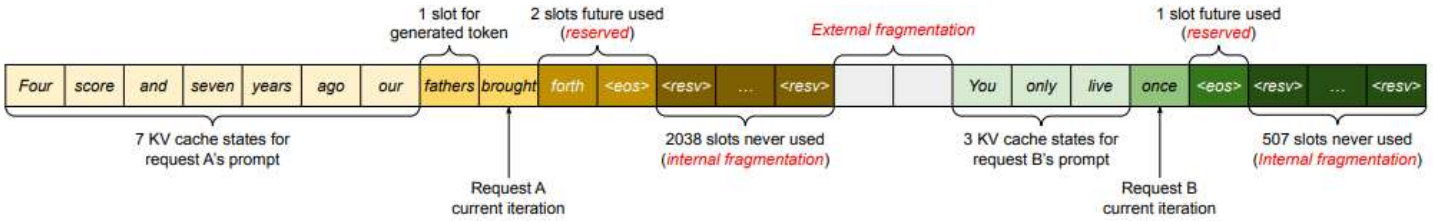
Summary | Performance per task | Task information

Filter...

Rank (Bo…	Model	Zero-shot	Memory U…	Number of P…	Embedding D…	Max Tokens	Mean (T…	Mean (TaskT…	Classification
1	bge-multilingual-gemma2	88%	35254	9B	3584	8192	**70.37**	**68.32**	**81.62**
2	gte-Qwen2-7B-instruct	⚠ NA	29040	7B	3584	32768	69.04	68.31	81.60
3	Linq-Embed-Mistral	100%	13563	7B	4096	32768	67.96	66.17	77.93
4	SFR-Embedding-Mistral	96%	13563	7B	4096	32768	67.16	65.40	73.79
5	e5-mistral-7b-instruct	100%	13563	7B	4096	32768	67.25	65.24	73.34
6	gte-Qwen2-1.5B-instruct	⚠ NA	6776	1B	8960	32768	66.14	65.35	77.83
7	SFR-Embedding-2_R	96%	13563	7B	4096	32768	67.82	64.93	79.32

Architecture des IA génératives: vLLM

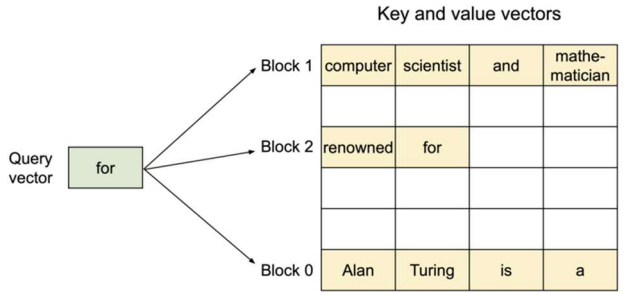
Préallocation de la mémoire avec stockage contigu



Source: arxiv 2309.06180

- **Réduction de la fragmentation mémoire** :
  - le mécanisme de PagedAttention minimise la perte de mémoire à moins de 4 % contre 60–80 % dans les systèmes traditionnels.(arxiv 2309.06180)
- **Partage efficace de la mémoire** :
  - évite des calculs redondants et économise de la mémoire, améliorant le débit global du système.
- **Prise en charge de la quantification** :
  - permettant l'optimisation de la performances du modèle et de réduire son empreinte mémoire.

Allocation dynamique de la mémoire par block avec stockage non-contigu



Mécanisme du PagedAttention

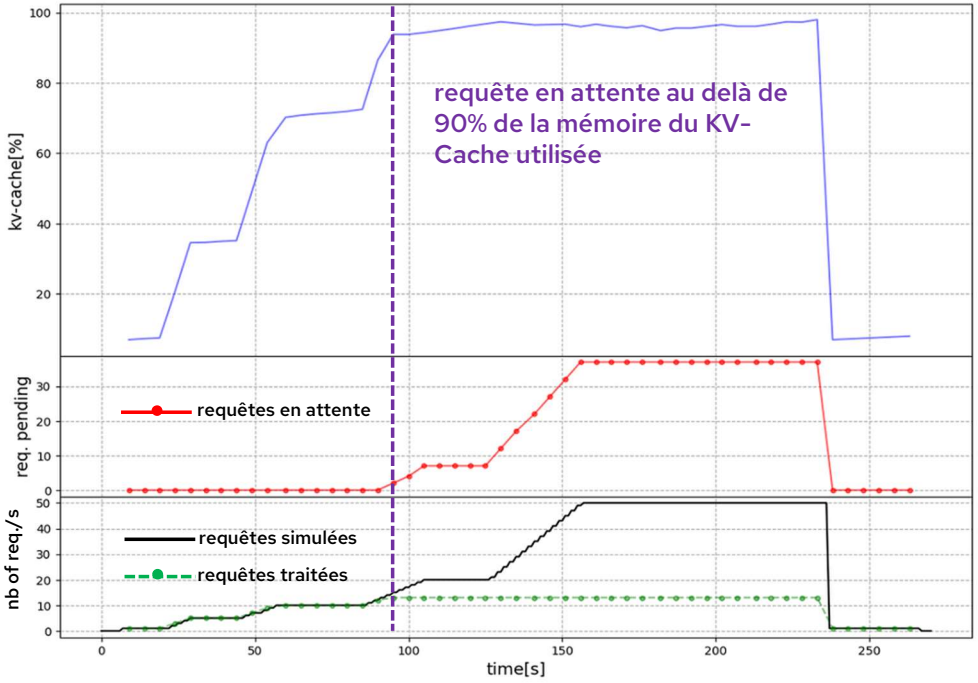
Source: vllm-blog

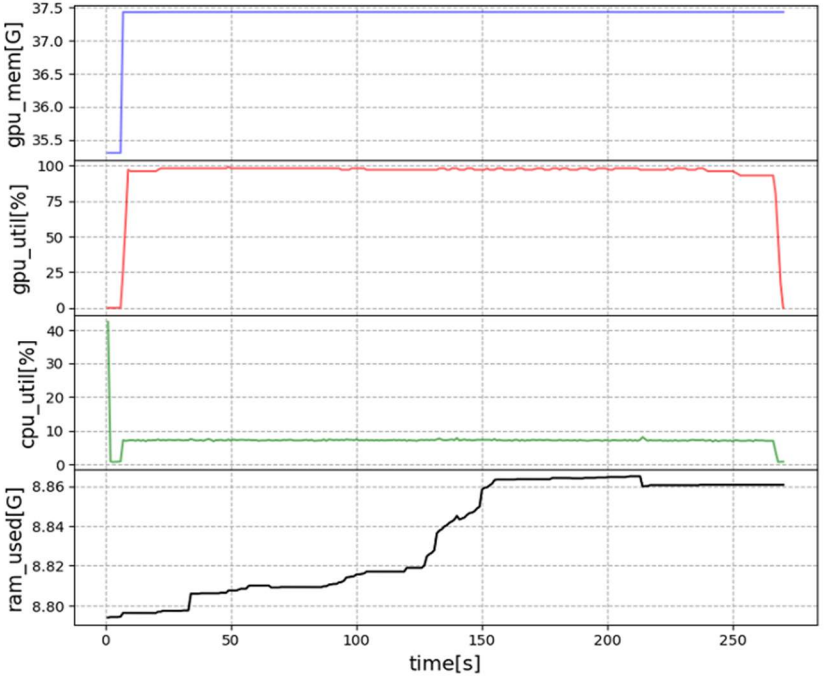
Architecture des IA génératives: GPU H100

Chargement du model Mistral-Small-24B-Base-2501-bnb-4bit

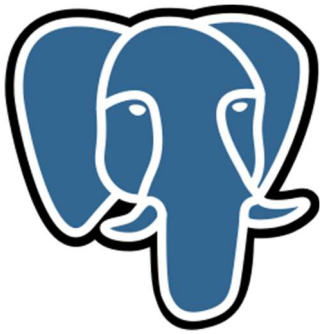
```
INFO 04-23 17:40:41 model_runner.py:1115] Loading model weights took 13.2274 GB
INFO 04-23 17:40:45 worker.py:267] Memory profiling takes 4.10 seconds
INFO 04-23 17:40:45 worker.py:267] the current vLLM instance can use total_gpu_memory (93.65GiB) x gpu_memory_utilization (0.40) = 37.46GiB
INFO 04-23 17:40:45 worker.py:267] model weights take 13.23GiB; non_torch_memory takes 0.15GiB; PyTorch activation peak memory takes 7.28GiB; the rest
of the memory reserved for KV Cache is 16.80GiB.
INFO 04-23 17:40:45 executor_base.py:111] # cuda blocks: 6880, # CPU blocks: 1638
INFO 04-23 17:40:45 executor_base.py:116] Maximum concurrency for 32768 tokens per request: 3.36x
INFO 04-23 17:41:02 api_server.py:958] Starting vLLM API server on http://0.0.0.0:8800
```

Exemple de test de charge



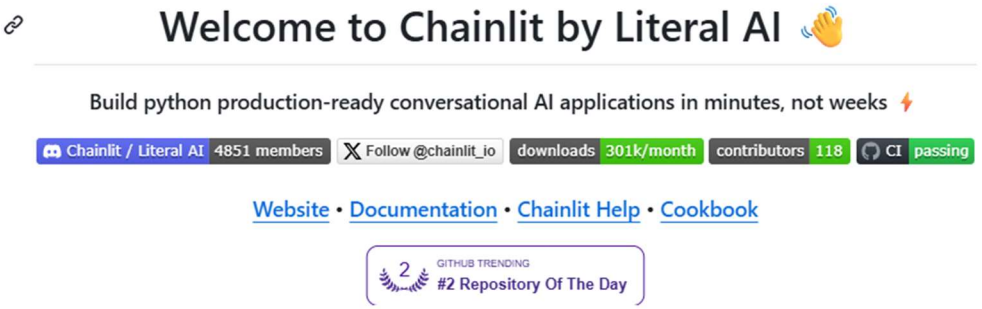


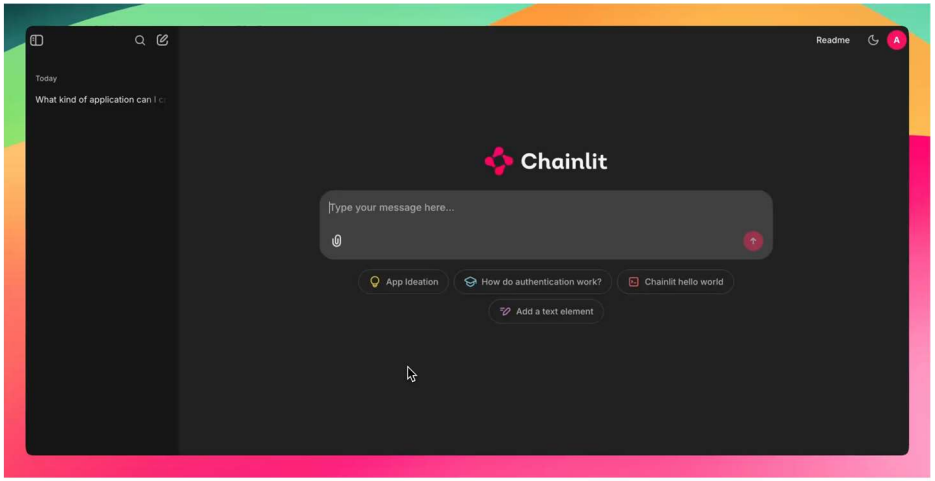
Architecture des IA génératives: Pgvector



- Intégration native à PostgreSQL
- Open Source, sans verrou propriétaire
- Écosystème mature et robuste
- Large communauté et adoption
- Requêtes hybrides riches
- Performance et scalabilité croissantes

Architecture des IA génératives: Chainlit





➢ Qu'est-ce que Chainlit ?

- Framework **open source** proposé par **Literal AI** pour créer des interfaces web similaires à ChatGPT, basées sur des agents LLM.
- Possibilité de personnaliser le front-end
- Permet de transformer un script Python en une application LLM interactive, sans effort de front-end.
- Idéal pour faciliter le **prototypage et le déploiement rapide** d'applications LLM.

➢ Fonctionnalité clés

- **Authentification**: Simple authentification par login/password, OAuth et Header (via reverse proxy)
- **Isolation des sessions**: Chaque utilisateur dispose d'une session isolée.
- **Persistance des données** : Collecte et sauvegarde des données pour retourner aux sessions passées ou analyser les retours utilisateurs.
- **Multimodalité**: Chainlit permet d'accéder au flux audio du microphone de l'utilisateur et de le traiter en temps réel.
- **Visualisation du raisonnement** intermédiaire ayant conduit à une réponse.

Merci pour votre attention

William Creus
Lionel Jasinski
Rafael Tiedra

william.creus @etat.ge.ch
lionel.jasinski@etat.ge.ch
rafael.tiedra@etat.ge.ch