

Etude scientifique et technique

L'architecture RAG comme alternative au ré-entraînement
des modèles génératifs



Problématique :

“En quoi le RAG permet de compléter l'utilisation d'un modèle génératif ?”

Sommaire :

- Les limites des modèles génératifs
- Théorie et technique du RAG
- Cas pratique du RAG en complément d'un modèle génératif



Les limites de l'IA Générative



Obsolescence ("Knowledge Cutoff") - Connaissances figées dans le temps



Hallucinations



Amnésie métier - Aucune connaissances de données interne



RAG et modèle génératifs - Changement de paradigme

R.A.G : Retrieval-Augmented Generation (Génération Augmentée par la Recherche)

L'idée : Ne plus forcer l'IA à tout savoir, mais lui donner accès à un moteur de recherche externe.

Différence de vision

Approche	Concept	Rôle
Entrainement de modèle	Apprendre par coeur	Mémoire + raisonnement
RAG	Lecture de donnée en temps réel	Logique et raisonnement

Le RAG à une vision différente, il fournit la vérité factuelle brute en contexte



Comment fonctionne le RAG - Partie 1 (Phase d'ingestion)



Sources de Données

PDF, SQL, Images

Phase 1 : Extraction



Vectorisation

Modèle d'Embedding (Traduction en espace vectoriel)

Phase 2 : Transformation



Découpage (Chunking)

Segmentation pour garder des bouts bien découpés de la donnée

Phase 2 : Transformation



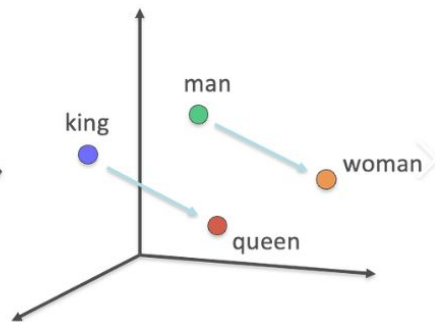
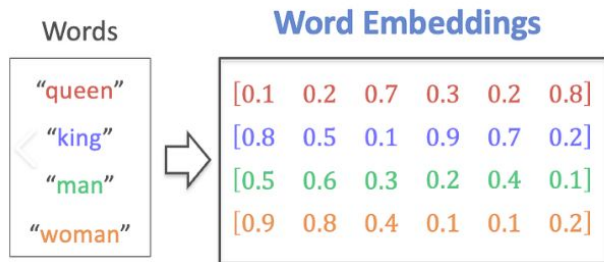
Indexation & Stockage

Bases Vectorielles (FAISS) ou
Relationnelles (H2/SQLite)

Phase 3 : Load

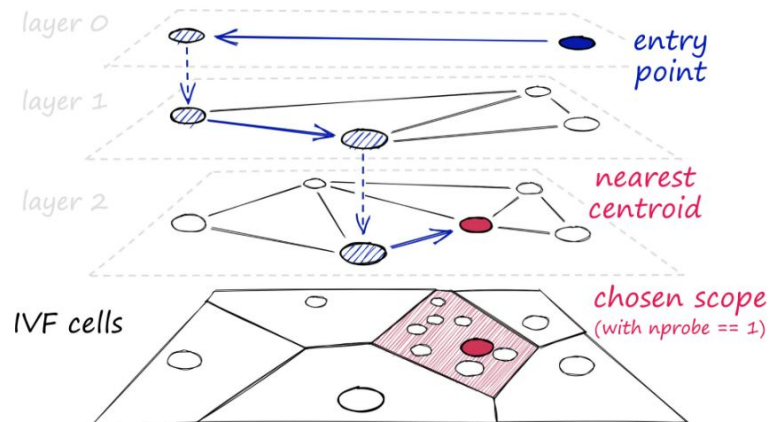


Explication plus approfondie de la vectorisation et indexation



Vectorisation

Indexation





Comment fonctionne le RAG - Partie 2 (Phase d'inférence)



Requête Utilisateur

Question posée en langage naturel

Étape 1 : Entrée



Recherche (Retrieval)

Recherche de similarité dans l'index FAISS
ou base SQL pour extraire le contexte

Étape 3 : Extraction contextuelle



Prompt Augmenté

Assemblage strict :
"Question" + "Contexte trouvé" sous
forme ultra strict limitant l'hallucination

Étape 5 : Augmentation



Vectorisation

La question est transformée en vecteur
(Même modèle d'Embedding qu'à
l'ingestion)

Étape 2 : Traitement



Reranking (Tri)

Réévaluation et tri des documents pour ne
garder que les plus pertinents

Étape 4 : Optimisation



Génération (LLM)

Inférence locale et réponse streamée à
l'utilisateur

Étape 6 : Sortie



Projet entreprise - Assistant ChatBot

Besoin :

Créer une interface ChatBot qui aide les techniciens du support pour la résolution des tickets en se basant sur les ressources internes (documentations, tickets).

Contraintes :

- Environnement complètement isolé et sécurisé
- Aucune hallucination
- Possibilité d'ajouter continuellement de nouveaux documents
- Pouvoir utiliser tout type de documentation technique



DATA & IA

Contexte dynamique & Moteurs de recherche (Vectoriel / SQL).

Concurrence

Gestion de la concurrence & Tâches asynchrones.

Logiciel

Déploiement d'API REST (Java/FastApi)

Architecture

Séparation des responsabilités (MVC)





Rappel problématique :

“En quoi le RAG permet de compléter l'utilisation d'un modèle génératif ?”

Bilan :

- **Changement de rôle** : Le modèle génératif sert uniquement de moteur de raisonnement et de formatage.
- **Extériorisation de la vérité** : La "mémoire" est stockée dans une base contrôlable (FAISS ou SQL).
- **Traçabilité & Confiance** : Citation des sources exactes pour éviter l'effet "boîte noire".
- **Agilité opérationnelle** : Mise à jour des connaissances en temps réel sans ré-entraînement.

Sources :

- **Amazon Web Services (AWS)**. *Qu'est-ce que la génération à enrichissement contextuel (RAG) ?*. Documentation officielle, aws.amazon.com.
- **Zhao, P. et al. (2024)**. *A Comprehensive Review of Retrieval-Augmented Generation (RAG): Key Challenges and Future Directions*. arXiv preprint.