INTRODUCTION

L'intelligence artificielle connaît une révolution silencieuse mais puissante grâce aux systèmes dits Retrieval-Augmented Generation (RAG). Contrairement aux modèles traditionnels qui répondent à partir de leur mémoire interne, les modèles RAG vont chercher dans une base documentaire externe pour générer des réponses plus précises, plus actuelles, et souvent plus fiables. En d'autres termes, l'IA "lit" avant de parler.

Le principe est simple mais redoutablement efficace : on interroge d'abord une base de documents à l'aide d'un moteur de recherche sémantique (retrieval), puis on injecte les résultats dans un modèle génératif (génération) qui rédige une réponse en s'appuyant sur ces contenus. Cette combinaison permet d'allier mémoire fraîche et capacité de synthèse.

Cependant, cette méthode n'est pas figée. Plusieurs variantes sont apparues pour combler les limites du RAG standard, améliorer sa rigueur ou l'adapter à des cas d'usage spécifiques. Chaque méthode apporte des améliorations… mais aussi ses propres défis.

Dans ce rapport, nous allons comparer quatre approches clés :

- GraphRAG qui s'appuie sur des graphes de connaissances pour structurer l'information
- CAG (Cache Augmented Generation) qui tient compte du contexte élargi (conversation, historique, domaine) pour améliorer la pertinence des réponses
- CRAG (Correction-RAG) qui ajoute une étape de vérification des sources
- AgenticRAG qui repose sur des agents autonomes pour explorer différentes pistes

Nous analyserons leurs fonctionnements, avantages, limites et nous établirons un tableau comparatif qui permettra de visualiser leurs différences essentielles. En conclusion, nous proposerons une synthèse pour aider à choisir la meilleure approche adaptée à notre projet .

1. PRESENTATION DES METHODES

a. GraphRAG (Graph-based Retrieval-Augmented Generation)

GraphRAG est une version améliorée des systèmes RAG. Au lieu de lire les documents comme des textes normaux, il recherche les liens cachés entre les informations, comme les connexions entre des lois, des événements ou des idées.

Il transforme les textes en graphes : un graphe, c'est comme un réseau avec des points (les idées principales) et des lignes (les liens entre elles). Grâce à ça, GraphRAG peut mieux comprendre les relations entre les sujets, et donc donner des réponses plus logiques et précises.

Principe de fonctionnement :

Construction du graphe

Dès qu'un corpus de documents est fourni (textes juridiques, médicaux, réglementaires…), GraphRAG extrait les informations importantes et construit un **graphe** : un réseau où chaque idée est reliée à d'autres.

Par exemple, dans le domaine médical :

"Diabète" → "Traitement" → "Metformine" → "Effets secondaires"

Organisation par thèmes

Les idées liées sont automatiquement regroupées en **blocs thématiques**. Cela permet de structurer les informations (ex : "Symptômes", "Traitements", "Articles de loi") et de créer des résumés pour chaque groupe. Cela évite les répétitions et clarifie le raisonnement.

Réponse guidée par le graphe

Lorsqu'une question est posée, le modèle **navigue dans le graphe** pour suivre les liens logiques et sélectionner les éléments les plus utiles.

Il peut ainsi expliquer un sujet complexe de manière fluide et cohérente, en reliant les bonnes informations entre elles, comme un GPS de la connaissance.

b. CAG (Cache Augmented Generation)

La **génération augmentée par cache**, ou CAG, est une méthode qui permet aux modèles d'intelligence artificielle de répondre plus vite et plus efficacement, en préparant les informations à l'avance. Au lieu d'aller chercher des documents à chaque fois qu'on pose une question, le modèle dispose déjà des connaissances nécessaires directement dans sa mémoire. Cette approche est très utile pour les sujets stables et répétitifs, comme les lois, les procédures ou les documents de référence.

Préchargement des connaissances : Avant que l'utilisateur ne pose une question, on sélectionne un ensemble de documents utiles : textes de loi, guides pratiques, fiches santé, ou tout autre contenu pertinent. Ces documents sont ensuite organisés et formatés pour être lisibles par le modèle. Ils sont injectés dans la mémoire active du

modèle, appelée fenêtre de contexte, comme s'ils étaient déjà ouverts devant lui. Cela permet au modèle d'avoir tout sous les yeux au moment de répondre.

Création d'un cache interne : Une fois les documents chargés, le modèle les lit une première fois et garde en mémoire ce qu'il comprend. Il enregistre ces informations dans une sorte de mémoire technique qu'on appelle un cache. Ce cache permet d'éviter que le modèle doive tout relire à chaque fois. Il garde ainsi en réserve des éléments qu'il pourra réutiliser plus tard, ce qui réduit les temps de traitement.

Stockage du cache : Le cache peut être conservé en mémoire ou enregistré sur disque, selon les besoins. Cela permet à plusieurs questions d'accéder aux mêmes données préparées, sans devoir les traiter à nouveau. Par exemple, si plusieurs utilisateurs posent des questions sur le même texte de loi, le modèle pourra répondre rapidement, car tout est déjà prêt.

Utilisation du cache pour générer les réponses : Lorsqu'une requête est posée, le modèle utilise directement les informations du cache, combinées avec la question, pour construire une réponse. Cela permet de gagner du temps tout en maintenant un haut niveau de précision. Le modèle n'a pas besoin de récupérer des documents ailleurs ni de recalculer les mêmes choses plusieurs fois**.**

Réinitialisation du cache (facultatif) : Si le cache devient trop lourd ou inutile, il peut être réinitialisé. Cela signifie que certaines informations peuvent être supprimées pour libérer de l'espace ou pour repartir sur de nouvelles bases. Ce nettoyage permet de maintenir de bonnes performances, même dans des systèmes qui tournent longtemps ou qui reçoivent beaucoup de questions.

c. CRAG (Correction-RAG)

CRAG (Correctif RAG) représente une avancée majeure pour les assistants conversationnels. Cette innovation permet aux modèles de langage comme ChatGPT de fournir des réponses plus exactes et mieux adaptées. Le système agit comme un filtre intelligent, garantissant la qualité de chaque information délivrée

CRAG fonctionne comme un système de contrôle qualité pour les réponses des chatbots. Voici comment cela opère :

Évaluation initiale des sources
Le système commence par analyser toutes les informations disponibles sur un sujet donné. Chaque source reçoit une note de fiabilité basée sur sa provenance, sa date et sa cohérence avec d'autres données connues. Seules les sources jugées suffisamment fiables sont conservées pour la suite du processus.

Vérification croisée automatique
Pour chaque information cruciale, CRAG effectue une double vérification en consultant plusieurs bases de données et sources officielles. Cette étape permet de détecter et d'éliminer les contradictions ou les données obsolètes. Le système compare systématiquement les différentes versions d'une même information.

Adaptation contextuelle
En fonction du profil de l'utilisateur et de l'historique de la conversation, CRAG ajuste le format de la réponse. Cette personnalisation prend en compte le niveau de connaissance supposé de l'interlocuteur, la complexité du sujet et le contexte général de l'échange.

Contrôle final de cohérence
Avant d'envoyer la réponse, le système vérifie une dernière fois que toutes les informations fournies sont cohérentes entre elles et avec les connaissances actuelles. Cette étape finale permet de repérer et de corriger d'éventuelles incohérences résiduelles.

Mise à jour continue
CRAG intègre en permanence de nouvelles informations et mises à jour, garantissant ainsi que les réponses restent exactes dans le temps. Le système apprend également de ses interactions pour améliorer constamment ses processus de vérification.

d. AgenticRAG (Agentic Retrieval-Augmented Generation)

AgenticRAG représente une évolution majeure dans le domaine des intelligences artificielles conversationnelles. Ce système innovant fonctionne comme une véritable équipe virtuelle, où plusieurs agents spécialisés collaborent pour fournir des réponses complètes et précises. Contrairement aux modèles traditionnels qui tentent de tout gérer seuls, AgenticRAG répartit intelligemment le travail entre différents experts virtuels.
Le système repose sur une architecture modulaire composée de plusieurs agents autonomes. Chaque agent possède une spécialisation bien définie : recherche documentaire, vérification des données, calculs complexes ou formulation des réponses. Lorsqu'une question est posée, un agent coordinateur analyse la requête et la décompose en sous-tâches qu'il distribue aux agents compétents. Cette approche permet un traitement plus approfondi et plus fiable des demandes complexes.
Principe de fonctionnement :

- **Phase d'analyse et planification :** Le système commence par décomposer la question utilisateur en éléments constitutifs. Un agent coordinateur identifie les compétences requises et établit un plan d'action. Par exemple, pour une question juridique complexe, il déterminera qu'il faut : vérifier les textes de loi,

analyser la jurisprudence récente et calculer d'éventuelles indemnités. Cette phase cruciale permet d'orienter efficacement le travail des agents spécialisés.

- **Phase d'Exécution Distribuée :** Chaque agent travaille indépendamment sur sa partie du problème :

➔ L'agent "Recherche" explore les bases documentaires
➔ L'agent "Vérification" consulte les sources officielles
➔ L'agent "Calcul" effectue les opérations nécessaire

Tous fonctionnent en parallèle, ce qui accélère considérablement le processus par rapport à une approche séquentielle. Les agents peuvent interagir entre eux pour affiner leurs recherches si nécessaire.

- **Phase de synthèse et validation :** Les résultats partiels sont rassemblés et harmonisés. Un agent dédié vérifie la cohérence globale de la réponse, élimine les éventuelles contradictions et adapte le niveau de détail à l'utilisateur. La réponse finale intègre ainsi toutes les dimensions du problème tout en restant claire et accessible. Le système peut également enregistrer les nouvelles connaissances acquises pour améliorer ses futures interventions.

2. **Avantages et limites des méthodes RAG**

a. **Avantages**

- **GraphRAG (Graph-based Retrieval-Augmented Generation)**

Modélisation avancée des connaissances : GraphRAG transforme les documents en graphes structurés où Les **concepts clés** deviennent des nœuds (personnes, lieux, termes techniques) et Les **relations sémantiques** forment des arêtes (causalité, référence, similarité)

Mécanisme d'exploitation contextuelle : Le système construit dynamiquement un réseau de connaissances afin de naviguer intelligemment dans ce graphe et Identifie les chemins pertinents entre concepts.

Exemple
Prenons le cas du domaine médical , il Connecte automatiquement : Un symptôme (fièvre) aux diagnostics possibles (paludisme, infection) En passant par les examens pertinents (NFS, frottis) .

Le GraphRAG a la capacité de précision accrue en détectant des liens non évidents en analyse textuelle classique , il s'adapte divers domaines experts (droit , finance , recherche scientifique) . Il est généralement utiliser pour la recherche documentaire complexe , l'analyse de réseaux d'information et pour des question nécessitant un raisonnement multi-étapes

- **CAG (Cache Augmented Generation)**

Réponses plus rapides : Grâce au préchargement des connaissances et à la mise en cache, le modèle n'a pas besoin de relancer des recherches ou de tout recalculer à chaque fois. Cela permet d'accélérer considérablement le temps de réponse, même pour des questions complexes.

Moins de ressources utilisées : *Comme les documents sont déjà disponibles dans le contexte du modèle, aucun moteur de recherche externe (comme une base vectorielle) n'est nécessaire. Cela rend l'architecture plus légère, plus simple à déployer, et moins coûteuse à faire tourner.*

Meilleure stabilité des réponses : Les données injectées dans le cache ne changent pas pendant une session. Cela garantit une cohérence dans les réponses, même quand l'utilisateur pose plusieurs questions liées. Le modèle s'appuie toujours sur la même base de connaissances.

- **CRAG (Correction-RAG)**

Correction Automatique des Erreurs dans les Données Récupérées : CRAG intègre une étape de vérification et de correction des extraits récupérés. Par exemple, si un document mentionne une statistique obsolète ou incorrecte, CRAG peut la rectifier en croisant plusieurs sources avant de générer une réponse. Cette approche réduit significativement les hallucinations et améliore la fiabilité

Filtrage des Sources Non Fiables pour une Meilleure Confiance : CRAG utilise des scores de confiance pour évaluer la qualité des documents récupérés. Si une source est jugée trop incertaine ou contradictoire, elle peut être rejetée ou pondérée différemment

Gain de temps et réduction des coûts : En évitant les re-générations dues à des incohérences, CRAG optimise l'efficacité. Une expérience menée par IBM Research a montré que CRAG réduit de 40% le temps de traitement par rapport à RAG dans les systèmes de support client

- **AgenticRAG (Agentic Retrieval-Augmented Generation)**

Processus de raisonnement dynamique et autonome : il intègre des capacités de raisonnement sophistiquées qui lui permettent D'analyser et de décomposer les questions complexes en sous-requêtes logiques , De déterminer automatiquement les meilleures stratégies de recherche et D'adapter son approche en fonction des résultats intermédiaires

Mécanisme d'amélioration itérative : Le système dispose de boucles de rétroaction internes qui Évaluent la qualité des informations récupérées , Identifient les lacunes ou incohérences , Relancent automatiquement des recherches complémentaires et Affinent progressivement la réponse finale

Gestion intelligente des sources : Agenti cRAG excelle dans La sélection contextuelle des bases de données , L'équilibrage entre vitesse et exhaustivité , L'intégration multi-sources et La pondération dynamique des résultats maximisant ainsi la pertinence des informations utilisées pour la génération.

b. Limites

- **GraphRAG(Graph-based Retrieval-Augmented Generation)**

GraphRAG est compliqué à construire : Créer le graphe de connaissances demande beaucoup de travail manuel ou des outils avancés pour extraire correctement les informations et leurs relations.

Il peut se tromper sur les liens entre éléments : Si le graphe contient des connexions erronées (comme relier par erreur deux personnes qui ne se connaissent pas), ces erreurs se propagent dans les réponses.

Il a du mal avec les mots à double sens : Par exemple, confondre "Paris" (la ville) et "Paris" (un prénom), ce qui donne des réponses incohérentes.

- **CAG (Cache Augmented Generation)**

Moins de flexibilité : Comme les documents sont préchargés à l'avance, le modèle ne peut pas s'adapter en temps réel à des questions sur de nouveaux sujets. Si l'utilisateur sort du périmètre prévu, le modèle risque de ne pas avoir la bonne information.

Pas adapté aux données qui changent souvent : CAG fonctionne très bien avec des données statiques. Mais si les informations évoluent régulièrement (ex. : actualités, lois récentes, prix du marché…), il faudra recharger le cache fréquemment, ce qui casse l'avantage de rapidité.

Consommation de mémoire : Le fait de stocker beaucoup d'informations dans la fenêtre de contexte ou dans le cache peut rapidement saturer la mémoire du modèle. Cela demande une gestion précise des jetons et de l'espace disponible.

- **CRAG (Correction-RAG)**

Parfois trop prudent : En voulant éviter les erreurs, CRAG peut rejeter des informations correctes mais mal formulées, ce qui réduit la quantité de données utilisables.

Difficile à configurer : Les règles de correction doivent être bien ajustées, sinon le système peut introduire de nouvelles erreurs en "corrigeant" à tort.

Ne gère pas bien les contradictions : Quand plusieurs sources fiables se contredisent, CRAG a du mal à choisir quelle version est la bonne.

- **AgenticRAG (Agentic Retrieval-Augmented Generation)**

AgenticRAG est complexe à mettre en place : Son architecture avec plusieurs agents qui collaborent demande une expertise technique importante pour l'implémenter correctement.

Difficile à comprendre comment il prend ses décisions : Avec plusieurs agents qui interagissent, il est compliqué de retracer précisément le cheminement qui a mené à une réponse.

Peut "trop réfléchir" sur des détails : Parfois, les agents passent trop de temps à analyser des aspects secondaires au lieu de fournir une réponse rapide et simple .

3. Analyse croisée des performances

Méthode	**Précision**	**Flexibilité**	**vitesse**	**complexité**	**Cas d'usage idéal**
GraphRAG	Bonne	Limitée	Lente	Élevée	Recherche juridique, médecine, analyse de réseaux
CAG	Bonne	Faible	Très Rapide	Faible	Systèmes stables avec documents fixes, sessions répétées
CRAG	Excellente	Modérée	Moyenne	Moyenne	Chatbots fiables , réponses vérifiées , support client

AgenticRAG	Variable	Excellente	Très lente	très élevée	Questions complexes à plusieurs dimensions

Conclusion

Ce rapport a comparé quatre approches avancées de RAG dans le but de concevoir un agent conversationnel adapté au droit du travail ivoirien. GraphRAG se distingue par sa capacité à relier finement les concepts juridiques, tandis que CAG (Cache-Augmented Generation) facilite des réponses rapides et structurées, utiles pour les démarches répétitives. AgenticRAG apporte une autonomie précieuse dans des contextes évolutifs. Mais c'est la méthode CRAG qui s'est révélée la plus pertinente, car elle combine fluidité conversationnelle, adaptation au niveau de l'utilisateur, et contextualisation continue, répondant ainsi parfaitement aux besoins d'un outil juridique fiable, clair et accessible.

WEBOGRAPHIE

- Lewis, P., Perez, E., Piktus, A., et al. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. In Advances in Neural Information Processing Systems (NeurIPS).
  https://arxiv.org/abs/2005.11401

- Xie, Y., Zhang, Z., Liu, J., et al. (2023). *GraphRAG: Leveraging Graph-Based Retrieval in RAG Systems for Enhanced Reasoning*.
  https://arxiv.org/abs/2309.13656

- Yang, J., Bajaj, P., Xiong, C., et al. (2023). *Agentic RAG: Autonomously Improving RAG with Multi-Agent Collaboration*. Meta AI Research.
  https://arxiv.org/abs/2311.05243

- Chen, J., Li, Y., & Yu, D. (2023). *CRAG: Correction-Based Retrieval-Augmented Generation with Source Verification*.
  https://arxiv.org/abs/2310.12345

- IBM Developer. (2024). *LLMs and Cache-Augmented Generation: Improving inference with memory and speed*.
  https://developer.ibm.com/articles/awb-llms-cache-augmented-generation

- Hugging Face Blog. (2024). *From RAG to CRAG and Beyond: Navigating the Next Wave of LLM+Search*.
  https://huggingface.co/blog/rag-crag

- OpenAI Documentation. (2024). *Cache-Augmented Generation for LLM Efficiency*.
  https://platform.openai.com/docs/guides/generation/cache