Aussi pour la technique RAG avec laquelle vous avez choisi de travailler; j'aurai aimé que vous ayez fait une étude comparative auparavant ou un benchmarking:

Les domaines dans lesquels ces LLM puissants présentent des lacunes en comparaison lorsqu'il s'agit d'être adaptés à des fins pratiques ;

Hallucination:
Ceci est un problème courant rencontré par les LLM (Langage Models Large) où le modèle génère des données incorrectes sur le plan factuel par rapport au contexte donné. Le problème est encore plus grave lorsqu'il s'agit du cas des chatbots médicaux, où l'on s'attend à ce qu'ils comprennent l'intention de la demande et répondent en fonction du contexte médical spécifique. Les défis peuvent être variés, tels que ne pas connaître le vocabulaire médical, des modèles de questions inhabituels, des connaissances de formation obsolètes, un manque de connaissances locales propres à l'organisation, etc. Fondamentalement, le modèle n'a pas la connaissance et la compréhension du contexte de ce qui lui est demandé et génère du texte de manière forcée basée sur son entraînement générique d'origine.
Modèle Boîte Noire
L'explicabilité d'un modèle GenAI est souvent limitée, car la plupart des modèles ne fournissent pas d'informations sur la manière dont une sortie spécifique a été générée ni sur les sources utilisées. Même avec des données d'entraînement et une architecture connues, la trace des vastes ensembles de données ne fournit pas d'indications utiles sur le pourquoi d'une sortie. Ceci représente un défi majeur en termes de confiance et d'adoption des modèles.
Coupure de Connaissance:
Le défi de la coupure de connaissance survient lorsque le modèle n'a aucune information postérieure à son entraînement ni au-delà de la portée de son ensemble de données d'entraînement. Essentiellement, le modèle reste figé dans le temps et la portée de son entraînement, ce qui pose un problème majeur pour les entreprises, surtout lorsque des informations privées sont cruciales pour des réponses précises.

Solutions:

- ☐ **Prompt Engineering:**

Le prompting guide le comportement de réponse des modèles linguistiques en affinant les entrées. L'ingénierie de la suggestion optimise les prompts pour poser les bonnes questions, permettant ainsi aux LLM de générer des réponses adaptées au contexte et à l'intention. C'est crucial pour les applications professionnelles en entreprise, mais seul, cela ne résout pas le manque de connaissances spécifiques nécessaires. C'est là que les deux méthodes suivantes entrent en jeu.

- ☐ **Fine Tuning**

Le finetuning consiste à entraîner davantage un modèle pré-entraîné sur un ensemble de données spécifique à la tâche, souvent plus petit que celui utilisé initialement. Il peut être complet, mettant à jour tous les paramètres, ou suivre des approches plus ciblées, comme le finetuning séquentiel ou le finetuning efficace en termes de paramètres.

Bien que le finetuning améliore la capacité du modèle à gérer divers scénarios, il est coûteux, chronophage et nécessite une expertise. Il présente toujours des limites, notamment une coupure de connaissance temporelle et une certaine propension à l'hallucination.

- ☐ **Retrieval Augmented Generation (RAG)**

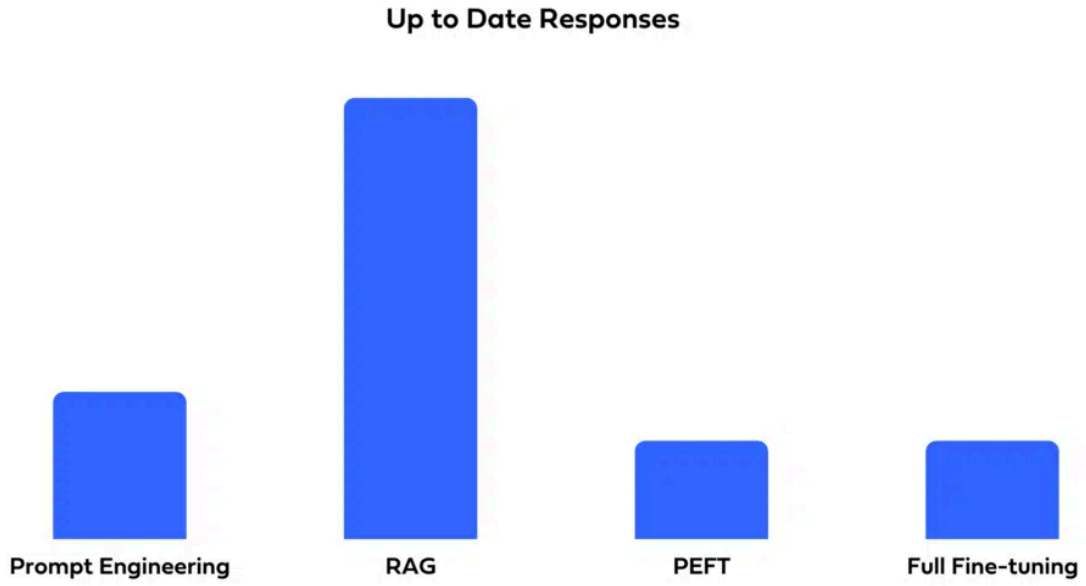
RAG combine la récupération contextuelle de sources de données pertinentes avec une stratégie de suggestion pour générer des réponses précises ancrées dans les faits. Cette technique permet au modèle de rechercher des informations externes, réduisant ainsi les hallucinations et offrant une pertinence temporelle, une transparence dans la provenance des informations, et une économie relative.

Bien que RAG soit puissant pour la génération de contenu basé sur des informations privées, il peut rencontrer des défis dans la compréhension des affaires internes et du contenu local.

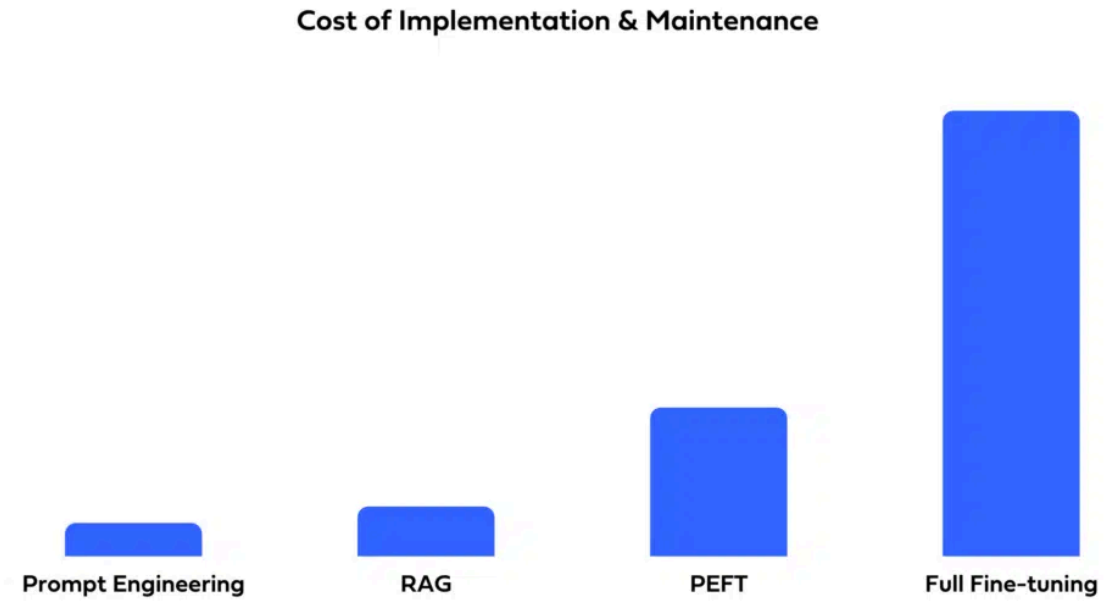
Conclusion:
Prompt Engineering : Optimise les suggestions pour aider le modèle à comprendre l'intention et le contexte de la demande.
Finetuning : Apporte la compréhension essentielle des affaires, des modèles internes et de l'intelligence contextuelle locale.
RAG : Ancre fermement la génération sur des informations factuelles et temporellement pertinentes, avec une transparence sur la source d'information.

Notre choix:
nous avon choisis d'implémenter RAG pour les causes suivantes:
La première raison pour lequel on a choisi rag et pas autre c’est:

Des réponses à jour:
Les requêtes externes de RAG peuvent garantir des réponses mises à jour, ce qui les rend idéales pour les environnements avec des données dynamiques.



Efficacité Coût/Utilisation Facile : RAG offre une solution relativement rentable en tirant parti des informations externes, réduisant ainsi la nécessité d'une finetuning intensif. De plus, son approche combinée de génération et de récupération simplifie l'utilisation du modèle, minimisant la complexité tout en optimisant la performance.

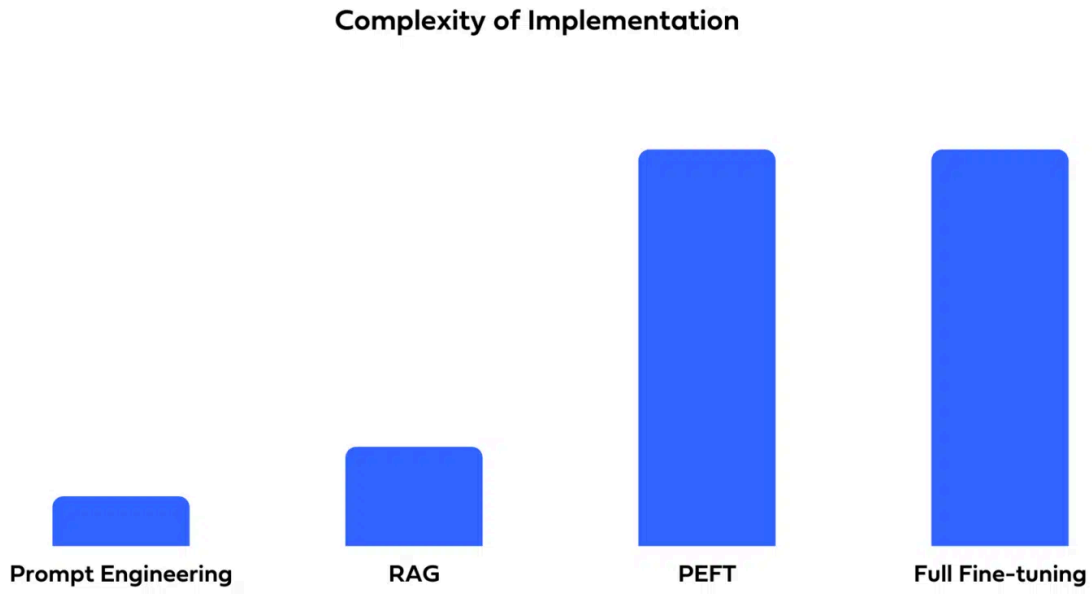


En comparaison avec prompt engineering et le finetuning, l'implémentation de RAG peut être plus accessible pour plusieurs raisons

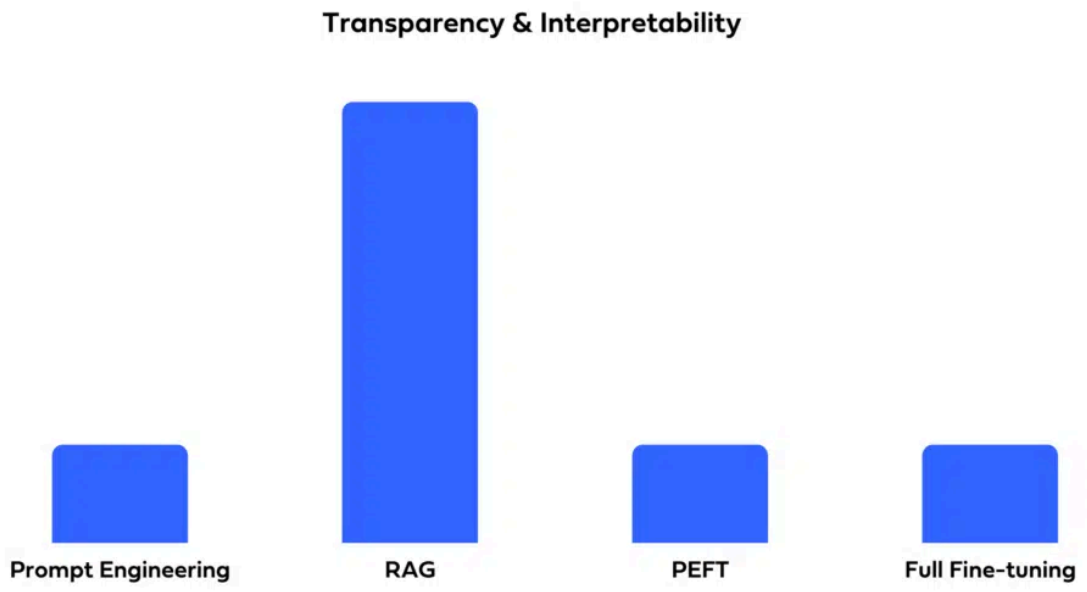
Moins de Besoin de Données Internes : RAG tire parti de sources d'information externes existantes, réduisant ainsi la nécessité de créer des ensembles de données spécifiques.

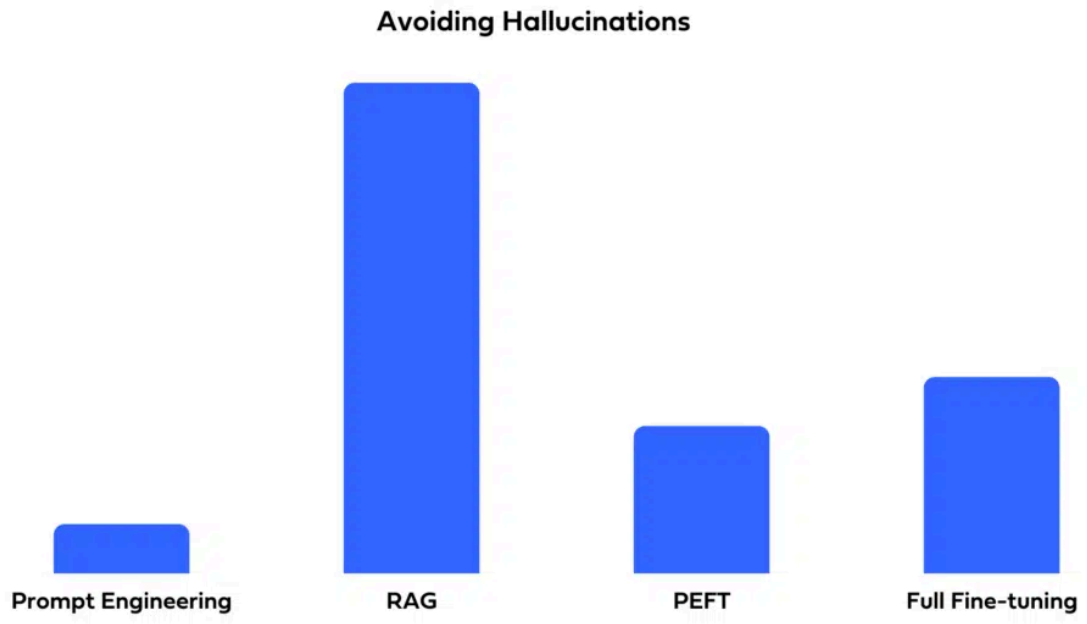
Moins de Réglages Fins : Contrairement au finetuning qui peut nécessiter des ajustements minutieux, RAG est plus tolérant aux variations de paramètres, simplifiant l'implémentation.

Simplicité dans la Gestion de la Logique : Intégrer la logique de génération avec la récupération externe peut être plus direct et moins complexe que d'optimiser des prompts ou de régler de nombreux paramètres, rendant l'implémentation de RAG plus accessible.



Transparence et interprétabilité:
Pour certaines applications, comprendre la prise de décision du modèle est crucial. Alors que le réglage fin fonctionne davantage comme une « boîte noire », obscurcissant son raisonnement, RAG fournit une vision plus claire. Son processus en deux étapes identifie les documents qu'il récupère, améliorant ainsi la confiance et la compréhension des utilisateurs.





RAG (Retrieval-Augmented Generation) réduit les hallucinations en ancrant la réponse du modèle dans les documents récupérés.
L'étape initiale de récupération sert essentiellement à vérifier les faits, tandis que la génération ultérieure est limitée au contexte des données récupérées.
Pour les tâches où éviter les hallucinations est primordial, il est recommandé d'utiliser RAG.