Rédaction des meilleures pratiques pour optimiser les applications RAG

AWS Conseils prescriptifs

AWS Conseils prescriptifs: Rédaction des meilleures pratiques pour optimiser les applications RAG

Table of Contents

Rédaction des meilleures pratiques pour optimiser les applications RAG

Ivan Cui et Samantha Stuart, Amazon Web Services

Juillet 2025 (historique du document)

Les grands modèles linguistiques (LLMs) ont révolutionné le domaine de l'intelligence artificielle grâce à leur remarquable capacité à comprendre et à générer du texte de type humain. Cependant, ils sont confrontés à une limite importante : ils ne peuvent travailler qu'avec les connaissances contenues dans leurs données d'entraînement. C'est là que la génération augmentée de récupération (RAG) est utile. Il propose une solution qui combine LLMs des sources de connaissances externes, telles que les données et les documents de votre organisation. Grâce à un processus en deux étapes impliquant la récupération d'informations et la génération de réponses, RAG permet aux systèmes d'IA d'accéder à des up-to-date informations provenant de diverses sources et de les intégrer, ce qui se traduit par des réponses plus précises et informées qui comblent le fossé entre les connaissances des modèles statiques et les besoins d'informations dynamiques du monde réel.

Comment optimiser le contenu à récupérer dans une application basée sur RAG ? Ce guide fournit les meilleures pratiques pour vous aider à optimiser le formatage et le style de rédaction du contenu textuel de la base de connaissances. L'optimisation du contenu améliore le contexte qui aide les applications RAG à comprendre les informations spécifiques aux tâches avec plus de précision. Lorsque le système récupère un contenu très pertinent et précis, la qualité de la réponse du LLM s'améliore. L'optimisation du processus de diffusion du contexte au niveau du système s'appelle l'ingénierie du contexte, et elle constitue un élément essentiel des architectures AGENTIC RAG. Dans un RAG agentic, une ou plusieurs LLMs raisons supplémentaires et agissez sur les demandes d'admission avant l'exécution du RAG. Cela facilite un processus de livraison d'informations en plusieurs étapes. Alors que les architectures RAG deviennent de plus en plus complexes, l'optimisation du contenu source reste le moyen le plus direct de fournir un contexte clair à LLMs. Ces meilleures pratiques sont conçues pour vous aider à optimiser l'investissement de votre organisation dans une application RAG.

Public visé

Ce guide est destiné aux ingénieurs en intelligence artificielle, aux scientifiques des données, aux ingénieurs de données ou aux développeurs de logiciels qui créent des applications LLM avec un

ou plusieurs composants RAG. Pour comprendre les concepts et les recommandations de ce guide, vous devez être familiarisé avec les bases de données vectorielles et les instructions pour LLMs.

Objectifs

Les recommandations de ce guide peuvent vous aider à atteindre les objectifs suivants :

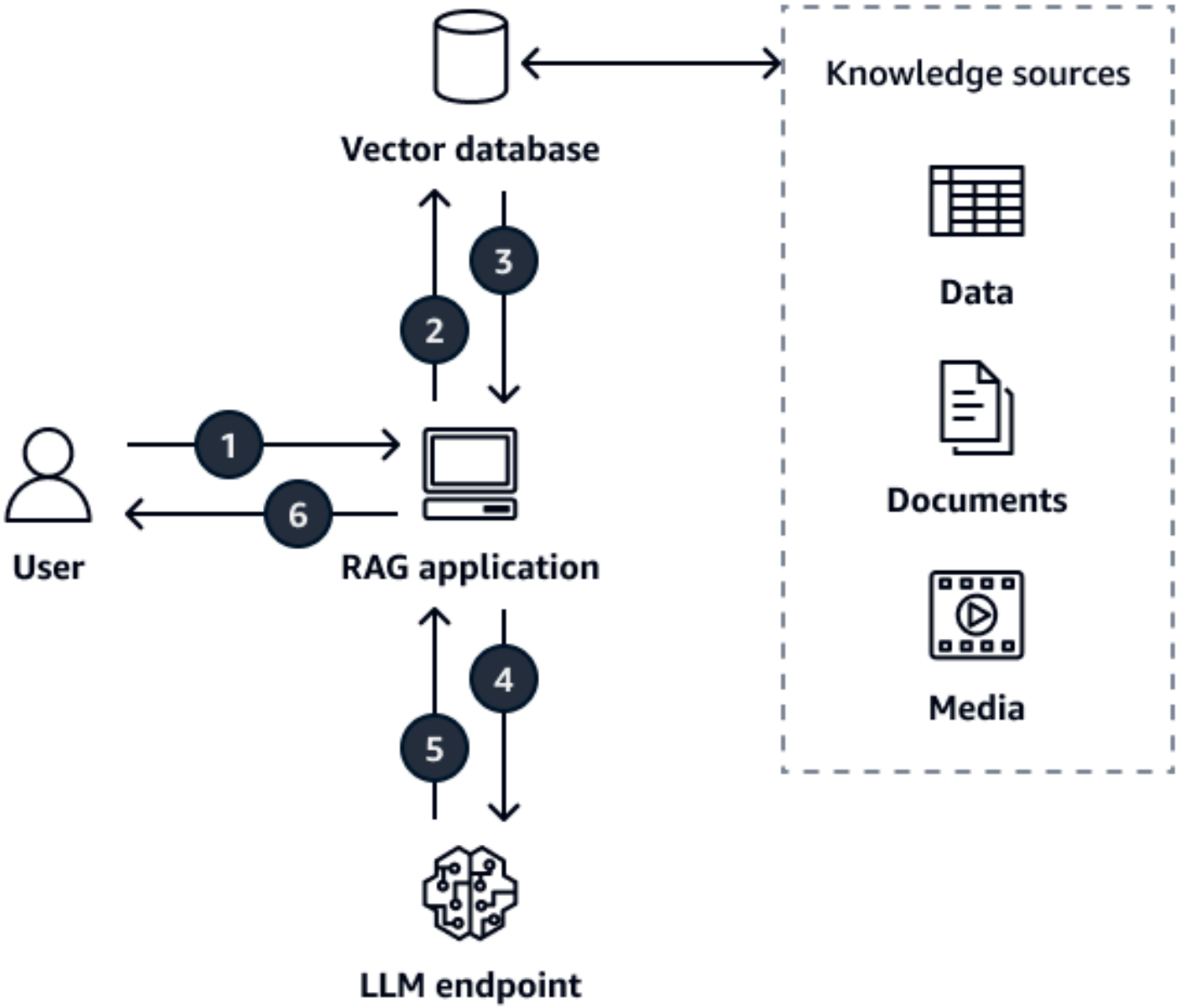
- Améliorez la précision et la pertinence des réponses générées par les applications RAG en fournissant des documents sources bien structurés et sémantiquement riches, optimisés pour l'utilisation des jetons et la redondance.
- Aidez les applications RAG à mieux comprendre les connaissances et le contexte spécifiques au domaine en fournissant des définitions et des explications claires dans les documents sources.
- Facilitez la maintenance et les mises à jour de la base de connaissances pour les applications RAG en respectant des directives de formatage et de structuration cohérentes dans les documents sources.
- Améliorez l'évolutivité des solutions RAG en décomposant les grands documents monolithiques en unités plus petites et autonomes qui peuvent être indexées et récupérées efficacement.

Comprendre le LLM et le RAG

Pour comprendre comment l'amélioration de la qualité du document source améliore la qualité d'une réponse RAG, vous devez comprendre le fonctionnement interne d'un LLM. Le véritable pouvoir des LLM réside dans leur capacité à utiliser des mécanismes d'attention personnelle et des architectures de transformateurs. Ces techniques avancées permettent aux modèles de traiter et de relier efficacement différentes parties de la séquence d'entrée, indépendamment de leur position ou de leur distance dans le texte. Cette fonctionnalité contraste nettement avec les modèles linguistiques traditionnels, qui ont souvent du mal à gérer les dépendances à long terme et à comprendre le contexte. De plus, les LLM sont formés à une échelle sans précédent. Certains des modèles les plus importants sont composés de milliards de paramètres et ont ingéré des téraoctets de données textuelles provenant de diverses sources. Cette échelle massive permet aux LLM de développer une riche compréhension du langage, en capturant des nuances subtiles, des expressions idiomatiques et des indices contextuels qui étaient auparavant difficiles pour les systèmes d'IA. Le résultat est une classe de modèles capables de générer un texte cohérent et fluide et de démontrer des capacités remarquables dans des tâches telles que la réponse aux questions, la synthèse de texte et même la génération de code.

Pour utiliser ces modèles, nous pouvons nous tourner vers des services tels qu'Amazon Bedrock, qui donne accès à une variété de modèles de base d'Amazon et de fournisseurs tiers, notamment Anthropic, Cohere et Meta. Vous pouvez utiliser Amazon Bedrock pour expérimenter des modèles de pointe, les personnaliser et les affiner, ou les intégrer à vos AI-powered solutions génératives via une API unique.

Bien que les LLM excellent dans la capture de modèles et la génération de texte cohérent, ils n'ont souvent pas accès à des informations actualisées ou spécialisées. RAG combine le pouvoir générateur des LLM avec un composant de récupération qui peut accéder aux informations pertinentes provenant de sources externes et les intégrer, dans le cadre de l'invite LLM matérialisée. Parmi les sources externes, on peut citer les bases de connaissances pour Amazon Bedrock, les systèmes de recherche intelligents tels qu'Amazon Kendra ou les bases de données vectorielles telles qu'Amazon OpenSearch Service.



Le diagramme décrit le flux de travail suivant :

1. L'utilisateur soumet une requête à l'application RAG.
2. L'application RAG interroge une base de données vectorielle contenant des sources de connaissances, telles que des documents, des données ou des médias.
3. L'application RAG extrait les informations pertinentes de la base de données vectorielle en fonction des similitudes sémantiques entre la requête et les documents stockés.
4. L'application RAG complète l'invite d'origine avec le contexte récupéré et l'envoie au point de terminaison LLM.

5. Le point de terminaison LLM génère une réponse et la renvoie à l'application RAG.
6. L'application RAG renvoie la réponse générée à l'utilisateur.

À la base, RAG utilise un processus en deux étapes. Dans un premier temps, un modèle de récupération identifie et extrait les documents ou passages pertinents en fonction de la requête d'entrée. Ce modèle de récupération peut être un système de récupération d'informations traditionnel, un modèle de récupération dense ou une combinaison des deux. Dans la deuxième étape, les informations récupérées et la requête d'origine sont introduites dans un LLM sous forme de modèle d'invite entièrement matérialisé. Les LLM dépendent fortement de la qualité du contenu source fourni par le composant retriever. Ils appliquent un mécanisme d'attention personnelle pour encoder mathématiquement le lien entre le contenu récupéré et la tâche. Le LLM génère ensuite une réponse basée à la fois sur la requête et sur les informations extraites. Dans RAG, le contrôle de la qualité des documents sources récupérés représente un moyen direct d'améliorer la représentation interne d'une tâche par un LLM. RAG augmente efficacement les données de formation du LLM avec des données externes pertinentes. Cette approche permet à RAG de tirer parti des points forts des LLM et des systèmes de récupération, permettant ainsi de générer des réponses plus précises et informées intégrant les connaissances actuelles et spécialisées.

Vecteurs et intégrations

Les vecteurs et les intégrations sont des concepts fondamentaux de l'apprentissage automatique et du traitement du langage naturel. Les vecteurs sont des objets mathématiques qui représentent des quantités possédant à la fois une amplitude et une direction. Dans le contexte du traitement du langage naturel (NLP), les mots, les phrases ou les documents sont souvent représentés sous forme de vecteurs dans des espaces vectoriels de grande dimension. Les intégrations, quant à elles, sont un moyen de représenter des objets tels que des mots ou des documents dans un espace vectoriel de dimension inférieure où les relations entre les vecteurs capturent des similitudes sémantiques ou syntaxiques. Les intégrations de mots, par exemple, permettent à des mots ayant des significations similaires d'avoir des représentations vectorielles similaires. Cela permet aux algorithmes de comprendre et de traiter le langage plus efficacement.

bases de données vectorielles

Dans l'IA générative, une base de données vectorielle est une base de données qui stocke et gère les représentations vectorielles de documents, de requêtes ou d'autres objets. Il est conçu pour stocker et récupérer efficacement les vecteurs. Cela prend en charge des opérations rapides et

évolutives telles que la recherche sémantique et la mise en correspondance des similitudes. Les bases de données vectorielles indexent les vecteurs à l'aide de structures de données spécialisées, telles que les graphes Hierarchical Navigable Small World (HNSW) ou les algorithmes K-Nearest Neighbors (KNN). Ces structures de données permettent d'effectuer des recherches rapides auprès des voisins les plus proches, ce qui permet de trouver rapidement des vecteurs similaires dans la base de données.

Recherche sémantique

La recherche sémantique est une technique qui améliore la pertinence des résultats de recherche en comprenant l'intention et le contexte de la requête, plutôt que de simplement faire correspondre des mots clés. En termes techniques, la recherche sémantique consiste à comparer les représentations vectorielles de la requête et les documents de la base de données afin de trouver les correspondances les plus pertinentes. Différentes stratégies de récupération peuvent être utilisées pour la recherche sémantique, notamment :

- HNSW — Structure de données basée sur des graphes qui organise les vecteurs de manière à faciliter la recherche des voisins les plus proches.
- KNN — Algorithme qui trouve les K vecteurs les plus proches d'un vecteur de requête sur la base d'une métrique de distance, telle que la similitude des cosinus.
- Similarité du cosinus : mesure de similitude entre deux vecteurs non nuls qui mesure le cosinus de l'angle entre eux. Il est souvent utilisé dans la recherche sémantique pour comparer la direction des vecteurs dans un espace de grande dimension.
- Locality-sensitive hachage (LSH) : technique qui hache des vecteurs similaires vers des compartiments identiques ou voisins avec une probabilité élevée. Cela permet d'effectuer des recherches approximatives auprès du plus proche voisin, ce qui peut être plus rapide que des recherches exactes dans des espaces de grande dimension.

Défis liés aux données sources qui affectent les applications RAG

L'un des principaux défis du développement d'une application de génération augmentée par extraction (RAG) optimale réside dans la nature des données brutes ou des documents utilisés. Les entreprises utilisent souvent des documents existants créés à des fins de référence humaine. Ces documents incluent souvent des hyperliens et des captures d'écran pour favoriser la compréhension. Cependant, ces éléments entravent la récupération sémantique en raison des limites de jetons d'extrait. Cela se traduit par de mauvaises performances du retriever.

Voici les défis les plus courants liés aux documents bruts pour une application RAG optimale :

- Absence de formatage structuré et de métadonnées — Les documents bruts peuvent ne pas avoir de titres de section, de sous-titres ou de métadonnées clairs. Il est donc difficile d'identifier et d'extraire les informations pertinentes. Par exemple, un long document sans titres clairs peut compliquer la détermination du contexte de certaines informations.
- Langage informel et incohérent — Les documents bruts contiennent souvent un langage informel ou une terminologie incohérente. Cela peut semer la confusion dans les modèles RAG. Par exemple, des abréviations qui ne sont pas définies dans le document ou qui sont déjà connues par le LLM peuvent être utilisées dans l'ensemble d'un document.
- Verbosité et redondance — Les documents bruts peuvent être détaillés et contenir des informations inutiles ou redondantes. Cela peut submerger les modèles RAG, ce qui entraîne des réponses moins concises et moins pertinentes. Les exemples incluent un document qui répète les mêmes informations plusieurs fois ou plusieurs documents contenant des informations similaires ou contradictoires.
- Termes et expressions ambigus — Les documents bruts peuvent contenir des termes ou des phrases ambigus susceptibles d'être interprétés de différentes manières. Cette ambiguïté peut entraîner des interprétations erronées par les modèles RAG et des réponses inexactes. Par exemple, un document qui utilise un terme ayant plusieurs significations peut donner lieu à une réponse qui ne correspond pas au sens voulu.
- Injection d'éléments graphiques et d'hyperliens — Les documents bruts contenant des graphiques et des informations liées à des hyperliens conviennent parfaitement à la consommation humaine. Cependant, ces éléments peuvent consommer la limite de jetons de récupération. Il en résulte que les extraits peuvent être incomplets. Par exemple, les graphiques et les hyperliens URLs

sont renvoyés dans le cadre de la récupération, ce qui utilise les jetons de récupération, et les informations clés des paragraphes suivants sont manquantes.

- Manque de connaissances ou de contexte spécifiques à un domaine — Les documents bruts peuvent ne pas disposer des connaissances spécifiques au domaine ou du contexte nécessaires pour une génération précise. Cela peut limiter la capacité des modèles RAG à générer des réponses pertinentes et précises. Un exemple est un document qui fait référence à des concepts spécialisés sans fournir de contexte. Cela peut conduire à des réponses qui ne sont pas pertinentes dans le domaine donné.

Bien que cette liste ne soit pas exhaustive, elle fournit aux entreprises un point de départ pour réfléchir à ce qui ne fonctionne pas et pourquoi. Les documents peuvent présenter un ou plusieurs de ces défis. La clé de l'optimisation d'une application RAG est d'utiliser un ensemble de documents conformes aux meilleures pratiques de rédaction qui optimisent la récupération.

Bonnes pratiques en matière de documentation pour les applications RAG

Le développement d'une application RAG (Retrieval-Augmented Generation) réussie nécessite une prise en compte attentive de divers facteurs liés aux documents afin d'optimiser ses performances. Les meilleures pratiques présentées dans cette section sont sélectionnées sur la base de l'expérience de création de systèmes RAG avec de nombreux dirigeants d'organisations. Voici quelques bonnes pratiques clés en matière de documents afin d'améliorer l'efficacité de votre application RAG :

- Utilisez correctement les titres et les sous-titres — L'organisation de votre contenu à l'aide de titres et de sous-titres clairs améliore la lisibilité et aide les modèles RAG à comprendre la structure de vos documents. Cette pratique permet aux modèles de mieux naviguer et d'extraire les informations des documents, ce qui améliore la qualité des réponses générées.
- Assurez-vous que la numérotation est séquentielle — Lorsque vous utilisez des listes numérotées, il est important de maintenir une numérotation correcte pour éviter toute confusion. Assurez-vous que chaque élément de la liste est numéroté de manière séquentielle sans sauter de chiffres. Cela permet de maintenir la clarté et la cohérence de votre contenu.
- Ajouter des transitions entre les éléments d'une liste — La fourniture de transitions entre les éléments d'une liste à puces ou numérotée aide le LLM à parcourir le contenu. Par exemple, vous pouvez utiliser des phrases telles que « Après avoir terminé l'étape 2, faites... » pour relier les idées et améliorer le flux d'informations.
- Remplacer les tables : évitez d'utiliser des tables. Formatez ces informations sous forme de listes à puces à plusieurs niveaux ou dans une syntaxe à niveau plat. La syntaxe à niveau plat consiste à organiser des éléments ou des éléments au même niveau hiérarchique, sans niveaux de subordination imbriqués. Ces structures aident LLMs à digérer les informations. Comme la plupart des documents indexés sont lus de gauche à droite, la syntaxe à niveau plat permet aux informations de suivre les informations de manière plus cohérente sans qu'il soit nécessaire de faire référence à une dimension supplémentaire. Ce format est plus propice aux applications RAG car il présente les informations de manière structurée et facile à digérer.
- Prétraitez les informations graphiques pour plus d'efficacité — Le mode multimodal LLMs peut ingérer à la fois des images et du texte. Réduisez la résolution des images, supprimez les images redondantes et décrivez le contenu des éléments graphiques au format texte. Ces mesures

améliorent le contexte significatif, évitent de consommer des jetons inutilement et améliorent l'accessibilité des modèles RAG.

- Ajoutez des démarreurs de session pour les requêtes courantes : lorsque vous répondez à des questions ou à des tâches courantes, telles que « Comment commander un logiciel ? » , ajoutez un démarreur de session qui permet au lecteur d'accéder au processus. Par exemple, vous pouvez ajouter « Si vous souhaitez commander un logiciel, suivez les étapes ci-dessous... ». Cela permet de créer une correspondance sémantique élevée, ce qui aide le LLM à construire une réponse cohérente.
- Ajouter un résumé à chaque section — Après chaque titre ou sous-titre, ajoutez un résumé bref et concis du contenu de cette section. Cela peut augmenter la couverture sémantique et renforcer les points clés. Cela améliore la précision de la recherche de similarité dans l'espace d'intégration, améliorant ainsi les performances de l'application RAG. Cela est particulièrement utile si le document est destiné à la fois au LLM et à la consommation humaine ou si des éléments tabulaires et graphiques sont nécessaires.
- Homonymie — Les documents doivent être concis et ciblés. LLMs générer des réponses basées sur des extraits extraits, afin que la désambiguïsation aide le modèle à utiliser des informations claires et pertinentes. Cela se traduit par des réponses plus précises et informatives.
- Définissez des abréviations et définissez le contexte : ils LLMs sont formés sur de grandes quantités de données Internet et, la plupart du temps, ils n'ont pas le contexte des documents internes d'une entreprise. Par conséquent, le fait de définir le contexte, de définir des abréviations et d'éviter ou de définir une terminologie spécifique à l'entreprise aide le LLM à comprendre les données de votre entreprise. Cela aide le LLM à répondre aux questions avec plus de précision et peut aider à prévenir les hallucinations.
- Restructurez les documents volumineux en documents plus petits pour un balisage et une indexation efficaces : évitez d'indexer un document volumineux contenant plusieurs sous-rubriques. Envisagez de diviser le document volumineux en documents plus petits et autonomes dotés de titres clairs. Cela améliore l'indexation et le balisage.

FAQ

Pourquoi est-il important d'optimiser les documents pour les applications RAG ?

Les documents bruts sont souvent rédigés pour la consommation humaine sans tenir compte des exigences des systèmes d'intelligence artificielle avancés, tels que les applications RAG (Retrieval Augmented Generation). L'optimisation des documents en suivant les meilleures pratiques peut améliorer de manière significative les performances et la précision des applications RAG en fournissant des informations structurées, claires et pertinentes aux modèles.

Quels sont les problèmes courants liés aux documents bruts qui peuvent nuire aux performances de RAG ?

Parmi les principaux défis, citons le manque de mise en forme et de métadonnées structurées, le langage informel ou incohérent, la verbosité et la redondance, les termes et expressions ambigus, l'inclusion d'éléments d'hyperliens et l'absence de contexte spécifique au domaine. Ces problèmes peuvent semer la confusion dans les modèles RAG et entraîner des réponses inexactes ou non pertinentes. Pour plus d'informations, consultez la section Défis liés aux données source qui affectent les applications RAG dans ce guide.

Comment l'utilisation de titres et de sous-titres peut-elle améliorer les performances du RAG ?

Des titres et sous-titres clairs aident les modèles RAG à comprendre la structure et le contexte du contenu. Cela leur permet de mieux naviguer et d'extraire les informations pertinentes des documents et d'améliorer la qualité des réponses générées. Pour plus d'informations, consultez la section Bonnes pratiques relatives à la documentation pour les applications RAG dans ce guide.

Pourquoi est-il recommandé de remplacer les informations du tableau par une syntaxe de niveau plat ?

Il peut être difficile pour les modèles RAG d'interpréter les tableaux car ils nécessitent une compréhension de la structure bidimensionnelle. La présentation des informations des tableaux

sous forme de syntaxe plate ou de liste à puces permet aux modèles de traiter plus facilement les informations, ce qui améliore les performances. Pour plus d'informations, consultez la section Bonnes pratiques relatives à la documentation pour les applications RAG dans ce guide.

Comment l'ajout de résumés peut-il améliorer les performances du RAG ?

L'inclusion de résumés concis au début de chaque section ou sous-section peut accroître la couverture sémantique et renforcer les points clés. Cela améliore la précision des recherches de similarité dans l'espace d'intégration, ce qui améliore en fin de compte les performances de l'application RAG. Pour plus d'informations, consultez la section Bonnes pratiques relatives à la documentation pour les applications RAG dans ce guide.

Pourquoi est-il important de définir des abréviations et d'en définir le contexte ? LLMs

LLMs sont formés sur un large éventail de données, mais ils manquent de contexte pour les abréviations ou la terminologie propres à l'entreprise. La définition d'abréviations et la mise en contexte aident à LLMs comprendre et à répondre avec plus de précision. Cela peut aider à prévenir les hallucinations ou les interprétations erronées. Pour plus d'informations, consultez la section Bonnes pratiques relatives à la documentation pour les applications RAG dans ce guide.

Prochaines étapes et ressources

Ce guide décrit un ensemble de défis au niveau des documents pour les applications RAG et les meilleures pratiques pour les atténuer. Ces enseignements sont issus d'entretiens et de discussions avec des leaders du secteur, et ils sont étayés par des cas d'utilisation en entreprise.

Pour commencer à optimiser vos documents pour les applications RAG, nous vous recommandons de réaliser un audit de vos documents existants. Identifiez les domaines qui présentent des défis pour l'application RAG. Les exemples incluent un manque de structure, un langage ambigu ou une utilisation excessive d'éléments graphiques. Priorisez les documents fréquemment consultés ou essentiels à vos activités commerciales. Collaborez avec des experts en la matière pour mettre en œuvre les meilleures pratiques décrites dans ce guide. Assurez-vous que les documents sont restructurés avec des titres clairs, un langage concis et des éléments contextuels. Pour les nouveaux documents, établissez des directives et des modèles qui garantissent la cohérence et aident les auteurs à respecter les meilleures pratiques. En outre, envisagez d'investir dans des outils ou des services capables d'automatiser certains aspects du processus d'optimisation des documents, tels que l'utilisation de l'IA générative pour restructurer les documents. En adoptant une approche proactive de l'optimisation des documents, vous pouvez exploiter tout le potentiel des applications RAG et obtenir des résultats plus précis et plus pertinents au sein de votre organisation.

Les ressources suivantes peuvent vous aider à comprendre et à créer des applications RAG dans votre organisation.

Ressources

AWS documentation

- Choix d'une base de données AWS vectorielle pour les cas d'utilisation de RAG (directives AWS prescriptives)
- Déployez un cas d'utilisation de RAG AWS en utilisant Terraform et Amazon Bedrock (directives prescriptives)AWS
- Développez des assistants avancés basés sur l'IA générative basés sur le chat en utilisant RAG et des ReAct instructions (directives prescriptives)AWS
- Récupérez des données et générez des réponses basées sur l'IA avec les bases de connaissances Amazon Bedrock (documentation Amazon Bedrock)

- Génération augmentée de récupération (documentation Amazon SageMaker AI)
- Récupérez les options et architectures de génération augmentée sur AWS(directivesAWS prescriptives)

Autres AWS ressources

- Créez un assistant multimodal avec RAG avancé et Amazon Bedrock (AWS article de blog)

Historique du document

Le tableau suivant décrit les modifications importantes apportées à ce guide. Pour être averti des mises à jour à venir, abonnez-vous à un fil RSS.

Modification	Description	Date
Publication initiale	—	24 juillet 2025

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.