Scrittura di best practice per ottimizzare le applicazioni RAG

AWS Linee guida prescrittive

AWS Linee guida prescrittive: Scrittura di best practice per ottimizzare le applicazioni RAG

Table of Contents

Scrittura di best practice per ottimizzare le applicazioni RAG

Ivan Cui e Samantha Stuart, Amazon Web Services

Luglio 2025 (storia del documento)

I modelli linguistici di grandi dimensioni (LLMs) hanno rivoluzionato il campo dell'intelligenza artificiale grazie alla loro straordinaria capacità di comprendere e generare testi simili a quelli umani. Tuttavia, devono affrontare un limite significativo: possono lavorare solo con le conoscenze contenute nei dati di formazione. È qui che aiuta Retrieval Augmented Generation (RAG). Offre una soluzione che si combina LLMs con fonti di conoscenza esterne, come i dati e i documenti dell'organizzazione. Attraverso un processo in due fasi che prevede il recupero delle informazioni e la generazione di risposte, RAG consente ai sistemi di intelligenza artificiale di accedere e incorporare up-to-date informazioni da varie fonti, ottenendo risposte più accurate e informate che colmano il divario tra la conoscenza dei modelli statici e le esigenze dinamiche di informazione del mondo reale.

Come è possibile ottimizzare i contenuti per il recupero in un'applicazione basata su RAG? Questa guida fornisce le migliori pratiche per aiutarvi a ottimizzare la formattazione e lo stile di scrittura dei contenuti basati su testo nella knowledge base. L'ottimizzazione del contenuto migliora il contesto che aiuta le applicazioni RAG a comprendere con maggiore precisione le informazioni specifiche delle attività. Quando il sistema recupera contenuti altamente pertinenti e accurati, la qualità della risposta del LLM migliora. L'ottimizzazione del processo di distribuzione del contesto a livello di sistema si chiama ingegneria del contesto e costituisce una parte essenziale delle architetture AGENTIC RAG. In agentic RAG, una o più LLMs motivazioni aggiuntive e agisci in base alle richieste di acquisizione prima dell'esecuzione del RAG. Ciò facilita un processo di distribuzione delle informazioni in più fasi. Poiché le architetture RAG diventano sempre più complesse, l'ottimizzazione dei contenuti sorgente rimane il mezzo più diretto per fornire un contesto chiaro. LLMs Queste best practice sono progettate per aiutarvi a massimizzare l'investimento della vostra organizzazione in un'applicazione RAG.

Destinatari principali

Questa guida è destinata agli ingegneri di intelligenza artificiale, ai data scientist, ai data engineer o agli sviluppatori di software che stanno creando applicazioni LLM con uno o più componenti RAG. Per comprendere i concetti e i consigli contenuti in questa guida, è necessario conoscere i database vettoriali e i prompt for. LLMs

Obiettivi

I consigli contenuti in questa guida possono aiutarti a raggiungere i seguenti obiettivi:

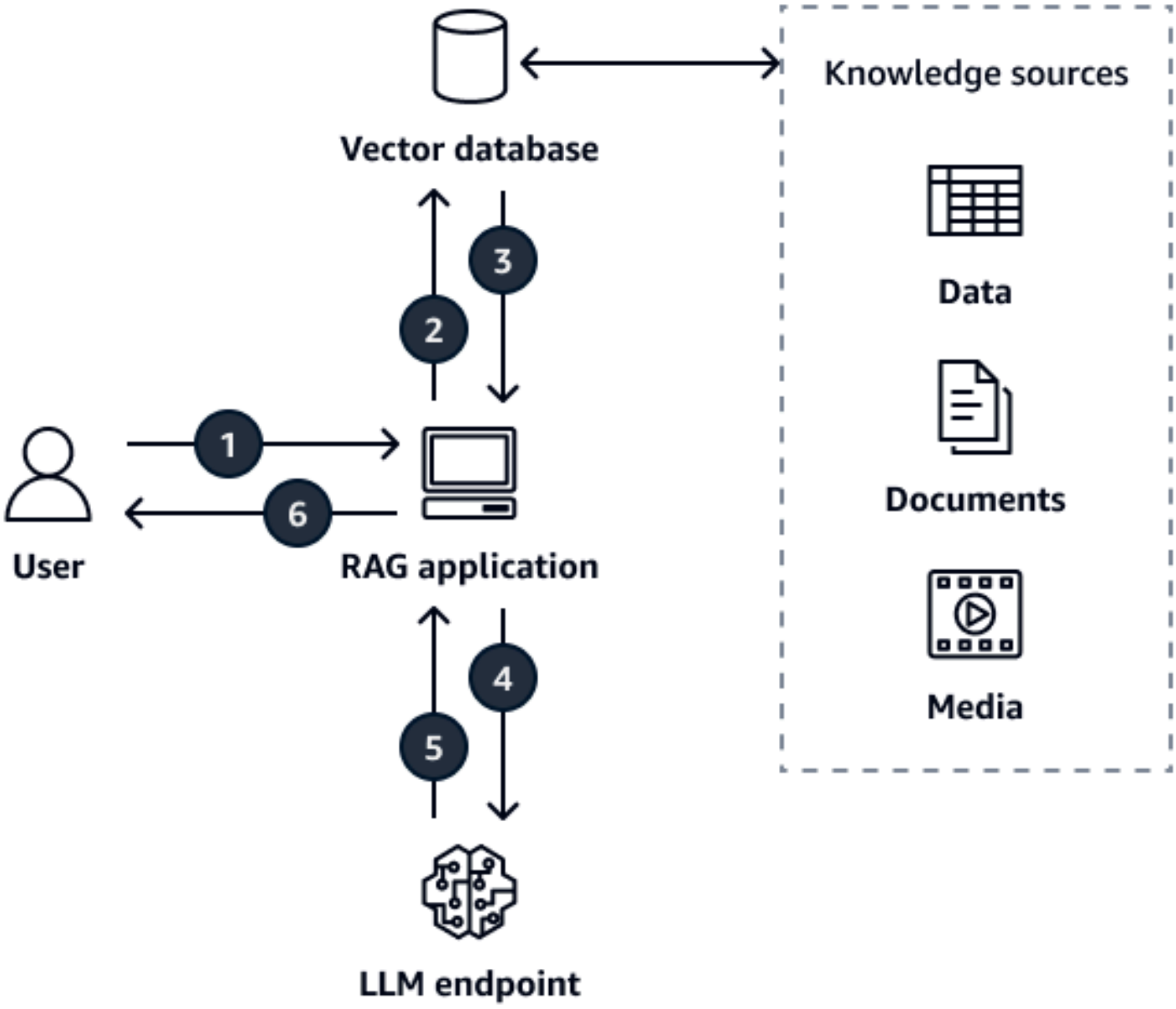
- Migliora l'accuratezza e la pertinenza delle risposte generate dalle applicazioni RAG fornendo documenti sorgente ben strutturati e semanticamente ricchi, ottimizzati per l'utilizzo e la ridondanza dei token.
- Aiuta le applicazioni RAG a comprendere meglio le conoscenze e il contesto specifici del dominio fornendo definizioni e spiegazioni chiare all'interno dei documenti di origine.
- Semplifica la manutenzione e gli aggiornamenti della knowledge base per le applicazioni RAG aderendo a linee guida di formattazione e strutturazione coerenti per tutti i documenti di origine.
- Migliora la scalabilità delle soluzioni RAG suddividendo documenti monolitici di grandi dimensioni in unità più piccole e autonome che possono essere indicizzate e recuperate in modo efficiente.

Comprendere LLM e RAG

Per capire come il miglioramento della qualità dei documenti di origine migliori la qualità di una risposta RAG, è necessario comprendere il funzionamento interno di un LLM. Il vero potere dei LLM risiede nella loro capacità di utilizzare meccanismi di autoattenzione e architetture di trasformazione. Queste tecniche avanzate consentono ai modelli di elaborare e mettere in relazione in modo efficace diverse parti della sequenza di input, indipendentemente dalla loro posizione o distanza all'interno del testo. Questa funzionalità è in netto contrasto con i modelli linguistici tradizionali, che spesso hanno difficoltà a gestire dipendenze a lungo termine e a comprendere il contesto. Inoltre, i LLM vengono formati su una scala senza precedenti. Alcuni dei modelli più grandi comprendono trilioni di parametri e hanno assorbito terabyte di dati testuali da diverse fonti. Questa vasta scala consente agli LLM di sviluppare una comprensione approfondita del linguaggio, cogliendo sfumature sottili, idiomi e segnali contestuali che in precedenza erano difficili per i sistemi di intelligenza artificiale. Il risultato è una classe di modelli in grado di generare un testo coerente e scorrevole e dimostrare capacità straordinarie in attività come la risposta alle domande, il riepilogo del testo e persino la generazione di codice.

Per utilizzare questi modelli, possiamo rivolgerci a servizi come Amazon Bedrock, che fornisce l'accesso a una varietà di modelli di base di Amazon e di fornitori terzi, tra cui Anthropic, Cohere e Meta. Puoi usare Amazon Bedrock per sperimentare modelli all'avanguardia, personalizzarli e perfezionarli o incorporarli nelle tue soluzioni generative AI-powered tramite un'unica API.

Sebbene i LLM siano eccellenti nell'acquisizione di modelli e nella generazione di testo coerente, spesso non hanno accesso a informazioni aggiornate o specializzate. RAG combina la potenza generativa dei LLM con un componente di recupero in grado di accedere e incorporare le informazioni pertinenti da fonti esterne, come parte del prompt LLM materializzato. Esempi di fonti esterne includono Knowledge Base per Amazon Bedrock, sistemi di ricerca intelligenti come Amazon Kendra o database vettoriali come Amazon Service. OpenSearch



Il diagramma descrive il seguente flusso di lavoro:

1. L'utente invia una query all'applicazione RAG.
2. L'applicazione RAG interroga un database vettoriale che contiene fonti di conoscenza, come documenti, dati o contenuti multimediali.
3. L'applicazione RAG recupera le informazioni pertinenti dal database vettoriale in base alle somiglianze semantiche tra la query e i documenti archiviati.
4. L'applicazione RAG amplia il prompt originale con il contesto recuperato e lo invia all'endpoint LLM.

5. L'endpoint LLM genera una risposta e la restituisce all'applicazione RAG.
6. L'applicazione RAG restituisce la risposta generata all'utente.

Fondamentalmente, RAG utilizza un processo in due fasi. Nella prima fase, un modello di recupero identifica e recupera i documenti o i passaggi pertinenti in base alla query di input. Questo modello di recupero può essere un sistema di recupero delle informazioni tradizionale, un modello di recupero ad alta densità o una combinazione di entrambi. Nella seconda fase, le informazioni recuperate e la query originale vengono inserite in un LLM come modello di prompt completamente materializzato. I LLM dipendono in larga misura dalla qualità del contenuto sorgente fornito dal componente retriever. Applicano un meccanismo di autoattenzione per codificare matematicamente il modo in cui il contenuto recuperato si riferisce all'attività. L'LLM genera quindi una risposta basata sia sulla domanda che sulle informazioni recuperate. In RAG, il controllo della qualità dei documenti sorgente recuperati rappresenta un mezzo diretto per migliorare la rappresentazione interna di un compito da parte di un LLM. RAG amplia efficacemente i dati di formazione del LLM con dati esterni pertinenti. Questo approccio consente a RAG di sfruttare i punti di forza dei LLM e dei sistemi di recupero, consentendo la generazione di risposte più accurate e informate che incorporano conoscenze attuali e specializzate.

Vettori e incorporamenti

I vettori e gli incorporamenti sono concetti fondamentali nell'apprendimento automatico e nell'elaborazione del linguaggio naturale. I vettori sono oggetti matematici che rappresentano quantità che hanno sia grandezza che direzione. Nel contesto dell'elaborazione del linguaggio naturale (NLP), parole, frasi o documenti sono spesso rappresentati come vettori in spazi vettoriali ad alta dimensione. Gli incorporamenti, d'altra parte, sono un modo per rappresentare oggetti come parole o documenti in uno spazio vettoriale di dimensioni inferiori in cui le relazioni tra i vettori catturano somiglianze semantiche o sintattiche. Gli incorporamenti di parole, ad esempio, consentono a parole con significati simili di avere rappresentazioni vettoriali simili. Questo aiuta gli algoritmi a comprendere ed elaborare il linguaggio in modo più efficace.

Database vettoriali

Nell'intelligenza artificiale generativa, un database vettoriale è un database che archivia e gestisce rappresentazioni vettoriali di documenti, query o altri oggetti. È progettato per archiviare e recuperare i vettori in modo efficiente. Ciò supporta operazioni veloci e scalabili come la ricerca semantica e la corrispondenza per somiglianza. I database vettoriali indicizzano i vettori utilizzando strutture

di dati specializzate, come i grafici Hierarchical Navigable Small World (HNSW) o gli algoritmi Neighbors (KNN). K-Nearest Queste strutture di dati consentono ricerche rapide tra i vicini più vicini, permettendo di trovare rapidamente vettori simili nel database.

Ricerca semantica

La ricerca semantica è una tecnica che migliora la pertinenza dei risultati di ricerca comprendendo l'intento e il contesto della query, anziché limitarsi a corrispondere le parole chiave. In termini tecnici, la ricerca semantica implica il confronto delle rappresentazioni vettoriali della query e dei documenti nel database per trovare le corrispondenze più pertinenti. È possibile utilizzare diverse strategie di recupero per la ricerca semantica, tra cui, a titolo esemplificativo ma non esaustivo:

- HNSW: una struttura di dati basata su grafici che organizza i vettori in modo da rendere efficiente la ricerca dei vicini più vicini.
- KNN — Un algoritmo che trova i K vettori più vicini a un vettore di query in base a una metrica di distanza, come la somiglianza del coseno.
- Somiglianza del coseno: una misura di somiglianza tra due vettori diversi da zero che misura il coseno dell'angolo tra di loro. Viene spesso utilizzato nella ricerca semantica per confrontare la direzione dei vettori in uno spazio ad alta dimensione.
- Locality-sensitive hashing (LSH): una tecnica che esegue l'hashing di vettori simili sullo stesso bucket o su quelli vicini con un'alta probabilità. Ciò consente ricerche approssimative tra i vicini più vicini, che possono essere più veloci delle ricerche esatte in spazi ad alta dimensione.

Sfide relative ai dati di origine che influiscono sulle applicazioni RAG

Una delle sfide principali nello sviluppo di un'applicazione Retrieval-Augmented Generation (RAG) ottimale risiede nella natura dei dati o dei documenti grezzi utilizzati. Spesso, le aziende utilizzano documenti esistenti creati come riferimento umano. Questi documenti spesso includono collegamenti ipertestuali e schermate di immagini per promuovere la comprensione. Tuttavia, questi elementi ostacolano il recupero semantico a causa dei limiti dei token di estratto. Ciò si traduce in prestazioni scadenti del retriever.

Di seguito sono elencate le sfide più comuni relative ai documenti non elaborati per un'applicazione RAG ottimale:

- Mancanza di formattazione e metadati strutturati: nei documenti raw possono mancare titoli di sezione, sottotitoli o metadati chiari. Ciò rende difficile identificare ed estrarre le informazioni pertinenti. Ad esempio, un documento lungo senza titoli chiari può rendere difficile determinare il contesto di informazioni specifiche.
- Linguaggio informale e incoerente: i documenti non elaborati spesso contengono un linguaggio informale o una terminologia incoerente. Ciò può confondere i modelli RAG. Ad esempio, le abbreviazioni che non sono definite nel documento o che sono già note al LLM potrebbero essere utilizzate in tutto il documento.
- Verbosità e ridondanza: i documenti non elaborati possono essere dettagliati e contenere informazioni non necessarie o ridondanti. Ciò può sovraccaricare i modelli RAG, portando a risposte meno concise e pertinenti. Gli esempi includono un documento che ripete le stesse informazioni più volte o più documenti che contengono informazioni simili o contraddittorie.
- Termini e frasi ambigui: i documenti non elaborati possono contenere termini o frasi ambigui che possono essere interpretati in diversi modi. Questa ambiguità può portare a interpretazioni errate da parte dei modelli RAG e a risposte imprecise. Ad esempio, un documento che utilizza un termine con significati multipli può generare una risposta che non è in linea con il significato previsto.
- Iniezione di elementi grafici e collegamenti ipertestuali: i documenti non elaborati che contengono immagini e informazioni sui collegamenti ipertestuali sono adatti al consumo umano. Tuttavia, questi elementi possono consumare il limite dei token di recupero. Il risultato è che alcuni estratti potrebbero essere incompleti. Ad esempio, la grafica e il collegamento ipertestuale URLs vengono

restituiti come parte del recupero, che utilizza i token di recupero e mancano le informazioni chiave dei paragrafi successivi.

- Mancanza di conoscenze o di contesto specifici del dominio: nei documenti non elaborati possono mancare le conoscenze o il contesto specifici del dominio necessari per una generazione accurata. Ciò può limitare la capacità dei modelli RAG di generare risposte pertinenti e accurate. Un esempio è un documento che fa riferimento a concetti specializzati senza fornire un contesto. Ciò potrebbe portare a risposte non significative nel dominio specificato.

Sebbene questo elenco non sia completo, fornisce alle aziende un punto di partenza per riflettere su cosa non funziona e perché. I documenti potrebbero presentare una o più di queste problematiche. La chiave per ottimizzare un'applicazione RAG consiste nell'utilizzare un set di documenti che rispettino le migliori pratiche di scrittura che ottimizzano il recupero.

Procedure ottimali di documentazione per le applicazioni RAG

Lo sviluppo di un'applicazione Retrieval-Augmented Generation (RAG) di successo richiede un'attenta considerazione di vari fattori relativi ai documenti per ottimizzarne le prestazioni. Le migliori pratiche in questa sezione sono curate sulla base dell'esperienza nella creazione di sistemi RAG con molti dirigenti di organizzazioni. Di seguito sono riportate alcune best practice chiave relative ai documenti per migliorare l'efficacia dell'applicazione RAG:

- Usa titoli e sottotitoli correttamente: l'organizzazione dei contenuti con titoli e sottotitoli chiari migliora la leggibilità e aiuta i modelli RAG a comprendere la struttura dei documenti. Questa pratica consente ai modelli di navigare meglio ed estrarre informazioni dai documenti, il che migliora la qualità delle risposte generate.
- Garantire che la numerazione sia sequenziale: quando si utilizzano elenchi numerati, è importante mantenere una numerazione corretta per evitare confusione. Assicurati che ogni voce dell'elenco sia numerata in sequenza senza saltare i numeri. Questo aiuta a mantenere la chiarezza e la coerenza dei contenuti.
- Aggiungere transizioni tra gli elementi dell'elenco: fornire transizioni tra gli elementi di un elenco puntato o numerato aiuta a guidare l'LLM attraverso il contenuto. Ad esempio, puoi usare frasi come «Dopo aver completato la fase 2, fai...» per collegare idee e migliorare il flusso di informazioni.
- Sostituisci le tabelle: evita di usare le tabelle. Formatta queste informazioni in elenchi puntati a più livelli o in una sintassi a livello semplice. La sintassi a livello piatto consiste nel disporre gli elementi o gli elementi allo stesso livello gerarchico, senza livelli annidati di subordinazione. Queste strutture aiutano a digerire le informazioni. LLMs Poiché la maggior parte dei documenti indicizzati viene letta da sinistra a destra, la sintassi a livello piatto consente alle informazioni di seguire in modo più coerente senza dover fare riferimento a una dimensione aggiuntiva. Questo formato è più adatto alle applicazioni RAG perché presenta le informazioni in modo strutturato e facilmente digeribile.
- Preelabora le informazioni grafiche per una maggiore efficienza: Multi-Modal LLMs può inserire sia immagini che testo. Riduci la risoluzione delle immagini, rimuovi le immagini ridondanti e descrivi il contenuto degli elementi grafici in formato testo. Queste misure migliorano il contesto significativo, evitano di consumare token inutilmente e migliorano l'accessibilità dei modelli RAG.
- Aggiungi iniziatori di sessione per domande comuni: quando rispondi a domande o attività comuni, come «Come posso ordinare un software?» , aggiungete uno starter di sessione che consenta

al lettore di entrare nel processo. Ad esempio, potresti aggiungere «Se stai cercando di ordinare del software, segui i passaggi seguenti...». Questo aiuta a creare una corrispondenza semantica elevata, che aiuta l'LLM a costruire una risposta coesa.

- Aggiungi un riepilogo a ciascuna sezione: dopo ogni titolo o sottotitolo, aggiungi un breve e conciso riepilogo del contenuto di quella sezione. Ciò può aumentare la copertura semantica e rafforzare i punti chiave. Ciò migliora l'accuratezza della ricerca di similarità all'interno dello spazio di incorporamento, migliorando così le prestazioni dell'applicazione RAG. Ciò è particolarmente utile se il documento è destinato sia al LLM che al consumo umano o se sono necessari elementi tabellari e grafici.
- Disambiguazione: i documenti devono essere concisi e mirati. LLMs generano risposte basate su estratti recuperati, quindi la disambiguazione aiuta il modello a utilizzare informazioni chiare e pertinenti. Ciò si traduce in risposte più accurate e informative.
- Definisci le abbreviazioni e imposta il contesto: LLMs vengono addestrati su grandi quantità di dati Internet e, nella maggior parte dei casi, non hanno il contesto dei documenti interni di un'azienda. Pertanto, impostare il contesto, definire le abbreviazioni ed evitare o definire la terminologia specifica dell'azienda aiuta l'LLM a comprendere i dati aziendali. Questo aiuta l'LLM a rispondere alle domande in modo più accurato e può aiutare a prevenire le allucinazioni.
- Ristruttura documenti di grandi dimensioni in documenti più piccoli per un'etichettatura e un'indicizzazione efficienti: evita di indicizzare un documento di grandi dimensioni che contiene più argomenti secondari. Prendi in considerazione la possibilità di dividere il documento di grandi dimensioni in documenti più piccoli e autonomi con titoli chiari. Ciò migliora l'indicizzazione e l'etichettatura.

Domande frequenti

Perché è importante ottimizzare i documenti per le applicazioni RAG?

I documenti non elaborati vengono spesso scritti per il consumo umano senza considerare i requisiti dei sistemi di intelligenza artificiale avanzati, come le applicazioni Retrieval Augmented Generation (RAG). L'ottimizzazione dei documenti seguendo le migliori pratiche può migliorare significativamente le prestazioni e la precisione delle applicazioni RAG fornendo informazioni strutturate, inequivocabili e pertinenti ai modelli.

Quali sono alcuni problemi comuni relativi ai documenti non elaborati che possono ostacolare le prestazioni RAG?

Alcune sfide chiave includono la mancanza di formattazione e metadati strutturati, un linguaggio informale o incoerente, la verbosità e la ridondanza, i termini e le frasi ambigui, l'inclusione di elementi di collegamenti ipertestuali e la mancanza di un contesto specifico del dominio. Questi problemi possono confondere i modelli RAG e portare a risposte imprecise o irrilevanti. Per ulteriori informazioni, consulta Sfide nei dati di origine che influiscono sulle applicazioni RAG in questa guida.

In che modo l'uso di titoli e sottotitoli può migliorare le prestazioni RAG?

Titoli e sottotitoli chiari aiutano i modelli RAG a comprendere la struttura e il contesto del contenuto. Ciò consente loro di navigare meglio ed estrarre le informazioni pertinenti dai documenti e migliora la qualità delle risposte generate. Per ulteriori informazioni, consulta la documentazione sulle migliori pratiche per le applicazioni RAG in questa guida.

Perché si consiglia di sostituire le informazioni della tabella con una sintassi a livello piatto?

Per i modelli RAG può essere difficile interpretare le tabelle perché richiedono una comprensione della struttura bidimensionale. La presentazione delle informazioni della tabella in una sintassi a

livello semplice o in un elenco puntato aiuta i modelli a elaborare più facilmente le informazioni, con conseguente miglioramento delle prestazioni. Per ulteriori informazioni, consultate la documentazione sulle migliori pratiche per le applicazioni RAG in questa guida.

In che modo l'aggiunta di riepiloghi può migliorare le prestazioni RAG?

L'inclusione di riassunti concisi all'inizio di ogni sezione o sottosezione può aumentare la copertura semantica e rafforzare i punti chiave. Ciò migliora l'accuratezza delle ricerche di similarità all'interno dello spazio di incorporamento, il che, in ultima analisi, migliora le prestazioni dell'applicazione RAG. Per ulteriori informazioni, consultate la documentazione sulle migliori pratiche per le applicazioni RAG in questa guida.

Perché è importante definire le abbreviazioni e impostare il contesto per? LLMs

LLMs sono formati su un'ampia gamma di dati, ma mancano di contesto per abbreviazioni o terminologie specifiche dell'azienda. Definire le abbreviazioni e fornire un contesto aiuta a LLMs comprendere e rispondere in modo più accurato. Questo può aiutare a prevenire allucinazioni o interpretazioni errate. Per ulteriori informazioni, consulta la documentazione sulle migliori pratiche per le applicazioni RAG in questa guida.

Risorse e passaggi successivi

Questa guida illustra una serie di sfide a livello di documento per le applicazioni RAG e le migliori pratiche per mitigarle. Questi apprendimenti derivano da interviste e discussioni con i leader del settore e sono supportati da casi d'uso aziendali.

Per iniziare a ottimizzare i documenti per le applicazioni RAG, consigliamo di condurre una verifica dei documenti esistenti. Identifica le aree che rappresentano una sfida per l'applicazione RAG. Gli esempi includono la mancanza di struttura, un linguaggio ambiguo o l'uso eccessivo di elementi grafici. Assegna priorità ai documenti a cui si accede di frequente o che sono fondamentali per le operazioni aziendali. Collabora con esperti in materia per implementare le migliori pratiche descritte in questa guida. Assicurati che i documenti siano ristrutturati con titoli chiari, un linguaggio conciso ed elementi contestuali. Per i nuovi documenti, stabilisci linee guida e modelli che garantiscano la coerenza e aiutino gli autori a rispettare le migliori pratiche. Inoltre, valuta la possibilità di investire in strumenti o servizi in grado di automatizzare alcuni aspetti del processo di ottimizzazione dei documenti, come l'utilizzo dell'intelligenza artificiale generativa per ristrutturare i documenti. Adottando un approccio proattivo all'ottimizzazione dei documenti, è possibile sbloccare tutto il potenziale delle applicazioni RAG e ottenere risultati più accurati e approfonditi in tutta l'organizzazione.

Le seguenti risorse possono aiutarvi a comprendere e creare applicazioni RAG nella vostra organizzazione.

Risorse

AWS documentazione

- Scelta di un database AWS vettoriale per i casi d'uso di RAG (AWS Prescriptive Guidance)
- Implementa uno use case RAG AWS utilizzando Terraform e Amazon Bedrock (Prescriptive Guidance)AWS
- Sviluppa assistenti avanzati basati sull'intelligenza artificiale generativa basati su chat utilizzando RAG e suggerendo (Prescriptive Guidance) ReAct AWS
- Recupera dati e genera risposte AI con Amazon Bedrock Knowledge Bases (documentazione Amazon Bedrock)
- Retrieval Augmented Generation (documentazione Amazon SageMaker AI)
- Opzioni e architetture di Retrieval Augmented Generation su (Prescriptive Guidance) AWSAWS

AWS Altre risorse

- Crea un assistente multimodale con RAG avanzato e Amazon Bedrock (post sul blog)AWS

Cronologia dei documenti

La tabella seguente descrive le modifiche significative apportate a questa guida. Per ricevere notifiche sugli aggiornamenti futuri, puoi abbonarti a un feed RSS.

Modifica	Descrizione	Data
Pubblicazione iniziale	—	24 luglio 2025

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.