Schreiben von Best Practices zur Optimierung von RAG-Anwendungen

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Schreiben von Best Practices zur Optimierung von RAG-Anwendungen

Table of Contents

Schreiben von Best Practices zur Optimierung von RAG-Anwendungen

Ivan Cui und Samantha Stuart, Amazon Web Services

Juli 2025 (Geschichte der Dokumente)

Große Sprachmodelle (LLMs) haben den Bereich der künstlichen Intelligenz mit ihrer bemerkenswerten Fähigkeit, menschenähnlichen Text zu verstehen und zu generieren, revolutioniert. Sie sind jedoch mit einer erheblichen Einschränkung konfrontiert: Sie können nur mit dem Wissen arbeiten, das in ihren Trainingsdaten enthalten ist. Hier hilft Retrieval Augmented Generation (RAG). Es bietet eine Lösung, die LLMs mit externen Wissensquellen wie den Daten und Dokumenten Ihres Unternehmens kombiniert werden kann. Durch einen zweistufigen Prozess, der das Abrufen von Informationen und die Generierung von Antworten umfasst, ermöglicht RAG KI-Systemen, auf up-to-date Informationen aus verschiedenen Quellen zuzugreifen und diese zu integrieren, was zu genaueren und fundierteren Antworten führt, die die Lücke zwischen statischem Modellwissen und dynamischem Informationsbedarf aus der realen Welt schließen.

Wie können Sie Inhalte für den Abruf in einer RAG-basierten Anwendung optimieren? Dieses Handbuch enthält bewährte Methoden, mit denen Sie die Formatierung und den Schreibstil von textbasierten Inhalten in der Wissensdatenbank optimieren können. Durch die Optimierung des Inhalts wird der Kontext verbessert, sodass RAG-Anwendungen aufgabenspezifische Informationen genauer verstehen können. Wenn das System hochrelevante und genaue Inhalte abruft, verbessert sich die Qualität der Antwort des LLM. Die Optimierung des Kontext-Bereitstellungsprozesses auf Systemebene wird als Kontext-Engineering bezeichnet und ist ein wesentlicher Bestandteil agentischer RAG-Architekturen. In agentic RAG werden vor der RAG-Ausführung ein oder mehrere zusätzliche LLMs Gründe und Aktionen auf Eingangsanforderungen angewendet. Dies erleichtert einen mehrstufigen Informationsbereitstellungsprozess. Da die RAG-Architekturen immer komplexer werden, ist die Optimierung von Quellinhalten nach wie vor das direkteste Mittel, um einen klaren Kontext zu liefern. LLMs Diese bewährten Methoden sollen Ihnen helfen, die Investition Ihres Unternehmens in eine RAG-Anwendung optimal zu nutzen.

Zielgruppe

Dieser Leitfaden richtet sich an KI-Ingenieure, Datenwissenschaftler, Dateningenieure oder Softwareentwickler, die LLM-Anwendungen mit einer oder mehreren RAG-Komponenten

erstellen. Um die Konzepte und Empfehlungen in diesem Handbuch zu verstehen, sollten Sie mit Vektordatenbanken und deren Eingabeaufforderungen vertraut sein. LLMs

Ziele

Die Empfehlungen in diesem Handbuch können Ihnen dabei helfen, Folgendes zu erreichen:

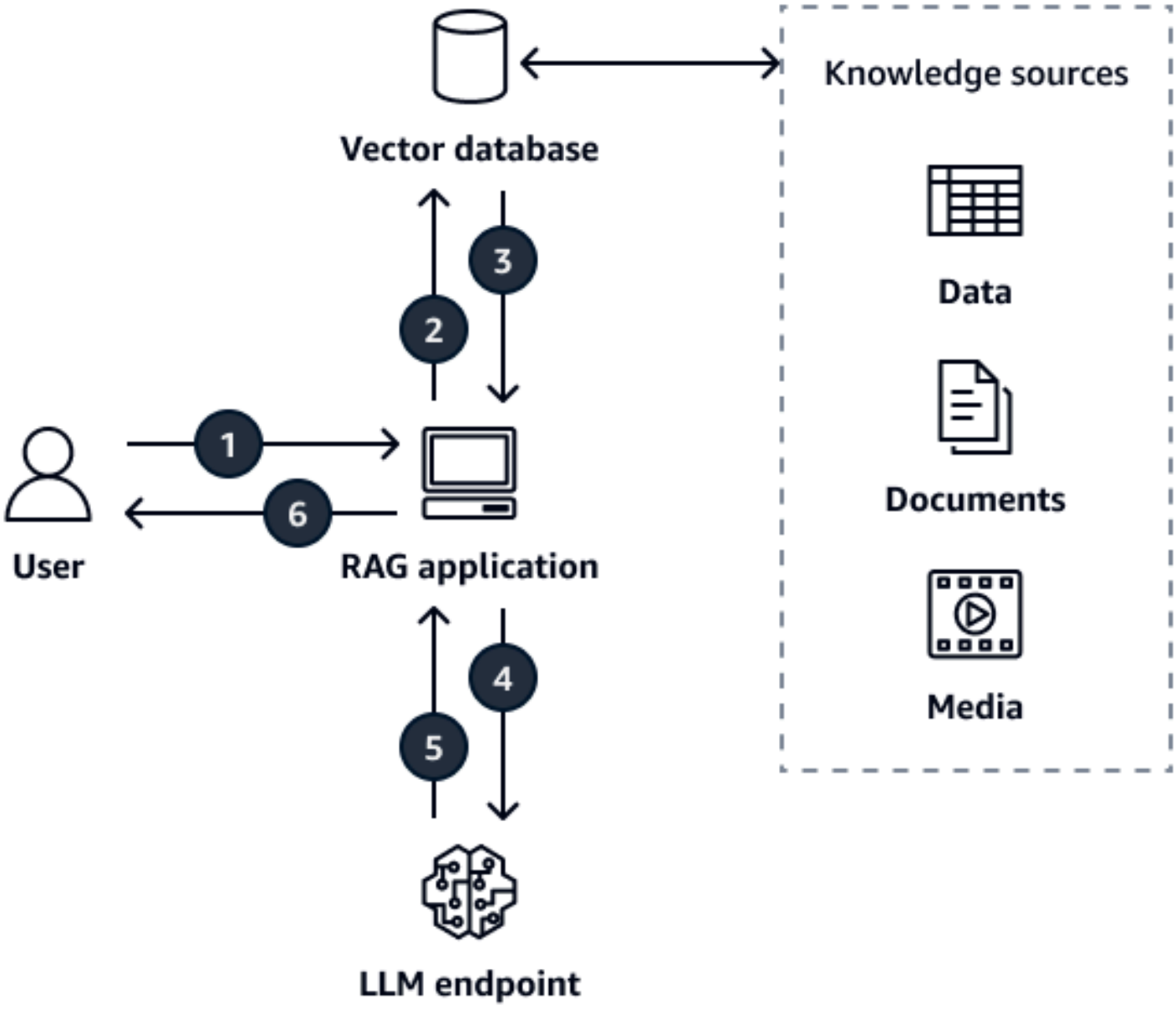
- Verbessern Sie die Genauigkeit und Relevanz der von RAG-Anwendungen generierten Antworten, indem Sie gut strukturierte und semantisch umfangreiche Quelldokumente bereitstellen, die für die Verwendung von Token und Redundanz optimiert sind.
- Helfen Sie RAG-Anwendungen, domänenspezifisches Wissen und Kontext besser zu verstehen, indem Sie klare Definitionen und Erklärungen in den Quelldokumenten bereitstellen.
- Erleichtern Sie die Wartung und Aktualisierung der Wissensdatenbank für RAG-Anwendungen, indem Sie in allen Quelldokumenten einheitliche Formatierungs- und Strukturierungsrichtlinien einhalten.
- Verbessern Sie die Skalierbarkeit von RAG-Lösungen, indem Sie große, monolithische Dokumente in kleinere, eigenständige Einheiten aufteilen, die effizient indexiert und abgerufen werden können.

LLMs und RAG verstehen

Um zu verstehen, wie die Verbesserung der Qualität von Quelldokumenten die Qualität einer RAG-Antwort verbessert, müssen Sie die internen Abläufe eines LLM verstehen. Die wahre Stärke von LLMs liegt in ihrer Fähigkeit, Selbstaufmerksamkeitsmechanismen und Transformatorarchitekturen zu nutzen. Diese fortschrittlichen Techniken ermöglichen es den Modellen, verschiedene Teile der Eingabesequenz effektiv zu verarbeiten und miteinander in Beziehung zu setzen, unabhängig von ihrer Position oder Entfernung innerhalb des Textes. Diese Fähigkeit steht in krassem Gegensatz zu herkömmlichen Sprachmodellen, die oft mit langfristigen Abhängigkeiten und dem Verständnis des Kontextes zu kämpfen haben. Darüber hinaus werden LLMs in einem noch nie dagewesenen Umfang geschult. Einige der größten Modelle bestehen aus Billionen von Parametern und haben Terabyte an Textdaten aus verschiedenen Quellen aufgenommen. Dieser enorme Umfang ermöglicht es LLMs, ein umfassendes Sprachverständnis zu entwickeln und subtile Nuancen, Redewendungen und kontextuelle Hinweise zu erfassen, die für KI-Systeme bisher eine Herausforderung darstellten. Das Ergebnis ist eine Klasse von Modellen, die in der Lage sind, kohärenten und fließenden Text zu generieren und bemerkenswerte Fähigkeiten bei Aufgaben wie der Beantwortung von Fragen, der Textzusammenfassung und sogar der Codegenerierung unter Beweis zu stellen.

Um diese Modelle zu nutzen, können wir uns an Dienste wie Amazon Bedrock wenden, das Zugriff auf eine Vielzahl von Fundamentmodellen von Amazon und Drittanbietern wie Anthropic, Cohere und Meta bietet. Sie können Amazon Bedrock verwenden, um mit hochmodernen Modellen zu experimentieren, sie anzupassen und zu optimieren oder sie über eine einzige API in Ihre generativen AI-powered Lösungen zu integrieren.

LLMs zeichnen sich zwar durch die Erfassung von Mustern und die Generierung von kohärentem Text aus, haben jedoch häufig keinen Zugang zu aktuellen oder speziellen Informationen. RAG kombiniert die generativen Fähigkeiten von LLMs mit einer Abrufkomponente, die im Rahmen der materialisierten LLM-Aufforderung auf relevante Informationen aus externen Quellen zugreifen und diese integrieren kann. Beispiele für externe Quellen sind Knowledge Bases für Amazon Bedrock, intelligente Suchsysteme wie Amazon Kendra oder Vektordatenbanken wie Amazon OpenSearch Service.



Das Diagramm beschreibt den folgenden Arbeitsablauf:

1. Der Benutzer sendet eine Anfrage an die RAG-Anwendung.
2. Die RAG-Anwendung fragt eine Vektordatenbank ab, die Wissensquellen wie Dokumente, Daten oder Medien enthält.
3. Die RAG-Anwendung ruft die relevanten Informationen auf der Grundlage semantischer Ähnlichkeiten zwischen der Abfrage und den gespeicherten Dokumenten aus der Vektordatenbank ab.

4. Die RAG-Anwendung erweitert die ursprüngliche Aufforderung um den abgerufenen Kontext und sendet ihn an den LLM-Endpunkt.
5. Der LLM-Endpunkt generiert eine Antwort und gibt sie an die RAG-Anwendung zurück.
6. Die RAG-Anwendung gibt die generierte Antwort an den Benutzer zurück.

RAG verwendet im Kern einen zweistufigen Prozess. In der ersten Phase identifiziert und ruft ein Abrufmodell relevante Dokumente oder Passagen auf der Grundlage der Eingabeabfrage ab. Bei diesem Abrufmodell kann es sich um ein herkömmliches Informationsabrufsystem, ein dichtes Abrufmodell oder eine Kombination aus beidem handeln. In der zweiten Phase werden die abgerufenen Informationen und die ursprüngliche Anfrage als vollständig materialisierte Eingabeaufforderungsvorlage in ein LLM eingespeist. LLMs hängen stark von der Qualität des Quellinhalts ab, der von der Retriever-Komponente geliefert wird. Sie wenden einen Mechanismus der Selbstaufmerksamkeit an, um mathematisch zu kodieren, wie sich der abgerufene Inhalt auf die Aufgabe bezieht. Das LLM generiert dann eine Antwort, die sowohl auf der Anfrage als auch auf den abgerufenen Informationen basiert. In RAG stellt die Kontrolle der Qualität der abgerufenen Quelldokumente ein direktes Mittel zur Verbesserung der internen Darstellung einer Aufgabe durch ein LLM dar. RAG ergänzt die Trainingsdaten des LLM effektiv mit relevanten externen Daten. Dieser Ansatz ermöglicht es der RAG, die Stärken sowohl von LLMs als auch von Retrievalsystemen zu nutzen und so genauere und fundiertere Antworten zu generieren, die aktuelles und spezialisiertes Wissen einbeziehen.

Vektoren und Einbettungen

Vektoren und Einbettungen sind grundlegende Konzepte des maschinellen Lernens und der Verarbeitung natürlicher Sprache. Vektoren sind mathematische Objekte, die Größen darstellen, die sowohl Größe als auch Richtung haben. Im Kontext der Verarbeitung natürlicher Sprache (NLP) werden Wörter, Sätze oder Dokumente häufig als Vektoren in hochdimensionalen Vektorräumen dargestellt. Einbettungen hingegen sind eine Möglichkeit, Objekte wie Wörter oder Dokumente in einem niederdimensionalen Vektorraum darzustellen, in dem die Beziehungen zwischen den Vektoren semantische oder syntaktische Ähnlichkeiten aufweisen. Worteinbettungen ermöglichen es beispielsweise Wörtern mit ähnlicher Bedeutung, ähnliche Vektordarstellungen zu haben. Dies hilft Algorithmen, Sprache effektiver zu verstehen und zu verarbeiten.

Vektor-Datenbanken

In der generativen KI ist eine Vektordatenbank eine Datenbank, die Vektordarstellungen von Dokumenten, Abfragen oder anderen Objekten speichert und verwaltet. Sie wurde entwickelt, um Vektoren effizient zu speichern und abzurufen. Dies unterstützt schnelle und skalierbare Operationen wie semantische Suche und Ähnlichkeitsabgleich. Vektordatenbanken indizieren Vektoren mithilfe spezialisierter Datenstrukturen wie Hierarchical Navigable Small World (HNSW) -Graphen oder K-Nearest Neighbors-Algorithmen (KNN). Diese Datenstrukturen ermöglichen eine schnelle Suche nach dem nächsten Nachbarn, sodass ähnliche Vektoren schnell in der Datenbank gefunden werden können.

Semantische Suche

Die semantische Suche ist eine Technik, mit der die Relevanz von Suchergebnissen verbessert wird, indem sie die Absicht und den Kontext der Anfrage versteht und nicht nur nach passenden Schlüsselwörtern sucht. Technisch gesehen beinhaltet die semantische Suche den Vergleich der Vektordarstellungen der Abfrage mit den Dokumenten in der Datenbank, um die relevantesten Treffer zu finden. Für die semantische Suche können verschiedene Abrufstrategien verwendet werden, darunter, aber nicht beschränkt auf:

- HNSW — Eine graphenbasierte Datenstruktur, die Vektoren so organisiert, dass die Suche nach nächstgelegenen Nachbarn effizient ist.
- KNN — Ein Algorithmus, der anhand einer Entfernungsmetrik, z. B. der Kosinusähnlichkeit, die K Vektoren ermittelt, die einem Abfragevektor am nächsten sind.
- Kosinusähnlichkeit — Ein Maß für die Ähnlichkeit zwischen zwei Vektoren, die ungleich Null sind, das den Kosinus des Winkels zwischen ihnen misst. Es wird häufig bei der semantischen Suche verwendet, um die Richtung von Vektoren in einem hochdimensionalen Raum zu vergleichen.
- Locality-sensitive Hashing (LSH) — Eine Technik, bei der ähnliche Vektoren mit hoher Wahrscheinlichkeit in dieselben oder benachbarte Buckets gehasht werden. Dies ermöglicht die ungefähre Suche nach dem nächsten Nachbarn, was schneller sein kann als exakte Suchen in hochdimensionalen Räumen.

Herausforderungen bei Quelldaten, die sich auf RAG-Anwendungen auswirken

Eine der größten Herausforderungen bei der Entwicklung einer optimalen RAG-Anwendung (Retrieval-Augmented Generation) liegt in der Art der verwendeten Rohdaten oder Dokumente. Häufig verwenden Unternehmen bestehende Dokumente, die als menschliche Referenz erstellt wurden. Diese Dokumente enthalten häufig Hyperlinks und Screenshots mit Bildern, um das Verständnis zu fördern. Diese Elemente behindern jedoch den semantischen Abruf aufgrund von Token-Beschränkungen für Auszüge. Dies führt zu einer schlechten Leistung des Retrievers.

Im Folgenden sind die häufigsten Probleme mit Rohdokumenten aufgeführt, die bei einer optimalen RAG-Anwendung auftreten:

- Fehlende strukturierte Formatierung und Metadaten — In Rohdokumenten können klare Abschnittsüberschriften, Zwischenüberschriften oder Metadaten fehlen. Dies macht es schwierig, relevante Informationen zu identifizieren und zu extrahieren. Ein langes Dokument ohne klare Überschriften kann es beispielsweise schwierig machen, den Kontext bestimmter Informationen zu bestimmen.
- Informelle und inkonsistente Sprache — Rohdokumente enthalten oft eine informelle Sprache oder eine inkonsistente Terminologie. Dies kann RAG-Modelle verwirren. Beispielsweise können Abkürzungen, die im Dokument nicht definiert sind oder dem LLM bereits bekannt sind, überall in einem Dokument verwendet werden.
- Ausführlichkeit und Redundanz — Rohdokumente können ausführlich sein und unnötige oder redundante Informationen enthalten. Dies kann die RAG-Modelle überfordern und zu weniger präzisen und relevanteren Antworten führen. Beispiele hierfür sind ein Dokument, das dieselben Informationen mehrfach wiederholt, oder mehrere Dokumente, die ähnliche oder widersprüchliche Informationen enthalten.
- Mehrdeutige Begriffe und Ausdrücke — Rohdokumente können mehrdeutige Begriffe oder Ausdrücke enthalten, die auf unterschiedliche Weise interpretiert werden können. Diese Mehrdeutigkeit kann zu Fehlinterpretationen durch RAG-Modelle und zu ungenauen Antworten führen. Beispielsweise kann ein Dokument, das einen Begriff mit mehreren Bedeutungen verwendet, zu einer Antwort führen, die nicht der beabsichtigten Bedeutung entspricht.
- Einfügen von Grafik- und Hyperlink-Elementen — Rohdokumente, die Grafiken und Hyperlink-Informationen enthalten, eignen sich gut für den menschlichen Gebrauch. Diese Elemente können jedoch das Limit für Abruf-Tokens überschreiten. Das Ergebnis ist, dass Auszüge möglicherweise

unvollständig sind. Beispielsweise URLs werden Grafiken und Hyperlinks als Teil des Abrufs zurückgegeben, wodurch die Abruf-Token verbraucht werden, und wichtige Informationen aus nachfolgenden Absätzen fehlen.

- Mangelndes domänenspezifisches Wissen oder Kontext — In Rohdokumenten fehlt möglicherweise das für eine korrekte Generierung erforderliche domänenspezifische Wissen oder der Kontext. Dies kann die Fähigkeit von RAG-Modellen einschränken, relevante und genaue Antworten zu generieren. Ein Beispiel ist ein Dokument, das auf spezielle Konzepte verweist, ohne Kontext bereitzustellen. Dies kann zu Antworten führen, die in der angegebenen Domäne nicht aussagekräftig sind.

Diese Liste ist zwar nicht vollständig, bietet Unternehmen jedoch einen Ausgangspunkt, um darüber nachzudenken, was nicht funktioniert und warum. Dokumente können mit einer oder mehreren dieser Herausforderungen konfrontiert sein. Der Schlüssel zur Optimierung einer RAG-Anwendung besteht darin, eine Reihe von Dokumenten zu verwenden, die den bewährten Schreibmethoden entsprechen, um den Abruf zu optimieren.

Bewährte Methoden der Dokumentation für RAG-Anwendungen

Die Entwicklung einer erfolgreichen RAG-Anwendung (Retrieval-Augmented Generation) erfordert die sorgfältige Berücksichtigung verschiedener dokumentbezogener Faktoren, um die Leistung zu optimieren. Die Best Practices in diesem Abschnitt basieren auf den Erfahrungen vieler Unternehmensleiter beim Aufbau von RAG-Systemen. Im Folgenden finden Sie einige wichtige bewährte Methoden für Dokumente, mit denen Sie die Effektivität Ihrer RAG-Anwendung verbessern können:

- Verwenden Sie Überschriften und Zwischenüberschriften richtig — Die Organisation Ihrer Inhalte mit klaren Überschriften und Zwischenüberschriften verbessert die Lesbarkeit und hilft RAG-Modellen, die Struktur Ihrer Dokumente zu verstehen. Diese Vorgehensweise ermöglicht es den Modellen, besser zu navigieren und Informationen aus den Dokumenten zu extrahieren, was die Qualität der generierten Antworten verbessert.
- Stellen Sie sicher, dass die Nummerierung fortlaufend ist — Bei der Verwendung nummerierter Listen ist es wichtig, die korrekte Nummerierung beizubehalten, um Verwechslungen zu vermeiden. Stellen Sie sicher, dass jedes Listenelement fortlaufend nummeriert ist, ohne dass Zahlen übersprungen werden. Dies trägt dazu bei, die Klarheit und Kohärenz Ihrer Inhalte zu wahren.
- Fügen Sie Übergänge zwischen Listenelementen hinzu — Die Bereitstellung von Übergängen zwischen Elementen in einer Liste mit Aufzählungszeichen oder Nummern hilft dem LLM, sich durch den Inhalt zu führen. Sie können beispielsweise Formulierungen wie „Nachdem Sie Schritt 2 abgeschlossen haben, tun Sie...“ verwenden, um Ideen miteinander zu verknüpfen und den Informationsfluss zu verbessern.
- Tabellen ersetzen — Vermeiden Sie die Verwendung von Tabellen. Formatieren Sie diese Informationen in Aufzählungen mit mehreren Ebenen oder in einer einfachen Syntax. Bei der Syntax auf flacher Ebene werden Elemente oder Elemente auf derselben Hierarchieebene ohne verschachtelte Unterordnungsebenen angeordnet. Diese Strukturen helfen dabei LLMs , die Informationen zu verdauen. Da die meisten indizierten Dokumente von links nach rechts gelesen werden, ermöglicht die Flat-Level-Syntax, dass Informationen kohärenter folgen können, ohne dass auf eine zusätzliche Dimension verwiesen werden muss. Dieses Format eignet sich besser für RAG-Anwendungen, da es Informationen strukturiert und leicht verdaulich darstellt.
- Aus Effizienzgründen grafische Informationen vorab verarbeiten — Multimodal LLMs kann sowohl Bild als auch Text aufnehmen. Reduzieren Sie die Auflösung von Bildern, entfernen Sie überflüssige Bilder und beschreiben Sie den Inhalt grafischer Elemente im Textformat. Diese

Maßnahmen verbessern den aussagekräftigen Kontext, vermeiden den unnötigen Verbrauch von Tokens und verbessern die Zugänglichkeit für RAG-Modelle.

- Fügen Sie Sitzungsstarter für häufig gestellte Fragen hinzu — bei der Beantwortung häufiger Fragen oder Aufgaben wie „Wie bestelle ich Software?“ , fügen Sie einen Sitzungsstarter hinzu, der den Leser in den Prozess überführt. Sie könnten zum Beispiel hinzufügen: „Wenn Sie Software bestellen möchten, gehen Sie wie folgt vor...“. Dies trägt dazu bei, einen hohen semantischen Abgleich zu erreichen, was dem LLM hilft, eine kohärente Antwort zu erstellen.
- Fügen Sie jedem Abschnitt eine Zusammenfassung hinzu — Fügen Sie nach jeder Überschrift oder Unterüberschrift eine kurze und präzise Zusammenfassung des Inhalts in diesem Abschnitt hinzu. Dies kann die semantische Abdeckung erhöhen und wichtige Punkte unterstreichen. Dies verbessert die Genauigkeit der Ähnlichkeitssuche innerhalb des Einbettungsbereichs und verbessert somit die Leistung der RAG-Anwendung. Dies ist besonders hilfreich, wenn das Dokument sowohl für LLM als auch für den menschlichen Gebrauch bestimmt ist oder wenn tabellarische und grafische Elemente erforderlich sind.
- Begriffsklärung — Dokumente sollten präzise und zielgerichtet sein. LLMs Generieren Sie Antworten auf der Grundlage der abgerufenen Auszüge, sodass die Begriffsklärung dem Modell hilft, klare und relevante Informationen zu verwenden. Dies führt zu genaueren und aussagekräftigeren Antworten.
- Definieren Sie Abkürzungen und legen Sie den Kontext fest — Sie LLMs sind mit großen Mengen an Internetdaten vertraut und haben in den meisten Fällen nicht den Kontext interner Dokumente eines Unternehmens. Daher hilft es dem LLM, Ihre Unternehmensdaten zu verstehen, den Kontext festzulegen, Abkürzungen zu definieren und unternehmensspezifische Terminologie zu vermeiden oder zu definieren. Dies hilft dem LLM, Fragen genauer zu beantworten, und kann Halluzinationen vorbeugen.
- Strukturieren Sie große Dokumente in kleinere Dokumente für effizientes Taggen und Indexieren — Vermeiden Sie die Indexierung eines großen Dokuments, das mehrere Unterthemen enthält. Erwägen Sie, das große Dokument in kleinere, eigenständige Dokumente mit eindeutigen Titeln aufzuteilen. Dadurch werden Indexierung und Tagging verbessert.

Häufig gestellte Fragen

Warum ist es wichtig, Dokumente für RAG-Anwendungen zu optimieren?

Rohdokumente werden häufig für den menschlichen Gebrauch geschrieben, ohne die Anforderungen fortschrittlicher KI-Systeme wie Retrieval Augmented Generation (RAG) -Anwendungen zu berücksichtigen. Die Optimierung von Dokumenten durch die Befolgung bewährter Verfahren kann die Leistung und Genauigkeit von RAG-Anwendungen erheblich verbessern, indem strukturierte, eindeutige und relevante Informationen für die Modelle bereitgestellt werden.

Was sind einige der häufigsten Probleme bei Rohdokumenten, die die Leistung von RAG beeinträchtigen können?

Zu den wichtigsten Herausforderungen gehören der Mangel an strukturierter Formatierung und Metadaten, informelle oder inkonsistente Sprache, Ausführlichkeit und Redundanz, mehrdeutige Begriffe und Ausdrücke, die Aufnahme von Hyperlink-Elementen und das Fehlen eines domänenspezifischen Kontextes. Diese Probleme können die RAG-Modelle verwirren und zu ungenauen oder irrelevanten Antworten führen. Weitere Informationen finden Sie in diesem Handbuch unter Herausforderungen bei Quelldaten, die sich auf RAG-Anwendungen auswirken.

Wie kann die Verwendung von Überschriften und Zwischenüberschriften die Leistung von RAG verbessern?

Klare Überschriften und Zwischenüberschriften helfen RAG-Modellen, die Struktur und den Kontext des Inhalts zu verstehen. Dadurch können sie besser navigieren und relevante Informationen aus den Dokumenten extrahieren, und die Qualität der generierten Antworten wird verbessert. Weitere Informationen finden Sie in diesem Handbuch unter Bewährte Methoden zur Dokumentation für RAG-Anwendungen.

Warum wird empfohlen, Tabelleninformationen durch eine Flat-Level-Syntax zu ersetzen?

Für RAG-Modelle kann es schwierig sein, Tabellen zu interpretieren, da sie ein Verständnis der zweidimensionalen Struktur erfordern. Die Darstellung von Tabelleninformationen in einer einfachen Syntax oder einer Aufzählung hilft Modellen, die Informationen einfacher zu verarbeiten, was zu einer besseren Leistung führt. Weitere Informationen finden Sie in diesem Handbuch unter Bewährte Methoden zur Dokumentation für RAG-Anwendungen.

Wie kann das Hinzufügen von Zusammenfassungen die Leistung von RAG verbessern?

Das Hinzufügen von kurzen Zusammenfassungen am Anfang jedes Abschnitts oder Unterabschnitts kann die semantische Abdeckung erhöhen und wichtige Punkte unterstreichen. Dies verbessert die Genauigkeit von Ähnlichkeitssuchen innerhalb des Einbettungsbereichs, was letztlich die Leistung der RAG-Anwendung verbessert. Weitere Informationen finden Sie in diesem Handbuch unter Bewährte Methoden zur Dokumentation für RAG-Anwendungen.

Warum ist es wichtig, Abkürzungen zu definieren und den Kontext dafür LLMs festzulegen?

LLMs sind an einem breiten Datenspektrum geschult, aber ihnen fehlt der Kontext für unternehmensspezifische Abkürzungen oder Terminologie. Die Definition von Abkürzungen und die Bereitstellung von Kontext helfen dabei, sie LLMs besser zu verstehen und genauer darauf zu reagieren. Dies kann dazu beitragen, Halluzinationen oder Fehlinterpretationen vorzubeugen. Weitere Informationen finden Sie in der Dokumentation zu bewährten Methoden für RAG-Anwendungen in diesem Handbuch.

Nächste Schritte und Ressourcen

In diesem Leitfaden werden eine Reihe von Herausforderungen für RAG-Anwendungen auf Dokumentenebene sowie bewährte Verfahren zu deren Behebung beschrieben. Diese Erkenntnisse stammen aus Interviews und Diskussionen mit Branchenführern und werden durch Anwendungsfälle in Unternehmen untermauert.

Um mit der Optimierung Ihrer Dokumente für RAG-Anwendungen zu beginnen, empfehlen wir, eine Prüfung Ihrer vorhandenen Dokumente durchzuführen. Identifizieren Sie Bereiche, die die RAG-Anwendung vor Herausforderungen stellen. Beispiele hierfür sind mangelnde Struktur, mehrdeutige Sprache oder übermäßige Verwendung grafischer Elemente. Priorisieren Sie Dokumente, auf die häufig zugegriffen wird oder die für Ihren Geschäftsbetrieb von entscheidender Bedeutung sind. Arbeiten Sie mit Fachexperten zusammen, um die Best Practices in diesem Leitfaden umzusetzen. Stellen Sie sicher, dass die Dokumente mit klaren Überschriften, einer präzisen Sprache und kontextsensitiven Elementen neu strukturiert sind. Erstellen Sie für neue Dokumente Richtlinien und Vorlagen, die für Konsistenz sorgen und den Autoren helfen, sich an die bewährten Verfahren zu halten. Erwägen Sie außerdem, in Tools oder Dienste zu investieren, mit denen Aspekte des Dokumentenoptimierungsprozesses automatisiert werden können, z. B. den Einsatz generativer KI zur Restrukturierung von Dokumenten. Durch einen proaktiven Ansatz zur Dokumentenoptimierung können Sie das volle Potenzial der RAG-Anwendungen ausschöpfen und in Ihrem gesamten Unternehmen genauere und aussagekräftigere Ergebnisse erzielen.

Die folgenden Ressourcen können Ihnen helfen, RAG-Anwendungen in Ihrem Unternehmen zu verstehen und zu entwickeln.

Ressourcen

AWS Dokumentation

- Auswahl einer AWS Vektordatenbank für RAG-Anwendungsfälle (AWS Prescriptive Guidance)
- Stellen Sie mithilfe AWS von Terraform und Amazon Bedrock einen RAG-Anwendungsfall bereit (AWS Prescriptive Guidance)
- Entwickeln Sie mithilfe von RAG und Prompting (Prescriptive Guidance) fortschrittliche, auf Chats basierende, generative KI-Assistenten ReAct AWS
- Rufen Sie Daten ab und generieren Sie KI-Antworten mit Amazon Bedrock Knowledge Bases (Amazon Bedrock-Dokumentation)

- Erweiterte Generierung abrufen (Amazon SageMaker AI-Dokumentation)
- Optionen und Architekturen der erweiterten Generation abrufen auf AWS(Prescriptive Guidance)AWS

Andere Ressourcen AWS

- Erstellen Sie einen multimodalen Assistenten mit fortschrittlichem RAG und Amazon Bedrock (AWS Blogbeitrag)

Dokumentverlauf

In der folgenden Tabelle werden wichtige Änderungen in diesem Leitfaden beschrieben. Um Benachrichtigungen über zukünftige Aktualisierungen zu erhalten, können Sie einen RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Erste Veröffentlichung	—	24. Juli 2025

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.