Redacción de mejores prácticas para optimizar las aplicaciones RAG

AWS Orientación prescriptiva

AWS Orientación prescriptiva: Redacción de mejores prácticas para optimizar las aplicaciones RAG

Table of Contents

Redacción de mejores prácticas para optimizar las aplicaciones RAG

Ivan Cui y Samantha Stuart, Amazon Web Services

Julio de 2025 (historial del documento)

Los grandes modelos lingüísticos (LLMs) han revolucionado el campo de la inteligencia artificial con su notable capacidad para comprender y generar textos de aspecto humano. Sin embargo, se enfrentan a una limitación importante: solo pueden trabajar con el conocimiento contenido en sus datos de entrenamiento. Aquí es donde la generación aumentada de recuperación (RAG) ayuda. Ofrece una solución que se combina LLMs con fuentes de conocimiento externas, como los datos y documentos de su organización. Mediante un proceso de dos etapas que incluye la recuperación de información y la generación de respuestas, el RAG permite a los sistemas de IA acceder a up-to-date información de diversas fuentes e incorporarla, lo que da como resultado respuestas más precisas e informadas que cierran la brecha entre el conocimiento de los modelos estáticos y las necesidades de información dinámicas del mundo real.

¿Cómo se puede optimizar el contenido para su recuperación en una aplicación basada en RAG? Esta guía proporciona las mejores prácticas para ayudarle a optimizar el formato y el estilo de escritura del contenido basado en texto en la base de conocimientos. La optimización del contenido mejora el contexto, lo que ayuda a las aplicaciones RAG a comprender la información específica de la tarea con mayor precisión. Cuando el sistema recupera contenido muy relevante y preciso, la calidad de la respuesta del LLM mejora. La optimización del proceso de entrega de contexto a nivel de sistema se denomina ingeniería de contexto y constituye una parte esencial de las arquitecturas RAG de las agencias. En el RAG de una agencia, hay una o más LLMs razones adicionales para actuar en función de las solicitudes recibidas antes de ejecutar el RAG. Esto facilita un proceso de entrega de información de varios pasos. A medida que las arquitecturas RAG se vuelven cada vez más complejas, la optimización del contenido fuente sigue siendo el medio más directo de ofrecer un contexto claro. LLMs Estas mejores prácticas están diseñadas para ayudarlo a maximizar la inversión de su organización en una aplicación RAG.

Destinatarios previstos

Esta guía está destinada a ingenieros de IA, científicos de datos, ingenieros de datos o desarrolladores de software que estén creando aplicaciones LLM con uno o más componentes de

RAG. Para comprender los conceptos y las recomendaciones de esta guía, debe estar familiarizado con las bases de datos vectoriales y las instrucciones correspondientes. LLMs

Objetivos

Las recomendaciones de esta guía pueden ayudarle a lograr lo siguiente:

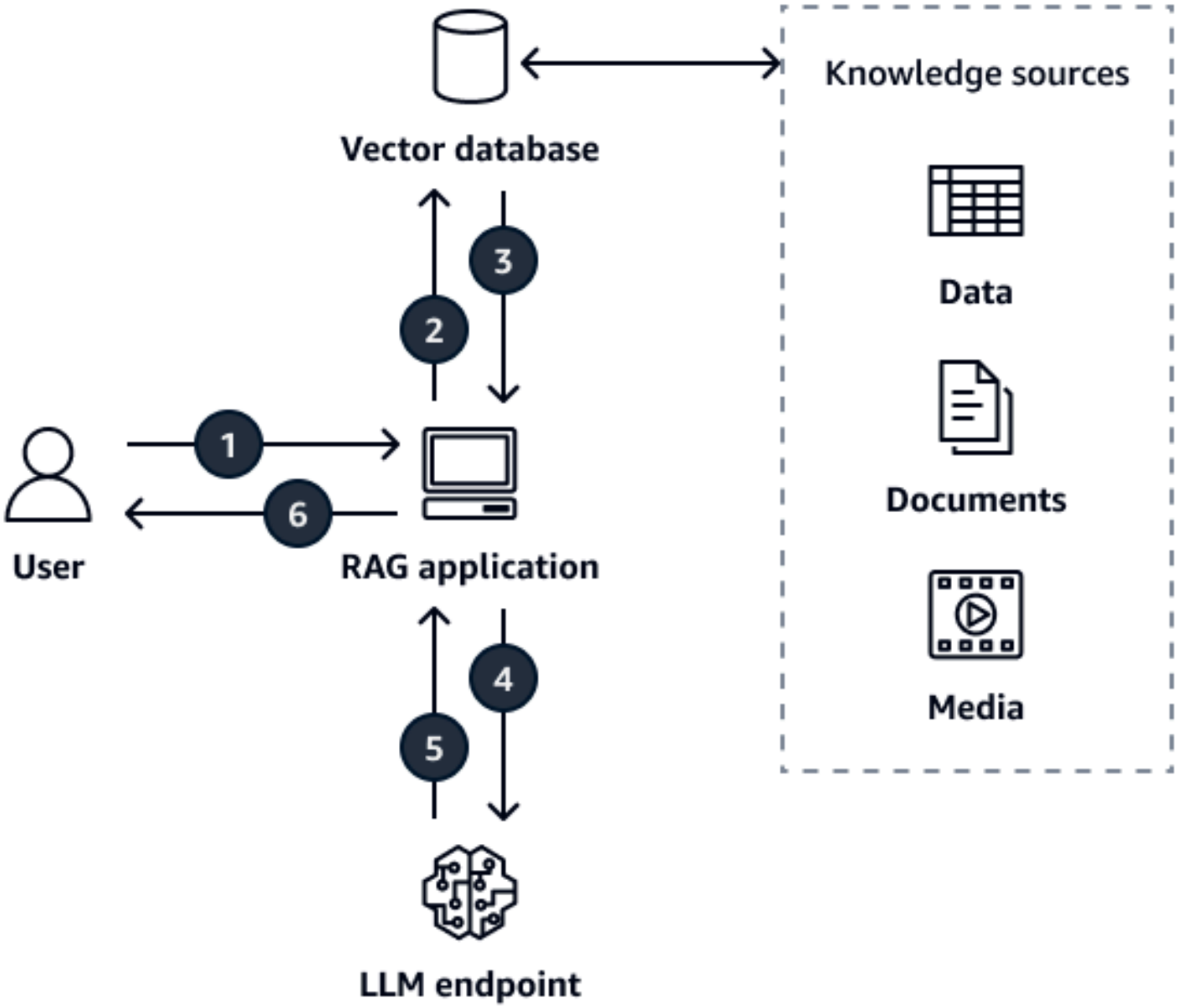
- Mejore la precisión y la relevancia de las respuestas generadas por las aplicaciones RAG proporcionando documentos fuente bien estructurados y con una gran riqueza semántica, optimizados para el uso de fichas y la redundancia.
- Ayude a las aplicaciones de RAG a comprender mejor el conocimiento y el contexto específicos del dominio proporcionando definiciones y explicaciones claras en los documentos fuente.
- Facilite el mantenimiento y las actualizaciones de la base de conocimientos de las aplicaciones RAG al seguir las pautas de formato y estructuración uniformes en todos los documentos fuente.
- Mejore la escalabilidad de las soluciones RAG dividiendo los documentos grandes y monolíticos en unidades más pequeñas e independientes que se puedan indexar y recuperar de manera eficiente.

Comprensión de los LLM y los RAG

Para comprender cómo mejorar la calidad del documento fuente mejora la calidad de la respuesta de un RAG, debe comprender el funcionamiento interno de un LLM. El verdadero poder de los LLM reside en su capacidad para utilizar mecanismos de autoatención y arquitecturas transformadoras. Estas técnicas avanzadas permiten a los modelos procesar y relacionar eficazmente diferentes partes de la secuencia de entrada, independientemente de su posición o distancia dentro del texto. Esta capacidad contrasta marcadamente con los modelos lingüísticos tradicionales, que a menudo tienen dificultades para comprender el contexto y las dependencias de largo alcance. Además, los LLM se capacitan a una escala sin precedentes. Algunos de los modelos más grandes están compuestos por billones de parámetros y han absorbido terabytes de datos textuales de diversas fuentes. Esta escala masiva permite a los LLM desarrollar una rica comprensión del lenguaje, capturando matices sutiles, expresiones idiomáticas y claves contextuales que antes suponían un desafío para los sistemas de IA. El resultado es una clase de modelos que pueden generar textos coherentes y fluidos y que demuestran una capacidad extraordinaria en tareas como la respuesta a preguntas, el resumen de textos e incluso la generación de código.

Para utilizar estos modelos, podemos recurrir a servicios como Amazon Bedrock, que proporciona acceso a una variedad de modelos básicos de Amazon y de proveedores externos, incluidos Anthropic, Cohere y Meta. Puede usar Amazon Bedrock para experimentar con modelos de última generación, personalizarlos y ajustarlos, o incorporarlos a sus AI-powered soluciones generativas a través de una sola API.

Si bien los LLM destacan por la captura de patrones y la generación de texto coherente, a menudo carecen de acceso a información actualizada o especializada. El RAG combina el poder generativo de los LLM con un componente de recuperación que permite acceder a información relevante de fuentes externas e incorporarla, como parte del mensaje LLM materializado. Algunos ejemplos de fuentes externas son las bases de conocimiento de Amazon Bedrock, los sistemas de búsqueda inteligente como Amazon Kendra o las bases de datos vectoriales como Amazon OpenSearch Service.



El diagrama describe el siguiente flujo de trabajo:

1. El usuario envía una consulta a la aplicación RAG.
2. La aplicación RAG consulta una base de datos vectorial que contiene fuentes de conocimiento, como documentos, datos o medios.
3. La aplicación RAG recupera la información relevante de la base de datos vectorial en función de las similitudes semánticas entre la consulta y los documentos almacenados.
4. La aplicación RAG amplía la solicitud original con el contexto recuperado y la envía al punto final de LLM.

5. El punto final LLM genera una respuesta y la devuelve a la aplicación RAG.
6. La aplicación RAG devuelve la respuesta generada al usuario.

En esencia, RAG emplea un proceso de dos etapas. En la primera etapa, un modelo de recuperación identifica y recupera los documentos o pasajes relevantes en función de la consulta de entrada. Este modelo de recuperación puede ser un sistema de recuperación de información tradicional, un modelo de recuperación densa o una combinación de ambos. En la segunda etapa, la información recuperada y la consulta original se introducen en un LLM como una plantilla de solicitud totalmente materializada. Los LLM dependen en gran medida de la calidad del contenido fuente entregado por el componente de recuperación. Aplican un mecanismo de autoatención para codificar matemáticamente la relación entre el contenido recuperado y la tarea. Luego, el LLM genera una respuesta basada tanto en la consulta como en la información recuperada. En el RAG, controlar la calidad de los documentos fuente recuperados representa un medio directo de mejorar la representación interna de una tarea por parte de un LLM. El RAG amplía eficazmente los datos de formación del LLM con datos externos relevantes. Este enfoque permite a RAG aprovechar las ventajas tanto de los LLM como de los sistemas de recuperación, lo que permite generar respuestas más precisas e informadas que incorporan conocimientos actuales y especializados.

Vectores e incrustaciones

Los vectores y las incrustaciones son conceptos fundamentales en el aprendizaje automático y el procesamiento del lenguaje natural. Los vectores son objetos matemáticos que representan cantidades que tienen magnitud y dirección. En el contexto del procesamiento del lenguaje natural (PNL), las palabras, las oraciones o los documentos suelen representarse como vectores en espacios vectoriales de alta dimensión. Las incrustaciones, por otro lado, son una forma de representar objetos como palabras o documentos en un espacio vectorial de dimensiones inferiores donde las relaciones entre los vectores capturan similitudes semánticas o sintácticas. Las incrustaciones de palabras, por ejemplo, permiten que palabras con significados similares tengan representaciones vectoriales similares. Esto ayuda a los algoritmos a entender y procesar el lenguaje de forma más eficaz.

bases de datos vectoriales

En la IA generativa, una base de datos vectorial es una base de datos que almacena y gestiona representaciones vectoriales de documentos, consultas u otros objetos. Está diseñada para almacenar y recuperar vectores de manera eficiente. Esto permite operaciones rápidas y escalables,

como la búsqueda semántica y la coincidencia de similitudes. Las bases de datos vectoriales indexan los vectores mediante estructuras de datos especializadas, como gráficos jerárquicos de mundos pequeños navegables (HNSW) o algoritmos de K-Nearest vecinos (KNN). Estas estructuras de datos permiten realizar búsquedas rápidas de los vecinos más cercanos, lo que permite encontrar rápidamente vectores similares en la base de datos.

Búsqueda semántica

La búsqueda semántica es una técnica que mejora la relevancia de los resultados de búsqueda al comprender la intención y el contexto de la consulta, en lugar de limitarse a hacer coincidir las palabras clave. En términos técnicos, la búsqueda semántica implica comparar las representaciones vectoriales de la consulta y los documentos de la base de datos para encontrar las coincidencias más relevantes. Se pueden utilizar diferentes estrategias de recuperación para la búsqueda semántica, que incluyen, entre otras:

- HNSW: estructura de datos basada en gráficos que organiza los vectores de manera que sea eficiente buscar los vecinos más cercanos.
- KNN: algoritmo que busca los K vectores más cercanos a un vector de consulta en función de una métrica de distancia, como la similitud de cosenos.
- Similitud de coseno: medida de similitud entre dos vectores distintos de cero que mide el coseno del ángulo que los separa. Suele utilizarse en la búsqueda semántica para comparar la dirección de los vectores en un espacio de alta dimensión.
- Locality-sensitive hash (LSH): técnica que utiliza códigos hash de vectores similares en los mismos cubos o en cubos cercanos con una alta probabilidad. Esto permite realizar búsquedas aproximadas de los vecinos más cercanos, que pueden ser más rápidas que las búsquedas exactas en espacios de alta dimensión.

Desafíos en los datos de origen que afectan a las aplicaciones RAG

Uno de los principales desafíos a la hora de desarrollar una aplicación óptima de generación aumentada (RAG) reside en la naturaleza de los datos o documentos sin procesar que se utilizan. A menudo, las empresas utilizan documentos existentes que se crearon como referencia humana. Estos documentos suelen incluir hipervínculos y capturas de pantalla de imágenes para fomentar la comprensión. Sin embargo, estos elementos obstruyen la recuperación semántica debido a los límites de los fragmentos simbólicos. Esto se traduce en un rendimiento deficiente del recuperador.

Los siguientes son los desafíos más comunes relacionados con los documentos sin procesar para una aplicación RAG óptima:

- Falta de metadatos y formatos estructurados: los documentos sin procesar pueden carecer de encabezados de sección, subtítulos o metadatos claros. Esto dificulta la identificación y extracción de la información relevante. Por ejemplo, un documento extenso sin encabezados claros puede dificultar la determinación del contexto de información específica.
- Lenguaje informal e incoherente: los documentos sin procesar suelen contener un lenguaje informal o una terminología incoherente. Esto puede confundir a los modelos RAG. Por ejemplo, las abreviaturas que no están definidas en el documento o que el LLM ya conoce pueden usarse en todo el documento.
- Verbosidad y redundancia: los documentos sin procesar pueden ser detallados y contener información innecesaria o redundante. Esto puede abrumar a los modelos RAG y generar respuestas menos concisas y relevantes. Los ejemplos incluyen un documento que repite la misma información varias veces o varios documentos que contienen información similar o contradictoria.
- Términos y frases ambiguos: los documentos sin procesar pueden contener términos o frases ambiguos que pueden interpretarse de varias maneras. Esta ambigüedad puede provocar interpretaciones erróneas por parte de los modelos RAG y respuestas inexactas. Por ejemplo, un documento que usa un término con múltiples significados puede generar una respuesta que no se alinee con el significado deseado.
- Inyección de elementos gráficos y de hipervínculos: los documentos sin procesar que contienen gráficos e información de hipervínculos funcionan bien para el consumo humano. Sin embargo, estos elementos pueden consumir el límite de fichas de recuperación. El resultado es que los extractos pueden estar incompletos. Por ejemplo, los gráficos y el hipervínculo URLs se devuelven

como parte de la recuperación, lo que agota los símbolos de recuperación, y falta la información clave de los párrafos siguientes.

- Falta de conocimiento o contexto específicos del dominio: los documentos sin procesar pueden carecer del conocimiento o el contexto necesarios para una generación precisa de un dominio específico. Esto puede limitar la capacidad de los modelos RAG para generar respuestas relevantes y precisas. Un ejemplo es un documento que hace referencia a conceptos especializados sin proporcionar un contexto. Esto podría dar lugar a respuestas que no son significativas en el dominio en cuestión.

Si bien esta lista no es exhaustiva, proporciona un punto de partida para que las empresas piensen en qué es lo que no funciona y por qué. Los documentos pueden tener uno o más de estos desafíos. La clave para optimizar una aplicación RAG es utilizar un conjunto de documentos que cumplan con las mejores prácticas de redacción que optimicen la recuperación.

Documentación: prácticas recomendadas para aplicaciones RAG

El desarrollo de una aplicación exitosa de generación aumentada por recuperación (RAG) requiere una consideración cuidadosa de varios factores relacionados con los documentos para optimizar su rendimiento. Las mejores prácticas de esta sección se han seleccionado en función de la experiencia en la creación de sistemas RAG con muchos líderes de la organización. Las siguientes son algunas de las mejores prácticas clave para que los documentos mejoren la eficacia de su aplicación de RAG:

- Utilice los encabezados y subtítulos correctamente: organizar el contenido con encabezados y subtítulos claros mejora la legibilidad y ayuda a los modelos RAG a comprender la estructura de los documentos. Esta práctica permite a los modelos navegar mejor y extraer información de los documentos, lo que mejora la calidad de las respuestas generadas.
- Asegúrese de que la numeración sea secuencial: cuando utilice listas numeradas, es importante mantener una numeración adecuada para evitar confusiones. Asegúrese de que cada elemento de la lista esté numerado secuencialmente sin omitir números. Esto ayuda a mantener la claridad y la coherencia del contenido.
- Agregue transiciones entre los elementos de una lista: proporcionar transiciones entre los elementos de una lista numerada o con viñetas ayuda a guiar al LLM a través del contenido. Por ejemplo, puedes usar frases como «Después de completar el paso 2, haz...» para conectar ideas y mejorar el flujo de información.
- Sustituir tablas: evite usar tablas. Formatee esta información en listas con viñetas de varios niveles o en una sintaxis de nivel plano. La sintaxis de nivel plano consiste en organizar elementos o elementos en el mismo nivel jerárquico, sin niveles anidados de subordinación. Estas estructuras ayudan a LLMs asimilar la información. Como la mayoría de los documentos indexados se leen de izquierda a derecha, la sintaxis de nivel plano permite que la información se muestre de manera más coherente sin necesidad de hacer referencia a una dimensión adicional. Este formato es más adecuado para las aplicaciones RAG porque presenta la información de una manera estructurada y fácil de digerir.
- Procese previamente la información gráfica para aumentar la eficiencia: Multimodal LLMs puede ingerir tanto imágenes como texto. Reduzca la resolución de las imágenes, elimine las imágenes redundantes y describa el contenido de los elementos gráficos en formato de texto. Estas

medidas mejoran el contexto significativo, evitan el consumo innecesario de fichas y mejoran la accesibilidad de los modelos RAG.

- Añada iniciadores de sesión para consultas habituales: cuando aborde preguntas o tareas habituales, como «¿Cómo solicito el software?» , añada un iniciador de sesión que haga que el lector participe en el proceso. Por ejemplo, puede añadir «Si quiere solicitar un software, siga los pasos que se indican a continuación...». Esto ayuda a crear una alta coincidencia semántica, lo que ayuda al LLM a construir una respuesta cohesiva.
- Agregue un resumen a cada sección: después de cada encabezado o subtítulo, agregue un resumen breve y conciso del contenido de esa sección. Esto puede aumentar la cobertura semántica y reforzar los puntos clave. Esto mejora la precisión de la búsqueda de similitudes en el espacio de incrustación, lo que mejora el rendimiento de la aplicación RAG. Esto resulta especialmente útil si el documento está destinado tanto a la enseñanza previa como al consumo humano o si se necesitan elementos gráficos y de tablas.
- Desambiguación: los documentos deben ser concisos y específicos. LLMs generen respuestas basadas en extractos recuperados, de modo que la desambiguación ayude al modelo a utilizar información clara y relevante. Esto da como resultado respuestas más precisas e informativas.
- Defina las abreviaturas y establezca el contexto: LLMs se entrenan con una gran cantidad de datos de Internet y, la mayoría de las veces, no tienen el contexto de los documentos internos de una empresa. Por lo tanto, establecer el contexto, definir abreviaturas y evitar o definir la terminología específica de la empresa ayuda al LLM a comprender los datos de la empresa. Esto ayuda al LLM a responder a las preguntas con mayor precisión y puede ayudar a prevenir las alucinaciones.
- Reestructura los documentos grandes en documentos más pequeños para etiquetarlos e indexarlos de manera eficiente: evita indexar un documento grande que contenga varios subtemas. Considere la posibilidad de dividir el documento grande en documentos más pequeños e independientes con títulos claros. Esto mejora la indexación y el etiquetado.

Preguntas frecuentes

¿Por qué es importante optimizar los documentos para las aplicaciones RAG?

Los documentos sin procesar suelen escribirse para el consumo humano sin tener en cuenta los requisitos de los sistemas de IA avanzados, como las aplicaciones de recuperación aumentada (RAG). La optimización de los documentos siguiendo las mejores prácticas puede mejorar significativamente el rendimiento y la precisión de las aplicaciones RAG al proporcionar información estructurada, inequívoca y relevante a los modelos.

¿Cuáles son algunos de los desafíos más comunes relacionados con los documentos sin procesar que pueden afectar el rendimiento del RAG?

Algunos desafíos clave incluyen la falta de formatos y metadatos estructurados, el lenguaje informal o incoherente, la verbosidad y la redundancia, los términos y frases ambiguos, la inclusión de elementos de hipervínculos y la falta de un contexto específico para el dominio. Estos problemas pueden confundir los modelos RAG y dar lugar a respuestas inexactas o irrelevantes. Para obtener más información, consulte los desafíos de los datos de origen que afectan a las aplicaciones de RAG en esta guía.

¿Cómo puede el uso de encabezados y subtítulos mejorar el rendimiento de los RAG?

Los encabezados y subtítulos claros ayudan a los modelos RAG a entender la estructura y el contexto del contenido. Esto les permite navegar mejor y extraer la información relevante de los documentos y mejora la calidad de las respuestas generadas. Para obtener más información, consulte la documentación sobre las prácticas recomendadas para las aplicaciones RAG en esta guía.

¿Por qué se recomienda sustituir la información de las tablas por una sintaxis de nivel plano?

Para los modelos RAG, interpretar las tablas puede resultar difícil porque requieren una comprensión de la estructura bidimensional. Presentar la información de las tablas en una sintaxis de nivel plano o en una lista con viñetas ayuda a los modelos a procesar la información con mayor facilidad, lo que se traduce en un mejor rendimiento. Para obtener más información, consulte la documentación sobre las prácticas recomendadas para las aplicaciones RAG en esta guía.

¿Cómo puede la adición de resúmenes mejorar el rendimiento de los RAG?

Incluir resúmenes concisos al principio de cada sección o subsección puede aumentar la cobertura semántica y reforzar los puntos clave. Esto mejora la precisión de las búsquedas de similitud dentro del espacio de incrustación, lo que, en última instancia, mejora el rendimiento de la aplicación RAG. Para obtener más información, consulte la documentación sobre las prácticas recomendadas para las aplicaciones RAG en esta guía.

¿Por qué es importante definir las abreviaturas y establecer el contexto? LLMs

LLMs están formados en una amplia gama de datos, pero carecen de contexto para utilizar abreviaturas o terminología específicas de cada empresa. Definir abreviaturas y proporcionar contexto ayuda a LLMs comprender y responder con mayor precisión. Esto puede ayudar a prevenir alucinaciones o interpretaciones erróneas. Para obtener más información, consulte la documentación sobre las prácticas recomendadas para las aplicaciones RAG en esta guía.

Próximos pasos y recursos

Esta guía analiza un conjunto de desafíos a nivel de documento para las aplicaciones RAG y las mejores prácticas para mitigarlos. Estos aprendizajes se obtienen a partir de entrevistas y debates con líderes del sector, y están respaldados por casos de uso empresarial.

Para empezar a optimizar sus documentos para las aplicaciones RAG, le recomendamos que lleve a cabo una auditoría de los documentos existentes. Identifique las áreas que plantean desafíos a la aplicación del RAG. Los ejemplos incluyen la falta de estructura, el lenguaje ambiguo o el uso excesivo de elementos gráficos. Priorice los documentos a los que accede con frecuencia o que son fundamentales para sus operaciones comerciales. Colabore con expertos en la materia para implementar las mejores prácticas de esta guía. Asegúrese de reestructurar los documentos con encabezados claros, un lenguaje conciso y elementos que definan el contexto. Para los documentos nuevos, establezca pautas y plantillas que garanticen la coherencia y ayuden a los autores a seguir las mejores prácticas. Además, considere la posibilidad de invertir en herramientas o servicios que puedan automatizar aspectos del proceso de optimización de los documentos, como el uso de la IA generativa para reestructurar los documentos. Al adoptar un enfoque proactivo para la optimización de los documentos, puede aprovechar todo el potencial de las aplicaciones RAG y obtener resultados más precisos y perspicaces en toda su organización.

Los siguientes recursos pueden ayudarle a comprender y crear aplicaciones de RAG en su organización.

Recursos

AWS documentación

- Elección de una base de datos AWS vectorial para los casos de uso de RAG (guía AWS prescriptiva)
- Implemente un caso de uso de RAG AWS mediante Terraform y Amazon Bedrock (AWS orientación prescriptiva)
- Desarrolle asistentes de IA generativos avanzados basados en el chat mediante el uso de RAG y el uso de indicaciones (orientación prescriptiva) ReAct AWS
- Recupere datos y genere respuestas de IA con las bases de conocimiento de Amazon Bedrock (documentación de Amazon Bedrock)

- Generación aumentada de recuperación (documentación de Amazon SageMaker AI)
- Acceda a las opciones y arquitecturas de generación aumentada AWS(orientación prescriptiva)AWS

Otros recursos AWS

- Cree un asistente multimodal con RAG avanzado y Amazon Bedrock (AWS entrada del blog)

Historial de documentos

En la siguiente tabla, se describen cambios significativos de esta guía. Si quiere recibir notificaciones de futuras actualizaciones, puede suscribirse a las notificaciones RSS.

Cambio	Descripción	Fecha
Publicación inicial	—	24 de julio de 2025

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.