撰寫最佳實務以最佳化 RAG 應用程式

AWS 方案指引

AWS 方案指引: 撰寫最佳實務以最佳化 RAG 應用程式

Table of Contents

撰寫最佳實務以最佳化 RAG 應用程式

Ivan Cui 和 Samantha Stuart，Amazon Web Services

2025 年 7 月 (文件歷史記錄)

大型語言模型 LLMs) 已徹底改變人工智慧領域，並具備絕佳的能力來了解和產生類似人類的文字。不過，他們面臨重大限制：他們只能使用訓練資料中包含的知識。這是 Retrieval Augmented Generation (RAG) 提供協助之處。它提供結合 LLMs與外部知識來源的解決方案，例如您組織的資料和文件。透過涉及資訊擷取和回應產生的兩階段程序，RAG 可讓 AI 系統存取和整合來自各種來源up-to-date，從而產生更準確和明智的回應，從而消除靜態模型知識和動態真實世界資訊需求之間的差距。

如何最佳化 RAG 型應用程式中擷取的內容？ 本指南提供最佳實務，協助您最佳化知識庫中文字型內容的格式和撰寫樣式。最佳化內容可增強內容，協助 RAG 應用程式更準確地了解任務特定資訊。當系統擷取高度相關且準確的內容時，LLM 回應的品質就會改善。在系統層級最佳化內容交付程序稱為內容工程，它構成代理 RAG 架構的重要部分。在代理程式 RAG 中，一或多個額外的 LLMs原因，並在 RAG 執行之前處理接收請求。這有助於多步驟資訊交付程序。隨著 RAG 架構越來越複雜，來源內容最佳化仍然是為 LLMs提供清晰內容的最直接方式。這些最佳實務旨在協助您將組織對 RAG 應用程式的投資最大化。

目標對象

本指南適用於使用一或多個 RAG 元件建置 LLM 應用程式的 AI 工程師、資料科學家、資料工程師或軟體開發人員。若要了解本指南中的概念和建議，您應該熟悉向量資料庫和 LLMs 的提示。

目標

本指南中的建議可協助您達成下列目標：

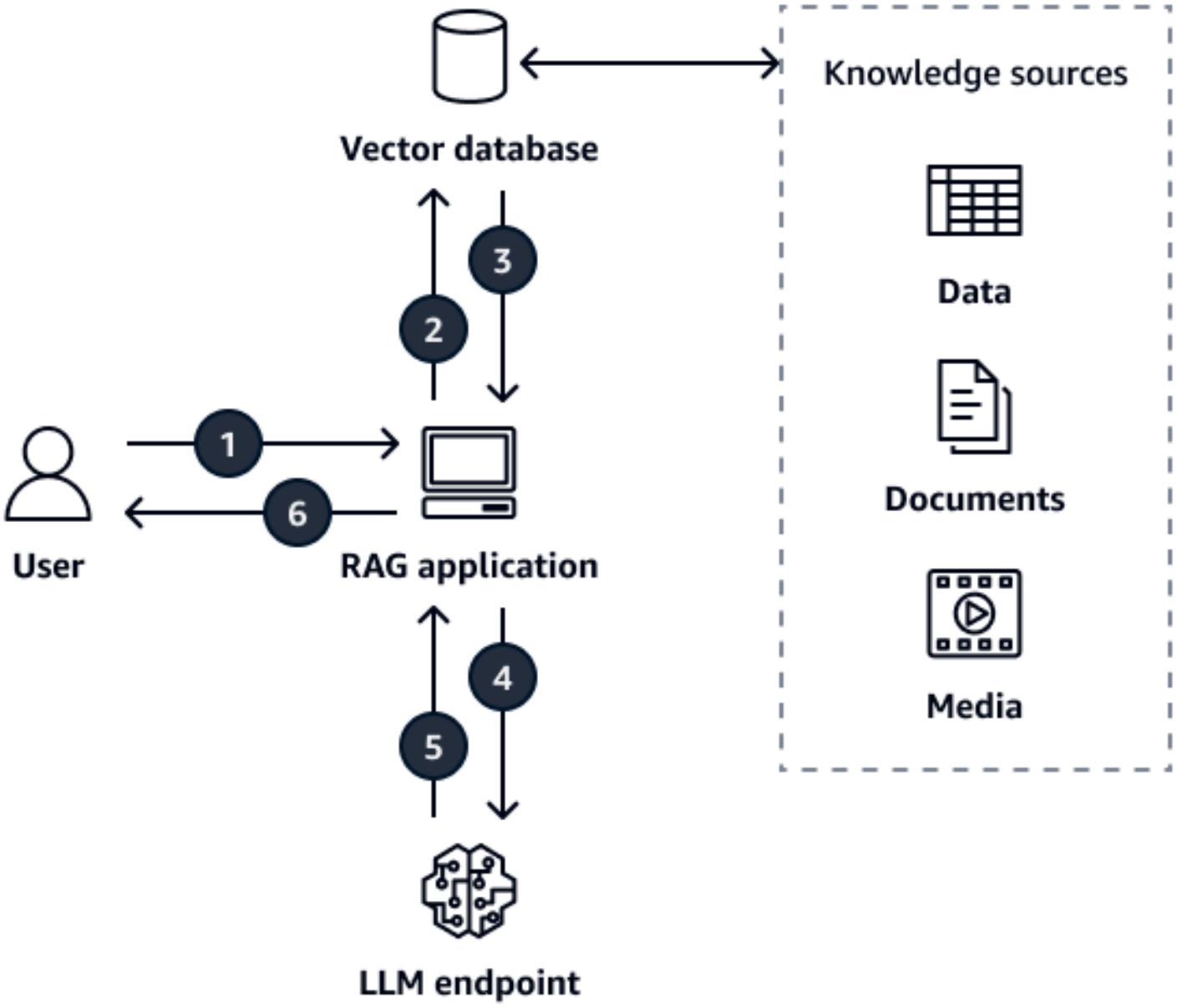
- 提供結構完善且語意豐富的來源文件，針對字符用量和備援進行最佳化，以改善 RAG 應用程式產生的回應的準確性和相關性。
- 透過在來源文件中提供明確的定義和說明，協助 RAG 應用程式更深入了解特定領域的知識和內容。
- 透過遵循跨來源文件的一致格式和結構準則，簡化 RAG 應用程式的維護和知識庫更新。
- 將大型單體文件分解為可有效編製索引和擷取的小型獨立單位，以改善 RAG 解決方案的可擴展性。

了解 LLMs和 RAG

若要了解增強來源文件品質如何增強 RAG 回應的品質，您必須了解 LLM 的內部工作。LLMs 的真正功能在於能夠使用自我關注機制和轉換器架構。這些進階技術可讓模型有效地處理和關聯輸入序列的不同部分，無論其在文字中的位置或距離為何。此功能與傳統語言模型形成強烈的對比，通常難以理解長期相依性和內容。此外，LLMs是以前所未有的規模進行訓練。有些最大的模型是由數兆個參數組成，並且從各種來源擷取了 TB 的文字資料。此大規模擴展可讓 LLMs 深入了解語言，擷取先前對 AI 系統具有挑戰性的細微細微差別、同義詞和情境提示。結果是一類模型，可以產生一致且流暢的文字，並在問題回答、文字摘要甚至程式碼產生等任務中展現顯著的功能。

若要使用這些模型，我們可以使用 Amazon Bedrock 等服務，這些服務可讓您從 Amazon 和第三方供應商存取各種基礎模型，包括 Anthropic、Cohere 和 Meta。您可以使用 Amazon Bedrock 來實驗 state-of-the-art模型、自訂和微調模型，或透過單一 API 將這些模型納入生成式 AI 解決方案。

雖然 LLMs擅長擷取模式和產生連貫文字，但通常無法存取up-to-date或專門的資訊。RAG 結合了 LLMs 的生成能力和擷取元件，該元件可以存取和整合來自外部來源的相關資訊，作為具體化 LLM 提示的一部分。外部來源的範例包括 Amazon Bedrock 的知識庫、Amazon Kendra 等智慧型搜尋系統，或 Amazon OpenSearch Service 等向量資料庫。



圖表說明下列工作流程：

1. 使用者向 RAG 應用程式提交查詢。
2. RAG 應用程式會查詢包含知識來源的向量資料庫，例如文件、資料或媒體。
3. RAG 應用程式會根據查詢與存放文件之間的語意相似性，從向量資料庫擷取相關資訊。
4. RAG 應用程式會使用擷取的內容增強原始提示，並將其傳送至 LLM 端點。
5. LLM 端點會產生回應並將其傳回至 RAG 應用程式。
6. RAG 應用程式會將產生的回應傳回給使用者。

RAG 的核心採用兩階段程序。在第一個階段中，擷取模型會根據輸入查詢識別和擷取相關文件或段落。此擷取模型可以是傳統的資訊擷取系統、密集擷取模型或兩者的組合。在第二個階段中，擷取的資訊和原始查詢會以完全具體化的提示範本的形式饋送至 LLM。LLMs很大程度上取決於擷取器元件交付的來源內容品質。它們會套用自我注意力機制，以數學方式編碼擷取的內容與任務的關係。LLM 接著會根據查詢和擷取的資訊產生回應。在 RAG 中，控制擷取來源文件的品質是改善 LLM 任務內部表示方式的直接方法。RAG 使用相關的外部資料有效地增強 LLM 的訓練資料。這種方法可讓 RAG 利用 LLMs 和擷取系統的優勢，產生更準確且明智的回應，其中包含目前和專業知識。

向量和內嵌

向量和內嵌是機器學習和自然語言處理的基礎概念。向量是數學物件，代表同時具有大小和方向的數量。在自然語言處理 (NLP) 的情況下，單字、句子或文件通常以立體向量空間中的向量表示。另一方面，嵌入是一種在低維度向量空間中代表字詞或文件等物件的方式，其中向量之間的關係會擷取語意或語法相似性。例如，字詞內嵌可讓具有類似意義的字詞具有類似的向量表示法。這有助於演算法更有效地了解和處理語言。

向量資料庫

在生成式 AI 中，向量資料庫是存放和管理文件、查詢或其他物件的向量表示法的資料庫。它旨在有效地存放和擷取向量。這支援快速且可擴展的操作，例如語意搜尋和相似性比對。向量資料庫使用特殊資料結構來編製向量索引，例如階層式導航小型世界 (HNSW) 圖形或 K 近鄰 (KNN) 演算法。這些資料結構允許快速最近鄰搜尋，因此可以在資料庫中快速找到類似的向量。

語意搜尋

語意搜尋是一種技術，可透過了解查詢的意圖和內容來改善搜尋結果的相關性，而不只是相符的關鍵字。在技術術語中，語意搜尋涉及比較查詢的向量表示法和資料庫中的文件，以尋找最相關的相符項目。不同的擷取策略可用於語意搜尋，包括但不限於：

- HNSW – 以圖形為基礎的資料結構，可整理向量，讓搜尋最接近的鄰近區域更有效率。
- KNN – 根據距離指標尋找最接近查詢向量的 K 向量的演算法，例如餘弦相似性。
- 餘弦相似性 – 兩個非零向量之間的相似性測量，可測量它們之間角度的餘弦。它通常用於語意搜尋，以比較高維度空間中的向量方向。
- 位置敏感雜湊 (LSH) – 一種將類似向量雜湊至具有高機率之相同或鄰近儲存貯體的技術。這允許近似最近鄰搜尋，這可以比在高維度空間中進行精確搜尋更快。

來源資料中影響 RAG 應用程式的挑戰

開發最佳擷取增強生成 (RAG) 應用程式的主要挑戰之一在於所使用的原始資料或文件。通常，企業會使用為人工參考而建立的現有文件。這些文件通常包含超連結和影像螢幕擷取畫面，以促進理解。不過，由於擷錄權杖限制，這些元素會阻礙語意擷取。這會導致擷取器效能不佳。

以下是最佳 RAG 應用程式最常見的原始文件挑戰：

- 缺乏結構化格式和中繼資料 – 原始文件可能缺少明確的區段標題、子標題或中繼資料。這使得識別和擷取相關資訊變得具有挑戰性。例如，沒有明確標題的長文件可能會讓您難以判斷特定資訊的內容。
- 非正式和不一致的語言 – 原始文件通常包含非正式語言或不一致的術語。這可能會混淆 RAG 模型。例如，未於文件中定義或 LLM 已知的縮寫可能在整個文件中使用。
- 動詞和備援 – 原始文件可能是詳細的，並包含不必要的或備援資訊。這可能會壓倒 RAG 模型，導致較不簡潔和相關的回應。範例包括多次重複相同資訊的文件，或包含類似或矛盾資訊的多個文件。
- 模棱兩可的術語和片語 – 原始文件可能包含模棱兩可的術語或片語，可能以多種方式解釋。這種模棱兩可性可能會導致 RAG 模型的錯誤解譯和不正確的回應。例如，使用具有多個意義的字詞的文件可能會導致回應不符合預期的意義。
- 注入圖形和超連結元素 – 包含圖形和超連結資訊的原始文件非常適合人類使用。不過，這些元素可能會使用擷取字符限制。結果是擷錄可能不完整。例如，圖形和超連結 URLs會傳回為擷取的一部分，這會使用擷取字符，而後續段落的金鑰資訊會遺失。
- 缺乏特定網域的知識或內容 – 原始文件可能缺乏準確產生所需的特定網域知識或內容。這可能會限制 RAG 模型產生相關且準確回應的能力。範例是參考特殊概念而不提供內容的文件。這可能會導致在指定網域中沒有意義的回應。

雖然此清單並不全面，但它為企業提供了起點，以考慮什麼無效和原因。文件可能有一個或多個這些挑戰。最佳化 RAG 應用程式的關鍵是使用一組文件，以遵循撰寫最佳化擷取的最佳實務。

RAG 應用程式的文件最佳實務

開發成功的擷取增強生成 (RAG) 應用程式需要仔細考慮各種文件相關因素，以最佳化其效能。本節中的最佳實務是根據與許多組織領導者一起建置 RAG 系統的經驗所策劃。以下是文件的幾個關鍵最佳實務，以增強 RAG 應用程式的有效性：

- 正確使用標題和子標題 – 整理具有清晰標題和子標題的內容可提高可讀性，並協助 RAG 模型了解文件的結構。此實務可讓模型更好地從文件中導覽和擷取資訊，從而增強產生的回應品質。
- 確保編號是循序的 – 使用編號清單時，請務必維持適當的編號以避免混淆。確保每個清單項目都按順序編號，而不會略過數字。這有助於維持內容的清晰度和一致性。
- 在清單項目之間新增轉換 – 在項目符號或編號清單中的項目之間提供轉換，有助於引導 LLM 完成內容。例如，您可以使用「在完成步驟 2 後，執行...」之類的片語來連接想法並改善資訊流程。
- 取代資料表 – 避免使用資料表。在多層項目符號清單或平面語法中格式化此資訊。平面語法是在相同的階層層級配置元素或項目，而沒有巢狀層級的次排序。這些結構可協助 LLMs 摘要資訊。由於大多數索引文件是從左到右讀取，因此平面語法可讓資訊更一致地遵循，而不需要參考額外的維度。這種格式對 RAG 應用程式更有利，因為它以結構化且易於理解的方式呈現資訊。
- 提高效率的預先處理圖形資訊 – 多模態 LLMs 可以同時擷取影像和文字。降低影像解析度、移除備援影像，並以文字格式描述圖形元素的內容。這些措施可改善有意義的內容、避免不必要的使用字符，並改善 RAG 模型的可存取性。
- 新增常見查詢的工作階段啟動者 – 解決常見問題或任務時，例如「如何訂購軟體？」，請新增工作階段啟動者，將讀取者轉換為 程序。例如，您可以新增「如果您想要訂購軟體，請遵循以下步驟...」。這有助於建立高語意比對，這有助於 LLM 建構凝聚性回應。
- 將摘要新增至每個區段 – 在每個標題或子標題 之後，新增該區段內容的簡短且簡潔摘要。這可以增加語意涵蓋範圍並強化關鍵點。這可改善內嵌空間內相似性搜尋的準確性，進而改善 RAG 應用程式的效能。如果文件同時適用於 LLM 和人工取用，或需要資料表和圖形元素，這特別有用。
- 歧義 – 文件應該簡潔且集中。LLMs 會根據擷取的摘錄產生回應，因此歧義可協助模型使用清晰的相關資訊。這會產生更準確且資訊豐富的回應。
- 定義縮寫和設定內容 – LLMs會根據大量網際網路資料進行訓練，而且大多數情況下，它們沒有企業內部文件的內容。因此，設定內容、定義縮寫，以及避免或定義公司特定的術語，有助於 LLM 了解您的企業資料。這有助於 LLM 更準確地回答問題，並有助於防止幻覺。
- 將大型文件重組為較小的文件，以有效率地標記和編製索引 – 避免將包含多個子主題的大型文件編製索引。考慮將大型文件分割成具有明確標題的小型獨立文件。這可改善索引和標記。

常見問答集

為什麼最佳化 RAG 應用程式的文件很重要？

原始文件通常寫入供人類使用，而不考慮進階 AI 系統的需求，例如擷取增強生成 (RAG) 應用程式。遵循最佳實務來最佳化文件，可透過為模型提供結構化、明確且相關的資訊，大幅提升 RAG 應用程式的效能和準確性。

原始文件可能會阻礙 RAG 效能的常見挑戰有哪些？

一些關鍵挑戰包括缺乏結構化格式和中繼資料、非正式或不一致的語言、語彙和備援、模棱兩可的詞彙和片語、包含超連結元素，以及缺乏特定網域的內容。這些問題可能會混淆 RAG 模型，並導致不正確或不相關的回應。如需詳細資訊，請參閱本指南中影響 RAG 應用程式來源資料的挑戰。

使用標題和子標題如何改善 RAG 效能？

清楚的標題和子標題可協助 RAG 模型了解內容的結構和內容。這使他們能夠更好地從文件中導覽和擷取相關資訊，並改善產生的回應品質。如需詳細資訊，請參閱本指南中的 RAG 應用程式的文件最佳實務。

為什麼建議使用平面語法取代資料表資訊？

RAG 模型解譯資料表可能具有挑戰性，因為它們需要了解二維結構。在平面語法或項目符號清單中呈現資料表資訊，有助於模型更輕鬆地處理資訊，進而獲得更好的效能。如需詳細資訊，請參閱本指南中的 RAG 應用程式的文件最佳實務。

新增摘要如何增強 RAG 效能？

在每個區段或子區段的開頭包含簡潔摘要，可以增加語意涵蓋範圍並強化關鍵點。這可提高內嵌空間內相似性搜尋的準確性，最終增強 RAG 應用程式的效能。如需詳細資訊，請參閱本指南中的 RAG 應用程式的文件最佳實務。

為什麼定義 LLMs 的縮寫和設定內容很重要？

LLMs會根據廣泛的資料進行訓練，但缺乏企業特定縮寫或術語的內容。定義縮寫並提供內容有助於 LLMs 更準確地了解和回應。這有助於防止幻覺或誤解。如需詳細資訊，請參閱本指南中的 RAG 應用程式的文件最佳實務。

後續步驟和資源

本指南討論 RAG 應用程式的文件層級挑戰，以及緩解這些挑戰的最佳實務。這些學習是透過與產業領導者進行採訪和討論來策劃的，並由企業使用案例提供支援。

若要開始最佳化 RAG 應用程式的文件，建議您對現有文件進行稽核。識別對 RAG 應用程式構成挑戰的區域。範例包括缺乏結構、語言不明確或過度使用圖形元素。優先處理經常存取或對業務營運至關重要的文件。與主題專家合作，實作本指南中的最佳實務。確保使用清晰的標題、簡潔的語言和內容設定元素來重組文件。對於新文件，請建立指導方針和範本，以確保一致性並協助作者遵守最佳實務。此外，請考慮投資可以自動化文件最佳化程序層面的工具或服務，例如使用生成式 AI 來重組文件。透過採取主動的方法進行文件最佳化，您可以釋放 RAG 應用程式的完整潛力，並在整個組織中推動更準確和更深入的結果。

下列資源可協助您了解並建置組織中的 RAG 應用程式。

資源

AWS 文件

- 選擇 RAG 使用案例的 AWS 向量資料庫 (AWS 規範性指導）
- AWS 使用 Terraform 和 Amazon Bedrock 在 上部署 RAG 使用案例 (AWS 方案指引）
- 使用 RAG 和 ReAct 提示來開發進階生成式 AI 聊天式助理 (AWS 方案指引）
- 使用 Amazon Bedrock 知識庫擷取資料並產生 AI 回應 (Amazon Bedrock 文件）
- 擷取增強生成 (Amazon SageMaker AI 文件）
- 在 上擷取增強生成選項和架構 AWS(AWS 方案指引）

其他 AWS 資源

- 建立具有進階 RAG 和 Amazon Bedrock 的多模式助理 (AWS 部落格文章）

文件歷史紀錄

下表描述了本指南的重大變更。如果您想收到有關未來更新的通知，可以訂閱 RSS 摘要。

變更	描述	日期
初次出版	—	2025 年 7 月 24 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。