perplexity

please list the architectures of RAG applications

please list the architectures of RAG applications

以下是RAG（Retrieval-Augmented Generation）應用程序的八種主要架構：

1. **簡單RAG**：
   - 這是最基本的RAG形式。用戶提供一個提示或問題，模型從一個固定的數據庫中搜索相關信息，然後基於這些信息生成一個響應。這種實現適用於數據庫相對較小且不需要複雜處理大規模或動態數據集的情況[1]。
2. **簡單RAG與記憶**：
   - 這種架構引入了存儲組件，允許模型保留之前交互的信息。這對於連續對話或需要跨多個查詢的上下文感知任務特別有用[1]。
3. **分支RAG**：
   - 分支RAG通過根據輸入確定應查詢哪些特定數據源來提供更靈活和高效的數據檢索方式。它評估查詢並選擇最相關的源進行信息檢索[1]。
4. **HyDe（假設文檔嵌入）**：
   - HyDe是一種獨特的RAG變體，它在檢索相關信息之前生成基於查詢的假設文檔。這種方法通過使用假設文檔來指導檢索，提高了結果的相關性和質量[1]。
5. **自適應RAG**：
   - 自適應RAG根據查詢的複雜性和上下文動態調整檢索策略。它可以根據查詢的難度或上下文的變化來選擇不同的檢索方法[1]。
6. **修正RAG（CRAG）**：
   - CRAG通過在生成過程中引入修正機制來提高生成的準確性和相關性。它可以根據用戶反饋或其他信號來調整生成的內容[1]。
7. **自主RAG**：
   - 自主RAG允許模型自主決定如何處理查詢，包括選擇檢索策略、生成策略等。這種架構適用於需要高度自主性的應用場景[1]。
8. **代理RAG**：
   - 代理RAG利用代理來管理和協調RAG過程中的各個步驟。它可以根據任務的需求來選擇和調整不同的RAG模塊[1]。

這些架構各有其特點和適用場景，選擇哪種架構取決於應用程序的具體需求，如數據庫的大小、查詢的複雜性、對於上下文感知的要求等。

⁂

1. https://humanloop.com/blog/rag-architectures