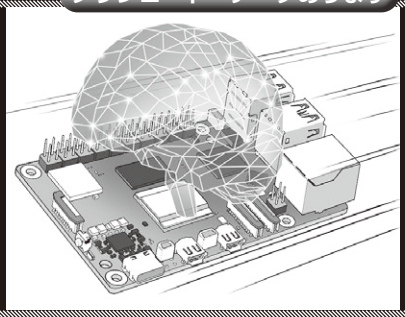
ダウンロード・データあります

リソースの限られた環境での実行性能を徹底解析！

ラズパイの限界に挑戦！ ローカルLLM 動作検証レポート



第3回 RAGを作る②…ベクトル検索ライブラリの構築

ご購入はこちら

澁谷 慎太郎

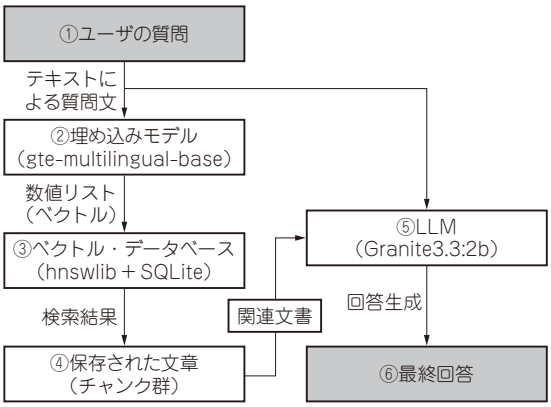


図1 作成するRAGの質問から最終回答までのフロー

● 今回のテーマ

動作検証の1つとして，前回からラズベリー・パイ5にローカルでRAGを構築しています．作成するRAGの処理フローを**図1**に示します．前回は一通り，RAGで使用する外部データをウェブ記事からhtmlやpdfの形で収集し，それらをマークダウン化するところまでを解説しました．

今回は，マークダウン化したデータを整え，適切なサイズのチャンクにして，リレーショナル・データベース（以降，RDB）に保管しつつ，ベクトル・インデックスを作り，ベクトル検索ライブラリに格納するところまでの一連の作業を行ってみます．

● 実行環境

前回から引き続き同じハードウェアを使用します（**表1**）．あくまで推奨構成なので，これと同じである必要はありません．実行環境構築の詳細は，本連載サポート・ページ，または本誌2025年12月号 pp.116-117を確認ください．

● 作成するRAGの構成

RAGの各処理（**図1**）を順に説明します．使用するソフトウェアを**表2**に示します．

表1 使用するハードウェア

スペック	内　容
機種	Raspberry Pi 5
メイン・メモリ	8Gバイト
ストレージ	Raspberry Pi SSD Kit 512Gバイト
冷却	Raspberry Pi 5用公式アクティブクーラ

①ユーザの質問

②埋め込みモデル
質問文を埋め込みモデル（今回はgte-multilingualbaseを使用）に通し，文章を数値ベクトルに変換します．

③ベクトル・インデックス（hnswlib）
埋め込みモデルによって，ベクトルに変換されたユーザの質問を使い，hnswlibに保存されている文書チャンク群（文章をチャンク化したもの．チャンク（chunk）化とは，1件の文書を意味が途切れない単位で分割した検索用の最小ブロックに分割すること）のベクトルと比較します．

④保存された文書（SQLiteにあるチャンク群）
③の検索結果で得られた文書IDをキーにしてSQLiteから関連度の高いテキスト（文字列）が取り出されます．

⑤LLM（Granite 3.3：2B，Ollama）
LLMが質問と③④で取り出された関連文書を同時に参照しながら回答を生成します

⑥最終回答
LLMが生成した回答がユーザに提示されます．LLMが学習していない事柄も回答してくれるようになります．

今回行う作業…マークダウン化された情報の整理からRAGの検索ライブラリ完成まで

図2に作業フローを示します．前回，マークダウン形式への変換まで行いました．今回は，マークダウン化された情報を整理整頓し，その情報のチャンク化から，ベクトル化，ベクトル・インデックス作成し，ベ