第2章 情報源の指定/インデックス化/検索/生成/チェーンの作成と実行

小説「注文の多い料理店」でRAGとLLMを比べる

山田 薫

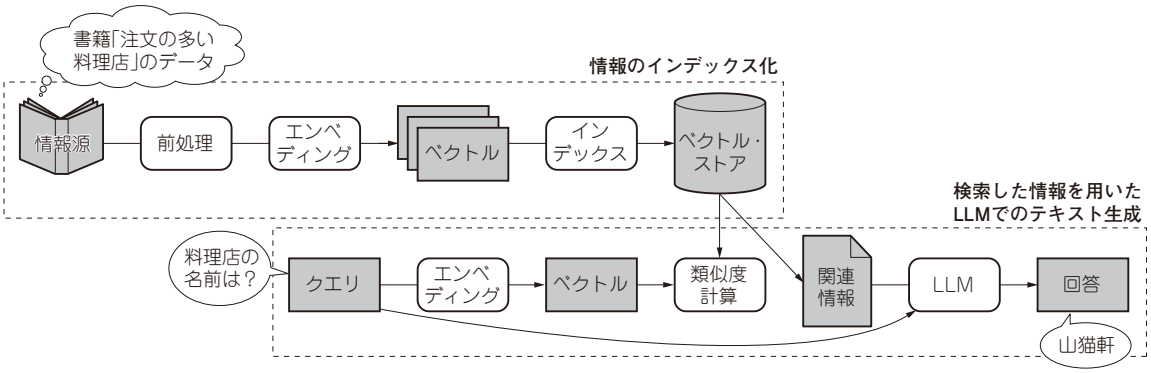

図1　本稿で作成するRAG
小説「注文の多い料理店」のテキストを検索情報とする．料理店の名前などを質問して，正しい回答が生成されるか試す

本稿では実際にRAGを作成し動かしてみます．今回は，小説「注文の多い料理店」(宮沢 賢治 著)についての質問に答えるRAGを作り(**図1**)，LLM単独で生成した答えとも比較します．(編集部)

開発環境

サンプル・コードは言語としてPythonを使用しており，Google Colaboratory[1]上で動作するように作成しています．Google Colaboratoryは機械学習，データ・サイエンス，教育向けのサービスで，Jupyter Notebookというインタラクティブにコードを実行できる環境を使用できます．コンピューティング・リソースはクラウド上のものを使用し，リソース制限はありますが無料でGPUなどを利用できます．**図1**の通り，今回はクラウド環境での実行を解説していますが，サンプル・コード自体はPCなどのローカル環境で実行する場合でも変わりません(**図2**)．そのため，メモリやVRAMに余裕があればPCでも実行可能です．またサンプル・データは青空文庫[2]のデータを使用します．ソフトウエア環境を**表1**に示します．

● フレームワークはLangChain

RAGを使ったアプリケーションを効率的に実装することができるライブラリを紹介します．

現在主流なのは，LangChain(ラングチェーン)とLlamaIndex(ラマインデックス)です．

LangChainはハリソン・チェイスによって立ち上げられたオープンソース・プロジェクトで，LLMを活用したアプリケーション開発のための汎用的なフレームワークです．LlamaIndexはカスタム・データ・ソースとLLMを接続するためのフレームワークでよりカスタム・データ活用に特化した位置付けとなっています．

どちらもモデルやベクトルストアなどのサードパーティ製品との連携が可能となるコードを提供しているので，スムーズに連携することが可能です．残念ながらどちらも英語のドキュメントのみとなります．英語ではあるものの，どちらもドキュメントが充実しているため，精度改善に役立ちそうなさまざまな手法を探したり，APIリファレンスを参照したりすることができます．

今回のサンプル・コードは，RAGを使用しないLLM単独の実行も試しているため，汎用的な使い方が可能なLangChainを採用しています．

● HuggingFaceライブラリ

今回のサンプル・コードでは，LLMをローカルに効率的にダウンロードするために，HuggingFaceの