第1章 実装しやすく効果も高いRAGやプロンプト・エンジニアリングに注目が集まる

ご購入はこちら

ローカルLLM補完技術の導入ガイド

氏森 充

表1 ローカルLLMの課題解消に役立つ解決策(補完技術)と特徴

技術名	対応する主な課題	主な技術的特徴，概要	主な用途・目的
プロンプト・エンジニアリング	運用，保守の負担/モデル性能の制約	単発プロンプトの構造を工夫して精度や安定性を向上	推論，分類，簡易対話
RAG(検索拡張生成)	知識ベースの鮮度と更新性/拡張性の限界	外部知識の参照．ベクトル検索などを通じて文脈を補強．再学習不要	ナレッジ活用，FAQ，社内検索
コンテキスト・エンジニアリング	運用，保守の負担/モデル性能の制約	会話，文脈の構成や履歴管理，外部連携，情報優先度の管理などを含む	継続対話，意思決定支援
エージェント(Agent)	運用，保守の負担/拡張性の限界	外部ツールを呼び出してタスク実行(例：ブラウザ操作，API連携)	タスク自動化・会話型UI
ファインチューニング(Fine Tuning)	知識ベースの更新/モデル性能の制約	特定の業務や文体に応じた追加学習	専門特化モデル
LoRA/PEFT	モデル性能の制約/運用，保守の負担	一部パラメータのみを微調整する効率的なファインチューニング	リソース制約下の適応
階層/構造化RAG[注1]	知識ベースの鮮度と更新性/拡張性の限界	長文の論理構造を保持したまま検索・生成を行うRAG	報告書・設計書・法務文書対応
量子化/蒸留/枝刈り	モデル・サイズ，性能制約/拡張性の限界	精度を維持しながらモデル・サイズや計算量を削減し軽量化，高速化を実現	ローカル実装，組み込み用途

注1：HiRAG，RAPTOR，MC-indexing，ALoFTRAG

近年，トランスフォーマ・アーキテクチャを基盤とした大規模言語モデル(LLM)は飛躍的な進化を遂げ，業務効率化や自然言語インターフェースの構築など，多様な分野で急速に普及しています．当初はクラウドLLM[注1]が主流でしたが，PCや小型デバイスで動作するローカルLLMが登場し，クラウドLLMに匹敵する性能のものもあります．

ローカルLLMはインターネット接続を必要とせず，全ての処理をローカルで完結できるため，通信遅延やクラウド側の障害，ネットワークの不安定さ，外部への情報送信によるセキュリティ・リスクといった，クラウドLLMで生じやすい課題を回避できます．こうした特性が，近年ローカルLLMの採用が広がる大きな理由となっています．

一方で，ローカルLLMにはクラウドLLMとは異なる導入，運用上の課題も存在し，それらを克服するための工夫が不可欠です．

そこでローカルLLMを実用レベルで活用するために押さえておきたいポイントや代表的な課題を整理し，具体的な対処法(**表1**)を交えて解説します．

注1：本稿では，クラウド上で提供される大規模言語モデルをクラウドLLM，PCなどのローカル・デバイス上で動作するモデルをローカルLLMと定義します．

ローカルLLMが注目される理由

● 理由①…クラウドLLMの構造的課題を解決できる

▶ローカルLLMに注目される要因となった課題

クラウドLLMが広く使われる一方で，その運用上の課題がローカルLLMへの注目を加速させる要因となってきました．普及の背景となった4つの主要課題を説明します．

(1) データ秘匿性・プライバシー保護への要求

クラウドLLMは，入力データを外部サーバに送信して処理する必要があるため，機密情報や個人情報の漏えいリスクが常に指摘されてきました．

ローカルLLMは全ての処理を手元のデバイス内で完結できるため，外部送信を伴わず，プライバシ保護やコンプライアンス面で有力な選択肢となります．

(2) コスト構造の最適化ニーズ

クラウドLLMはAPIの呼び出しごとに料金が発生する従量課金型が多く，大規模運用では継続的なコス