



実践編

環境を整えたIT担当者のもとには、営業部から「社内で使う文書の型を揃えたい」、在庫管理部門から「最新の在庫をすぐ知りたい」、調査担当から「社内文書とWebを横断して調べたい」という相談が届いた。さてどうすべきか?その解決策には「型を持たせるSkill」、「生きたデータに触れるMCP」、「調べ物を任せるWeb検索とRAG」の3つがある。これらのプランを1台のRadeon AI PRO R9700搭載マシンで順に実践してみよう。

1台のローカルLLMを3つの現場で活用する

RAGだけでは解決しない3つの相談

例示した3つの相談は、社内文書を検索して答えるRAGだけでは解決できない。提案書や稟議書には自社の「型」があり、在庫数は刻々と変化していく。最新の業界動向を調べるにはWebにも目を向ける必要がある。そこで、すでに構築した基本のローカルLLMサーバーに「型を持たせる」、「生きたデータに触らせる」、「探す範囲を広げる」機能を加えていく。

社員が自然な日本語で話しかける点は変わらない。その裏側でAIがテンプレートを読み込み、DBへSQLを発行し、Webや社内文書を検索する。これを実現するのがSkill、MCP (Model Context Protocol)、Web検索だ。まず、RAG、Skill、MCPの違いを整理しておこう (表1参照)。

右ページの図3は、すでに構築済みのLLMサーバー環境を土台に追加する3つの機能だ。本書では、これらの機能を活用した以下の実践例を解説していく。

・営業=Skill

→総務・人事・営業事務が担当可能。数時間で一通り試せる

・在庫管理=MCP

→情報システム部門が主導。導入目安は数日～1週間程度

・リサーチ=Web検索+RAG

→管理者がAPIキーを設定するだけ。数分で導入可能



文書の「型」が人に付いて回る

過去の提案書を探し、書き方を先輩に聞いて回る。文書の「型」が共有されていない現場の日常だ

営業

提案書・稟議書の「型」をSkillで揃える

今回の例では、Open WebUIの「Skill」という機能を使って文書フォーマットの標準化を試すことにした。この設計思想は、Anthropicが提唱するAgent Skillsとほぼ同じである。Open WebUI独自の実装だが、狙いは共通しており「モデルに大量のルールを常時持たせず、必要な分だけ読ませる」という考え方に基づいている。

議事録・稟議書・提案書・メールは、内容そのものより「型」が重要な文書だ。会社ごとに書き方のルールがあり、新入社員や中途入社者はまずこの型を覚えるところから始める。ここをSkillとして用意しておけば、モデルに「新規のお客様向けに提案書を書いて」、「出張申請の稟議書を書いて」と頼む

表1 RAG・Skill・MCPの違い

	RAG	Skill	MCP
何を持たせるか	検索対象のドキュメント (社内文書や Web ページ)	再利用可能な指示書・型	外部処理を呼び出す手段
典型的な用途	規約・マニュアルの検索、Web 検索	議事録・稟議書等のフォーマット統一	DB・API へのライブアクセス
実行されるか	検索のみ (テキストを返す)	指示書を読み込むだけ (実行はしない)	実際に SQL 等を実行する
実装の重さ	中 (ベクトル DB 等が必要)	軽い (UI でテキスト入力するだけ)	重い (サーバー・権限設計が必要)

表2 登録したSkillのカテゴリと内容例

カテゴリ	内容例
議事録テンプレート	定例会議 / プロジェクトキックオフ / 顧客打ち合わせ / 品質会議 / 経営会議 / 採用面接 / 障害対応会議 / 予算検討会議 / ベンダー打ち合わせ / 1on1 面談
稟議書テンプレート	備品購入 / 出張申請 / システム導入 / 外部委託契約 / 採用稟議 / 交際費 / 研修費用 / 設備投資 / 契約更新 / 広告出稿
メール文体ガイド	初回問い合わせ返信 / 謝罪 / 見積もり送付 / 進捗報告 / 依頼 / 督促 / お礼 / アポイント調整 / クレーム対応 / 契約後フォロー
提案書テンプレート	新規提案 / 価格・見積提示 / 競合比較訴求 / 導入事例引用 / フォローアップ提案