中小企業での生成 AI の活用に向けた基礎的検討
― ローカル環境上で動作する AI チャットボットの試作 ―

千家　雅之（情報・生産技術部 システム技術グループ）

1.　はじめに

近年，生成 AI の進化は日進月歩であり，それに伴って業務での活用が急速に広まっている．日本企業においても，情報検索，文章作成，文章要約，プログラミングなどの領域で活用されていることが報告されている[1]．生成 AI の利用は，大規模な計算資源が必要なことから，大手 IT ベンダーが用意する生成 AI サービスにアクセスして利用する形態が主であり，このようなサービスは社外へのデータ送信を伴うため，情報セキュリティの観点から導入に慎重な企業も少なくない．そのため，ローカル環境で動作する生成 AI にも多くの関心が集まっており，限られた計算資源の中で性能を向上させるための研究開発が活発に進められている．特に，文章生成等の高度な言語タスクの処理を可能にするモデルである大規模言語モデル（LLM; Large Language Model）は積極的に更新されており，ローカル環境でも実用的な性能になりつつある．しかし，日本語向けのローカル環境で動作する LLM（以降，ローカル LLM）においては，LLM 関連のソフトウェアが英語圏を中心に開発されているということもあり，その性能とそれを用いて構築する LLM アプリケーションの実用性との関係については，今なお十分に解明されているとは言いがたい側面もある．

本稿では，中小企業におけるローカル LLM の実用可能性について考察するために，一例として社内文書に対する質問応答を行う AI チャットボットをローカル環境で構築して評価する．以降では，限られた計算資源で実装する手法を紹介した後，試作したチャットボットのシステム構成とその評価について述べ，最後に総括する．

2.　限られた計算資源で実装する手法

2.1　低ビット量子化

LLM は，大量のテキストデータをもとに事前学習された生成モデルであり，文章生成では，学習した文章の中から前後の文脈を反映した適切な表現を予測し，文章を出力する．人間のような自然な会話をするためには，LLM の規模（パラメータ数）が重要であり，規模が大きいほど人間に近い会話が可能になるが，それに応じた計算資源が必要となる．特にメモリ量が重要で，パラメータ数に応じたメモリ量が必要になる．市販の GPU で LLM を動作させるためには，性能とのトレードオフになるが，パラメータの量子化の精度を，16bit から 8bit や 4bit のように下げることで，メモリ（VRAM）量を抑えることができる．

2.2　RAG (Retrieval-Augmented Generation)

LLM はある時点のデータをもとに構築されるため，最新の知識を反映しているとは限らない．新たな知識を反映させる再学習やファインチューニングは，膨大な計算資源を必要とするため，ローカル LLM の動作環境では不十分であることが多い．

RAG は，LLM の能力を拡張するためのアプローチであり，外部データベースから関連情報を取得して回答を補強する技術である．これによって，特定の業務や専門領域に適した情報を柔軟に取り込むことが可能となる．

RAG の基本構成を図 1 に示す．ユーザからの質問文[query] は検索器（Retriever）に渡され，それを基にナレッジベース（Knowledge base）を検索する．取得した関連情報[contexts]は LLM に渡され，回答[response] が得られる．ナレッジベースは，対象ファイルを読み込み，情報の抽出・解析・変換（Extraction, Parsing），検索精度を向上させるための文章分割（Chunking），チャンクをベクトル化して検索可能にするための埋め込み（Embedding），検索を最適化するためのインデックス化（Indexing）といった複数の処理を経て構築されるデータベースであり，ベクトルデータベースを中心に，適切なデータ管理システムが利用される．

3.　AI チャットボットシステムの試作

3.1　システム構成

試作したシステムでは，図 1 を基本とするローカル LLM と RAG による構成を採用した．表 1 にハードウェア及び

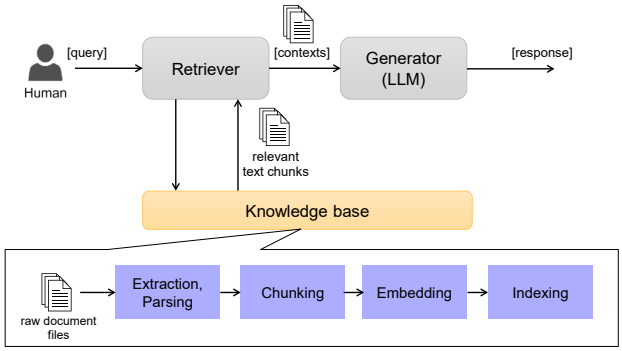

図 1　RAG の基本構成とナレッジベースの構築方法

表 1　ハードウェア及びソフトウェアの構成

ハードウェア	CPU	Intel Core i9-10980XE
	RAM	256GB
	GPU	NVIDIA RTX6000 Ada (48GB) ×2
	OS	Ubuntu 22.04 LTS
ソフトウェア	推論エンジン	Ollama 0.5.1
	フレームワーク	LlamaIndex 0.12.15
	ベクトル DB	Qdrant 1.14.2
	PDF Parser	PyMuPDF4LLM 0.0.18

ソフトウェア構成を示す. LLM は 8bit 量子化した tokyotech-llm/Llama-3.3-Swallow-70B-v0.4 と Embedding モデルに cl-nagoya/ruri-v3-310m を用いた.

3.2 ナレッジベースの構築

ナレッジベースの構築手順は図 1 の通りである．本節では AI チャットボット等の LLM アプリケーションを作成するフレームワークである LlamaIndex を使用して構築したナレッジベースについて詳述する.

ナレッジベースに取り込む文書は，当所が公開している規定（社内規定等）の PDF 文書とし，その中から 5 件を選択した．これらの PDF 文書を表 1 の PDF Parser を用いて Markdown 形式のテキストデータに変換し，さらに自作の整形プログラムを介して，章や節のタイトルを Markdown 表記の見出しにする等の修正を加えた．Chunking は，Markdown 表記に応じた Chunking を可能にする LlamaIndex のモジュール（MarkdownNodeParser）を使用した．これに合わせて Retriever には文書構造に応じた検索を可能にするモジュール（RecursiveRetriver）を用いた．これによって文書構造に着目した検索が可能となった.

4. 評価

4.1 評価方法

人手で 50 問の質問とその答え（基準回答）を用意し，RAG の性能評価を，人手と自動評価ツール Ragas[2]を用いて行った.

人手による評価方法は，基準回答[ground_truth] と AI の回答 [response] を比較して，意味的に一致しているかを判断する方法とした．評価基準は表 2 の通りである.

Ragas による評価では，人が入力する質問文 [query]，基準回答 [ground_truth]，検索によって取得された関連情報 [contexts]，及び AI によって生成された回答 [response] を基に，一部では評価用 LLM による判断を介して評価値が

表 2 人手による評価の評価基準

評価基準	5：完全に一致している
	4：ほぼ一致している
	3：少し誤りはあるが概ね一致している
	2：一部に明確な誤りがある
	1：大部分で明確な誤りがある

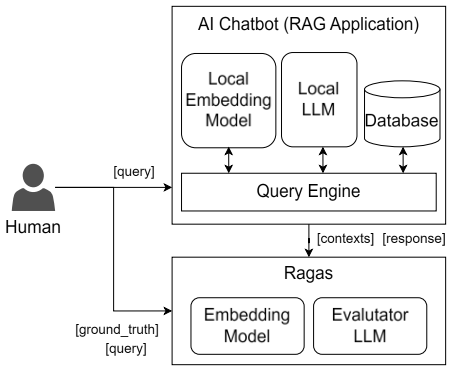

図 2 Ragas による評価

表 3 評価結果

人手	平均スコア	4.42
	評価 4 と 5 の割合	0.84
Ragas	Context Recall	0.95
	Faithfulness	0.92
	Factual correctness (precision)	0.55
	Semantic similarity	0.95

算出される（図 2）．本稿では，Context recall（関連情報が基準回答にどれだけ関連しているかを測る指標），Faithfulness（AI の回答が関連情報を正しく反映しているかを測る指標），Factual correctness（AI の回答が基準回答と矛盾していないかを測る指標），及び Semantic similarity（AI の回答と基準回答の意味的な類似度を測る指標）を算出した．評価用 LLM と評価用 Embedding モデルには，Azure OpenAI サービスで提供されている ChatGPT-4o, text-embedding-ada-002 をそれぞれ使用した.

4.2 評価結果

表 3 に人手による評価結果と Ragas v.0.2.15 を用いた自動評価の結果を示す．人手評価では，スコア 4 および 5 を正答とみなした場合の正答率は 84%であった．残りの 16%には，LLM が[contexts]から適切な回答を生成できなかったケースや，ハルシネーションにより文脈と乖離した回答が含まれていた．Ragas による評価では，Factual correctness が 0.55 であり，4 割以上の回答に矛盾があるという判定であった．自動評価ツールと人手評価の間には差があることが報告されており[3]，両者を単純に比較することは難しいが，表 3 から，Context Recall，Faithfullness，Semantic similarity は 0.9 以上と 1.0 に近いことから，AI の回答は正解に近いが基準回答との間に過不足があると推察できる.

5. おわりに

本稿ではローカル環境で動作する AI チャットボットを試作した．5 件の条文スタイルの社内文書を Markdown 形式のテキストに変換し，その文書構造に応じてチャンク分割を行い，ベクトルデータベースに格納した．このような手順を経て，人手評価で約 8 割の正解であった．約 2 割に誤りが含まれる可能性があることから，現状では正確性が求められるユースケースには適用が難しいと考えられる．今後は，適用可能なユースケースを広げるために，正答率を向上させるための最新の手法の実装に取り組む予定である.

【参考文献】

(1) 総務省：「国内外における最新の情報通信技術の研究開発及びデジタル活用の動向に関する調査研究の請負 成果報告書」，(2024 年 3 月)

(2) Ragas, https://github.com/explodinggradients/ragas

(3) Dongyu Ru, et al. “RAGChecker: A fine-grained framework for diagnosing retrieval-augmented generation,” NeurIPS 2024 (2024)