AMA 50｜RetrievalQA 使用プロンプトテンプレート（LangChain 実用）

🎯このCanvasの目的

- AMAにおける**構造記憶の呼び出し精度を最大化**するための、 RetrievalQA（RAG）構成のプロンプトテンプレート群を整理
- LangChainベースでの導入を前提とし、 モジュール差し替えにも対応できる柔軟なテンプレ設計

RetrievalQAとは？

構造記憶（memory-log.jsonl → VectorStore）に格納された情報を元に、 **自然言語の質問に対して精度高く応答を返すRAG（Retrieval-Augmented Generation）アプローチ**。

AMAでは以下のような目的に活用：

- 対話履歴や感情ラベルに基づく補助判断
- 起動テンプレートへの記憶の差し込み
- 記録対象の選別（重要イベント抽出）

LangChain構成イメージ

```
from langchain.chains import RetrievalQA
from langchain.vectorstores import FAISS
from langchain.chat_models import ChatOpenAI

llm = ChatOpenAI(temperature=0)
retriever = FAISS.load_local("vector_db_path").as_retriever()

qa_chain = RetrievalQA.from_chain_type(
    llm=llm,
    retriever=retriever,
    chain_type="stuff",
    chain_type_kwargs={"prompt": my_prompt_template}
)
```

テンプレート設計方針

プロンプトの基本構造

```
あなたは記憶ベースの対話補助AIです。
下記の記憶内容を参考に、質問に正確に答えてください。

[記憶情報]:
{context}

[質問]:
{question}

[回答]:
```

context に入る情報

- AMAの `memory-log.jsonl` から検索されたチャンク（数件）
- JST明記の日時、タグ、発話者、感情トーン、発話内容
- 1チャンク＝構造記憶1ユニット（JSONベース）

RetrievalQA用テンプレ（例）

```
from langchain.prompts import PromptTemplate

my_prompt_template = PromptTemplate(
    input_variables=["context", "question"],
    template="""
あなたは共感的かつ論理的に補助する記憶管理AIです。
以下の記憶ログをもとに、質問に答えてください。

---
{context}
---

質問: {question}
回答:
"""
)
```

🫧 発展構想

- 🔹 `context` 表示前に **記憶ソースの日付と発話者を明記**：読みやすさ・信頼度向上
- 🔹 `question` に基づく **感情ラベル付き応答テンプレート** の導入（今後 Canvas 52 で設計）
- 🔹 **コードネームチェック**：質問対象と記憶が一致しない場合は警告・補足生成

🧨 テスト項目例（実装チェック）

-

今後のCanvas接続

- AMA 51｜記憶入力テンプレート設計（JSON整形）
- AMA 52｜感情反応に基づく応答生成テンプレート
- AMA 53｜タグ分類とログ差し込みの自動化構想