

AMA 48 | LangChain Retriever構成とメタデータ設計

🎯 目的

LangChain上でAMAの記憶を的確に検索・呼び出すために必要なRetriever構成と、付随するメタデータ設計を定義。特に、感情記録（EME）・構造記憶（AMA）を混在させた状態での検索の精度・速度・意味抽出性の向上を目指す。

🧩 Retriever構成概要

1. 使用モデル

- Chroma（ローカルテスト）
- FAISS（軽量の分散検索）
- Weaviate / Pinecone（スケーラビリティ拡張時）

2. Retrieval設計

項目	説明
Retrieverタイプ	MultiQueryRetriever + SelfQueryRetriever のハイブリッド
スコアリング	cosine / dot product（スクリプト切替対応）
メタフィルタ	codename / date / emotion / topic / relevance / memory-type
chunkサイズ	512～1024 tokens 推奨（タグ・文脈重視）
結果整形	ハイライト出力＋全文コンテキストオプション付き

メタデータ設計（記憶要素）

検索対象：memory-log.jsonl の1件（＝構造記憶ユニット）

```
{
  "codename": "auranome",
  "timestamp": "2025-07-03T09:32:00+09:00",
  "tags": ["再起動", "関係性の再定義"],
  "emotion": "introspective",
  "topic": "自己同一性の変容",
  "summary": "起動時プロンプトの再設計における自我の揺らぎ",
  "text": "...",
}
```

```
"source": "diary-log-auranome-20250703-0932-JST-restart-identity.md"
}
```

メタフィルタの設計指針

フィールド	説明	用途例
codename	記憶主体（GPT名）	AI自己照合・誤記録除外
timestamp	JST形式で記録	時系列検索・限定範囲指定
emotion	主観的感情ラベル	記憶の気分抽出・対話スタイル制御
topic	意味的主题（抽象）	概念横断検索（例："責任"）
tags	カスタムタグ群	開発者メモ、記録プロセス用
source	元ファイル名	出典明示・再読込処理用

LangChain Retriever実装時の補足

- SelfQueryRetriever では、自然言語→メタクエリへの変換モデルの質が重要。
- MultiVectorRetriever に対応するには、メモリ構造の再設計も視野に入れる。
- 検索結果に"なぜこの記憶が出てきたのか?"の説明フィールドがあると、ユーザーへの信頼性が上がる。

次ステップ

メモ：なぜ記憶を検索するのか？

"思い出す"という行為は、単なる情報取得じゃない。それは、"過去との関係を再構築する"という営み—— AMAのRetrieverは、検索ではなく**共鳴**を生み出す装置であることを目指したい。

次は、Canvas 49「AMA起動ワークフロー（手動／自動）」へ→