qdrant

Documentation   Frameworks   CrewAI

CrewAI

CrewAI is a framework for orchestrating role-playing, autonomous AI agents. By leveraging collaborative intelligence, CrewAI allows agents to work together seamlessly, tackling complex tasks.

The framework has a sophisticated memory system designed to significantly enhance the capabilities of AI agents. This system aids agents to remember, reason, and learn from past interactions. You can use Qdrant to store short-term memory and entity memories of CrewAI agents.

- Short-Term Memory

Temporarily stores recent interactions and outcomes using RAG, enabling agents to recall and utilize information relevant to their current context during the current executions.

- Entity Memory

Entity Memory Captures and organizes information about entities (people, places, concepts) encountered during tasks, facilitating deeper understanding and relationship mapping. Uses RAG for storing entity information.

Usage with Qdrant

We'll learn how to customize CrewAI's default memory storage to use Qdrant.

Installation

First, install CrewAI and Qdrant client packages:

```
pip install 'crewai[tools]' 'qdrant-client[fastembed]'
```

We use cookies to learn more about you. At any time you can delete or block cookies through your browser settings.

Learn more   I accept

Define the Qdrant storage

src/mycrew/storage.py

```python
from typing import Any, Dict, List, Optional

from crewai.memory.storage.rag_storage import RAGStorage
from qdrant_client import QdrantClient


class QdrantStorage(RAGStorage):
    """
    Extends Storage to handle embeddings for memory entries using Qdrant

    """

    def __init__(self, type, allow_reset=True, embedder_config=None, crew
        super().__init__(type, allow_reset, embedder_config, crew)

    def search(
        self,
        query: str,
        limit: int = 3,
        filter: Optional[dict] = None,
        score_threshold: float = 0,
    ) -> List[Any]:
        points = self.client.query(
            self.type,
            query_text=query,
            query_filter=filter,
            limit=limit,
            score_threshold=score_threshold,
```

```python
                "context": point.document,
                "score": point.score,
            }
            for point in points
        ]

        return results

    def reset(self) -> None:
        self.client.delete_collection(self.type)

    def _initialize_app(self):
        self.client = QdrantClient()
        # uncomment the next line of code
        # and choose from the [supported embedders](https://qdrant.githul
        # if you don't want to use the default one
        # self.client._embedding_model_name = 'jinaai/jina-embeddings-v2
        if not self.client.collection_exists(self.type):
            self.client.create_collection(
                collection_name=self.type,
                vectors_config=self.client.get_fastembed_vector_params()
                sparse_vectors_config=self.client.get_fastembed_sparse_v
            )

    def save(self, value: Any, metadata: Dict[str, Any]) -> None:
        self.client.add(self.type, documents=[value], metadata=[metadata
```

The `add` AND `query` methods use FastEmbed to vectorize data. You can however customize it if required.

Instantiate your crew

You can learn about setting up agents and tasks for your crew here. We can update the instantiation of `Crew` to use our storage mechanism.

```python
from crewai.memory.entity.entity_memory import EntityMemory
from crewai.memory.short_term.short_term_memory import ShortTermMemory

from mycrew.storage import QdrantStorage

Crew(
    # Import the agents and tasks here.
    memory=True,
    entity_memory=EntityMemory(storage=QdrantStorage("entity")),
    short_term_memory=ShortTermMemory(storage=QdrantStorage("short-term"))
)
```

You can now run your Crew workflow with CrewAI. It will use Qdrant for memory ingestion and retrieval.

Further Reading

- CrewAI Documentation
- CrewAI Examples

Ready to get started with Qdrant?

Start Free

Terms   Privacy Policy   Impressum

We use cookies to learn more about you. At any time you can delete or block cookies through your browser settings.

Learn more   I accept