LangChain Docs

OllamaEmbeddings

OllamaEmbeddings

Copy page

This will help you get started with Ollama embedding models using LangChain. For detailed documentation on `OllamaEmbeddings` features and configuration options, please refer to the **API reference**.

Overview

Integration details

Setup

First, follow **these instructions** to set up and run a local Ollama instance:

- **Download** and install Ollama onto the available supported platforms (including Windows Subsystem for Linux aka WSL, macOS, and Linux)
  - macOS users can install via Homebrew with `brew install ollama` and start with `brew services start ollama`
- Fetch available LLM model via `ollama pull <name-of-model>`
  - View a list of available models via the **model library**
  - e.g., `ollama pull llama3`
- This will download the default tagged version of the model. Typically, the default points to the latest, smallest sized-parameter model.

> On Mac, the models will be download to `~/.ollama/models` On Linux (or WSL), the models will be stored at `/usr/share/ollama/.ollama/models`

Specify the exact version of the model of interest as such `ollama pull vicuna:13b-v1.5-16k-q4_0` (View the **various tags for the `Vicuna`** model in this

LangChain Docs

OllamaEmbeddings

model>

View the **Ollama documentation** for more commands. You can run `ollama help` in the terminal to see available commands.

To enable automated tracing of your model calls, set your **LangSmith** API key:

```
os.environ["LANGSMITH_TRACING"] = "true"
os.environ["LANGSMITH_API_KEY"] = getpass.getpass("Enter your LangSmith API
```

Installation

The LangChain Ollama integration lives in the `langchain-ollama` package:

```
pip install -qU langchain-ollama
```

Instantiation

Now we can instantiate our model object and generate embeddings:

```
from langchain_ollama import OllamaEmbeddings

embeddings = OllamaEmbeddings(
    model="llama3",
)
```

Indexing and retrieval

Embedding models are often used in retrieval-augmented generation (RAG) flows, both as part of indexing data as well as later retrieving it. For more detailed instructions, e see our **RAG tutorials**.

LangChain Docs

OllamaEmbeddings

LangChain Docs

OllamaEmbeddings

Below, see how to index and retrieve data using the `embeddings` object we initialized above. In this example, we will index and retrieve a sample document in the `InMemoryVectorStore`.

```
vectorstore = InMemoryVectorStore.from_texts(
    [text],
    embedding=embeddings,
)

# Use the vectorstore as a retriever
retriever = vectorstore.as_retriever()

# Retrieve the most similar text
retrieved_documents = retriever.invoke("What is LangChain?")

# Show the retrieved document's content
print(retrieved_documents[0].page_content)
```

```
LangChain is the framework for building context-aware reasoning a
```

Direct usage

Under the hood, the vectorstore and retriever implementations are calling `embeddings.embed_documents(...)` and `embeddings.embed_query(...)` to create embeddings for the text(s) used in `from_texts` and retrieval `invoke` operations, respectively.

You can directly call these methods to get embeddings for your own use cases.

Embed single texts

You can embed single texts or documents with `embed_query`:

```
single_vector = embeddings.embed_query(text)
print(str(single_vector)[:100])  # Show the first 100 characters of the vecto
```

Embed multiple texts

You can embed multiple texts with `embed_documents`:

```
text2 = (
    "LangGraph is a library for building stateful, multi-actor applications w
)
two_vectors = embeddings.embed_documents([text, text2])
for vector in two_vectors:
    print(str(vector)[:100])  # Show the first 100 characters of the vector
```

```
[-0.0039849705, 0.023019705, -0.001768838, -0.0058736936, 0.00040
[-0.0066985516, 0.009878328, 0.008019467, -0.009384944, -0.029560851, 0.02574
```

API reference

For detailed documentation on `OllamaEmbeddings` features and configuration options, please refer to the **API reference**.

LangChain Docs

OllamaEmbeddings

LangChain Docs

Resources

Forum

Changelog

LangChain Academy

Trust Center

Company

About

Careers

Blog

Powered by mintlify