



Modules

Retrieval

Text embedding models

Text embedding models

! INFO

Head to [Integrations](#) for documentation on built-in integrations with text embedding model providers.

The Embeddings class is a class designed for interfacing with text embedding models. There are lots of embedding model providers (OpenAI, Cohere, Hugging Face, etc) - this class is designed to provide a standard interface for all of them.

Embeddings create a vector representation of a piece of text. This is useful because it means we can think about text in the vector space, and do things like semantic search where we look for pieces of text that are most similar in the vector space.

The base Embeddings class in LangChain provides two methods: one for embedding documents and one for embedding a query. The former takes as input multiple texts, while the latter takes a single text. The reason for having these as two separate methods is that some embedding providers have different embedding methods for documents (to be searched over) vs queries (the search query itself).

Get started

Setup

To start we'll need to install the OpenAI partner package:

```
pip install langchain-openai
```

Accessing the API requires an API key, which you can get by creating an account and heading [here](#). Once we have a key we'll want to set it as an environment variable by running:

```
export OPENAI_API_KEY="..."
```

If you'd prefer not to set an environment variable you can pass the key in directly via the `openai_api_key` named parameter when initiating the OpenAI LLM class:

```
from langchain_openai import OpenAIEmbeddings

embeddings_model =
OpenAIEmbeddings(openai_api_key="...")
```

Otherwise you can initialize without any params:

```
from langchain_openai import OpenAIEmbeddings

embeddings_model = OpenAIEmbeddings()
```

embed_documents

Embed list of texts

```
embeddings =
embeddings_model.embed_documents(
    [
        "Hi there!",
        "Oh, hello!",
        "What's your name?",
        "My friends call me World",
        "Hello World!"
    ]
)
len(embeddings), len(embeddings[0])
```

```
(5, 1536)
```

embed_query

Embed single query

Embed a single piece of text for the purpose of comparing to other embedded pieces of texts.

```
embedded_query =  
embeddings_model.embed_query("What was the  
name mentioned in the conversation?")  
embedded_query[:5]
```

```
[0.0053587136790156364,  
 -0.0004999046213924885,  
 0.038883671164512634,  
 -0.003001077566295862,  
 -0.00900818221271038]
```