🏠   Modules   Data connection   Retrievers   Vector store-backed retriever

Vector store-backed retriever

A vector store retriever is a retriever that uses a vector store to retrieve documents. It is a lightweight wrapper around the Vector Store class to make it conform to the Retriever interface. It uses the search methods implemented by a vector store, like similarity search and MMR, to query the texts in the vector store.

Once you construct a Vector store, it's very easy to construct a retriever. Let's walk through an example.

```python
from langchain.document_loaders import TextLoader
loader = TextLoader('../../../state_of_the_union.txt')
```

```python
from langchain.text_splitter import CharacterTextSplitter
from langchain.vectorstores import FAISS
from langchain.embeddings import OpenAIEmbeddings

documents = loader.load()
text_splitter = CharacterTextSplitter(chunk_size=1000, chunk_overlap=0)
texts = text_splitter.split_documents(documents)
embeddings = OpenAIEmbeddings()
db = FAISS.from_documents(texts, embeddings)
```

```
    Exiting: Cleaning up .chroma directory
```

```python
retriever = db.as_retriever()
```

```python
docs = retriever.get_relevant_documents("what did he say about ketanji brown jackson")
```

Maximum Marginal Relevance Retrieval

By default, the vectorstore retriever uses similarity search. If the underlying vectorstore support maximum marginal relevance search, you can specify that as the search type.

```python
retriever = db.as_retriever(search_type="mmr")
```

```python
docs = retriever.get_relevant_documents("what did he say about ketanji brown jackson")
```

Similarity Score Threshold Retrieval

You can also a retrieval method that sets a similarity score threshold and only returns documents with a score above that threshold

```python
retriever = db.as_retriever(search_type="similarity_score_threshold", search_kwargs={"score_threshold": .5})
```

```python
docs = retriever.get_relevant_documents("what did he say about ketanji brown jackson")
```

Specifying top k

You can also specify search kwargs like `k` to use when doing retrieval.

```python
retriever = db.as_retriever(search_kwargs={"k": 1})
```

```python
docs = retriever.get_relevant_documents("what did he say about ketanji brown jackson")
```

```python
len(docs)
```

```
    1
```