VectorStore Retriever

The index - and therefore the retriever - that LangChain has the most support for is a VectorStoreRetriever. As the name suggests, this retriever is backed heavily by a VectorStore.

Once you construct a VectorStore, its very easy to construct a retriever. Let’s walk through an example.

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

```
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
```

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

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

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.

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

Skip to main content

```
jackson")
```

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

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

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

```
len(docs)
```

```
1
```