Weaviate

This notebook shows how to use functionality related to the Weaviate vector database.

```
from langchain.embeddings.openai import OpenAIEmbeddings
from langchain.text_splitter import CharacterTextSplitter
from langchain.vectorstores import Weaviate
from langchain.document_loaders import TextLoader
```

```
from langchain.document_loaders import TextLoader
loader = TextLoader('../../../state_of_the_union.txt')
documents = loader.load()
text_splitter = CharacterTextSplitter(chunk_size=1000, chunk_overlap=0)
docs = text_splitter.split_documents(documents)

embeddings = OpenAIEmbeddings()
```

```
import weaviate
import os

WEAVIATE_URL = ""
client = weaviate.Client(
    url=WEAVIATE_URL,
    additional_headers={
        'X-OpenAI-Api-Key': os.environ["OPENAI_API_KEY"]
    }
)
```

```
client.schema.delete_all()
client.schema.get()
schema = {
    "classes": [
        {
            "class": "Paragraph",
            "description": "A written paragraph",
            "vectorizer": "text2vec-openai",
              "moduleConfig": {
                "text2vec-openai": {
                  "model": "babbage",
                  "type": "text"
                }
```

Skip to main content

```
                {
                    "dataType": ["text"],
                    "description": "The content of the paragraph",
                    "moduleConfig": {
                        "text2vec-openai": {
                          "skip": False,
                          "vectorizePropertyName": False
                        }
                      },
                    "name": "content",
                },
            ],
        },
    ]
}

client.schema.create(schema)
```

```
vectorstore = Weaviate(client, "Paragraph", "content")
```

```
query = "What did the president say about Ketanji Brown Jackson"
docs = vectorstore.similarity_search(query)
```

```
print(docs[0].page_content)
```