pdf_to_faiss_db

July 4, 2025

[6]:

```
import fitz
import os
from api_key import OPENAI_API_KEY
```

[9]:

```
def pdf_to_txt(path):
    pdf_path = 'docs/' + path + '.pdf'
    doc = fitz.open(pdf_path)
    txt_path = os.path.splitext(pdf_path)[0] + ".txt"
    with open(txt_path, "w", encoding="utf-8") as f:
        for page in doc:
            f.write(page.get_text())
    print(f"    {txt_path}")
```

[11]:

```
pdf_to_txt('scikit-learn-docs')
```

```
docs/scikit-learn-docs.txt
```

[12]:

```
pdf_to_txt('xgboost-readthedocs-io-en-release_0.80')
```

```
docs/xgboost-readthedocs-io-en-release_0.80.txt
```

[ ]:

```
pdf_to_txt('pandas')
```

[ ]:

```
pdf_to_txt('numpy-user')
```

[14]:

```
from langchain.schema import Document
from langchain.text_splitter import CharacterTextSplitter
```

[15]:

```
splitter = CharacterTextSplitter(chunk_size=800, chunk_overlap=100)
documents = []

for filename in ["docs/pandas.txt", "docs/numpy-user.txt", "docs/
 ↪scikit-learn-docs.txt", "docs/xgboost-readthedocs-io-en-release_0.80.txt"]:
    with open(filename, encoding="utf-8") as f:
        text = f.read()
    splits = splitter.split_text(text)
    for chunk in splits:
        documents.append(Document(page_content=chunk, metadata={"source":␣
 ↪filename}))
```

[16]:

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

[27]:

```python
from langchain.vectorstores import FAISS
from langchain.embeddings import OpenAIEmbeddings
from langchain.schema import Document
import numpy as np

embedding = OpenAIEmbeddings(openai_api_key=OPENAI_API_KEY)

#
batch_size = 100
embeddings = []
texts = []
metadatas = []

for i in range(0, len(split_docs), batch_size):
    batch = split_docs[i:i+batch_size]
    texts_batch = [doc.page_content for doc in batch]
    embeddings_batch = embedding.embed_documents(texts_batch)
    embeddings.extend(embeddings_batch)
    texts.extend(texts_batch)
    metadatas.extend([doc.metadata for doc in batch])
    print('now in range ' + str(i))
```

```
now in range 0
now in range 100
now in range 200
now in range 300
now in range 400
now in range 500
now in range 600
now in range 700
now in range 800
now in range 900
now in range 1000
now in range 1100
now in range 1200
now in range 1300
now in range 1400
now in range 1500
now in range 1600
now in range 1700
now in range 1800
now in range 1900
now in range 2000
now in range 2100
now in range 2200
now in range 2300
now in range 2400
now in range 2500
now in range 2600
now in range 2700
now in range 2800
now in range 2900
now in range 3000
now in range 3100
now in range 3200
now in range 3300
now in range 3400
now in range 3500
now in range 3600
now in range 3700
now in range 3800
now in range 3900
now in range 4000
now in range 4100
now in range 4200
now in range 4300
now in range 4400
now in range 4500
now in range 4600
now in range 4700
now in range 4800
now in range 4900
now in range 5000
now in range 5100
now in range 5200
now in range 5300
now in range 5400
now in range 5500
now in range 5600
now in range 5700
now in range 5800
now in range 5900
now in range 6000
now in range 6100
now in range 6200
now in range 6300
now in range 6400
now in range 6500
now in range 6600
now in range 6700
now in range 6800
now in range 6900
now in range 7000
now in range 7100
now in range 7200
now in range 7300
now in range 7400
now in range 7500
now in range 7600
now in range 7700
now in range 7800
now in range 7900
now in range 8000
now in range 8100
now in range 8200
now in range 8300
now in range 8400
now in range 8500
now in range 8600
now in range 8700
now in range 8800
now in range 8900
now in range 9000
now in range 9100
now in range 9200
now in range 9300
now in range 9400
now in range 9500
now in range 9600
now in range 9700
now in range 9800
now in range 9900
now in range 10000
now in range 10100
now in range 10200
now in range 10300
now in range 10400
now in range 10500
now in range 10600
now in range 10700
now in range 10800
now in range 10900
now in range 11000
now in range 11100
now in range 11200
now in range 11300
now in range 11400
now in range 11500
now in range 11600
now in range 11700
now in range 11800
now in range 11900
now in range 12000
now in range 12100
now in range 12200
now in range 12300
now in range 12400
now in range 12500
```

[29]:

```
#   FAISS
text_embedding_pairs = list(zip(texts, embeddings))
vectorstore = FAISS.from_embeddings(
    text_embeddings=text_embedding_pairs,
    embedding=embedding,  #     embedding
    metadatas=metadatas
)

vectorstore.save_local("faiss_db")
```

[ ]:

```
```