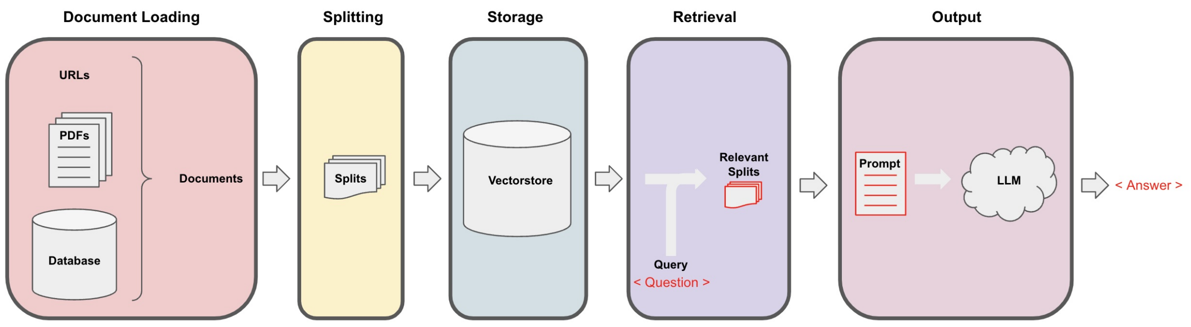
Question Answering

Overview

Recall the overall workflow for retrieval augmented generation (RAG):



We discussed `Document Loading` and `Splitting` as well as `Storage` and `Retrieval`.

Let's load our vectorDB.

In [ ]:
```
import os
import openai
import sys
sys.path.append('../..')

from dotenv import load_dotenv, find_dotenv
_ = load_dotenv(find_dotenv()) # read local .env file

openai.api_key  = os.environ['OPENAI_API_KEY']
```

The code below was added to assign the openai LLM version filmed until it is deprecated, currently in Sept 2023. LLM responses can often vary, but the responses may be significantly different when using a different model version.

In [ ]:
```
import datetime
current_date = datetime.datetime.now().date()
if current_date < datetime.date(2023, 9, 2):
    llm_name = "gpt-3.5-turbo-0301"
else:
    llm_name = "gpt-3.5-turbo"
print(llm_name)
```

```
gpt-3.5-turbo
```

In [ ]:
```
from langchain.vectorstores import Chroma
from langchain.embeddings.openai import OpenAIEmbeddings
persist_directory = 'docs/chroma/'
embedding = OpenAIEmbeddings()
vectordb = Chroma(persist_directory=persist_directory, embedding_function=embedding)
```

In [ ]:
```
print(vectordb._collection.count())
```

```
209
```

In [ ]:
```
question = "What are major topics for this class?"
docs = vectordb.similarity_search(question,k=3)
len(docs)
```

Out[ ]:
```
3
```

In [ ]:
```
from langchain.chat_models import ChatOpenAI
llm = ChatOpenAI(model_name=llm_name, temperature=0)
```

RetrievalQA chain

In [ ]:
```
from langchain.chains import RetrievalQA
```

In [ ]:
```
qa_chain = RetrievalQA.from_chain_type(
    llm,
    retriever=vectordb.as_retriever()
)
```

In [ ]:
```python
result = qa_chain({"query": question})
```

In [ ]:
```python
result["result"]
```

Out[ ]:
```
'The major topics for this class include machine learning, statistics, and algebra. Additionally, there will be discussio
ns on extensions of the material covered in the main lectures.'
```

Prompt

In [ ]:
```python
from langchain.prompts import PromptTemplate

# Build prompt
template = """Use the following pieces of context to answer the question at the end. If you don't know the answer, just sa
{context}
Question: {question}
Helpful Answer:"""
QA_CHAIN_PROMPT = PromptTemplate.from_template(template)
```

In [ ]:
```python
# Run chain
qa_chain = RetrievalQA.from_chain_type(
    llm,
    retriever=vectordb.as_retriever(),
    return_source_documents=True,
    chain_type_kwargs={"prompt": QA_CHAIN_PROMPT}
)
```

In [ ]:
```python
question = "Is probability a class topic?"
```

In [ ]:
```python
result = qa_chain({"query": question})
```

In [ ]:
```python
result["result"]
```

Out[ ]:
```
'Yes, probability is a class topic as the instructor assumes familiarity with basic probability and statistics. Thanks fo
r asking!'
```

In [ ]:
```python
result["source_documents"][0]
```

Out[ ]:
```
Document(page_content="of this class will not be very program ming intensive, although we will do some \nprogramming, mos
tly in either MATLAB or Octa ve. I'll say a bit more about that later.  \nI also assume familiarity with basic proba bili
ty and statistics. So most undergraduate \nstatistics class, like Stat 116 taught here at Stanford, will be more than eno
ugh. I'm gonna \nassume all of you know what ra ndom variables are, that all of you know what expectation \nis, what a va
riance or a random variable is. And in case of some of you, it's been a while \nsince you've seen some of this material.
At some of the discussion sections, we'll actually \ngo over some of the prerequisites, sort of as  a refresher course un
der prerequisite class. \nI'll say a bit more about that later as well.  \nLastly, I also assume familiarity with basi c
linear algebra. And again, most undergraduate \nlinear algebra courses are more than enough. So if you've taken courses l
ike Math 51, \n103, Math 113 or CS205 at Stanford, that would be more than enough. Basically, I'm \ngonna assume that all
of you know what matrix es and vectors are, that you know how to \nmultiply matrices and vectors and multiply matrix and
matrices, that you know what a matrix inverse is. If you know what an eigenvect or of a matrix is, that'd be even better.
\nBut if you don't quite know or if you're not qu ite sure, that's fine, too. We'll go over it in \nthe review sections."
, metadata={'source': 'docs/cs229_lectures/MachineLearning-Lecture01.pdf', 'page': 4})
```

RetrievalQA chain types

In [ ]:
```python
qa_chain_mr = RetrievalQA.from_chain_type(
    llm,
    retriever=vectordb.as_retriever(),
    chain_type="map_reduce"
)
```

In [ ]:
```python
result = qa_chain_mr({"query": question})
```

In [ ]:
```python
result["result"]
```

Out[ ]:
```
'Yes, probability is a class topic mentioned in the document. The instructor assumes familiarity with basic probability a
nd statistics, including concepts like random variables, expectation, variance, and random variables. The document mentio
ns that most undergraduate statistics classes, like Stat 116, will provide sufficient background knowledge for the cours
e.'
```

If you wish to experiment on the `LangSmith platform` (previously known as LangChain Plus):

- Go to LangSmith and sign up
- Create an API key from your account's settings
- Use this API key in the code below
- uncomment the code
  Note, the endpoint in the video differs from the one below. Use the one below.

In [ ]:
```
#import os
#os.environ["LANGCHAIN_TRACING_V2"] = "true"
#os.environ["LANGCHAIN_ENDPOINT"] = "https://api.langchain.plus"
#os.environ["LANGCHAIN_API_KEY"] = "..." # replace dots with your api key
```

In [ ]:
```
qa_chain_mr = RetrievalQA.from_chain_type(
    llm,
    retriever=vectordb.as_retriever(),
    chain_type="map_reduce"
)
result = qa_chain_mr({"query": question})
result["result"]
```

Out[ ]:
```
'Yes, probability is a class topic.'
```

In [ ]:
```
qa_chain_mr = RetrievalQA.from_chain_type(
    llm,
    retriever=vectordb.as_retriever(),
    chain_type="refine"
)
result = qa_chain_mr({"query": question})
result["result"]
```

Out[ ]:
```
"The class will cover probability topics, assuming familiarity with basic probability and statistics. The instructor ment
ioned that linear regression can be endowed with a probabilistic interpretation, which will be used to derive the next le
arning algorithm, a classification algorithm. This algorithm will address problems where the variable Y being predicted i
s a discrete value, such as in binary classification where Y takes on only two values. Examples of classification problem
s include medical diagnosis and predicting whether a house will sell in the next six months. The course may involve some
programming, primarily in MATLAB or Octave, but it will not be heavily programming intensive. Additionally, basic linear
algebra knowledge is also assumed, with most undergraduate linear algebra courses being adequate preparation. The instruc
tor will provide refresher courses on prerequisites during discussion sections for those who may need it. Later in the qu
arter, the discussion sections will also cover extensions for the material taught in the main lectures, as machine learni
ng is a vast field with additional topics that couldn't be covered in the main lectures."
```

RetrievalQA limitations

QA fails to preserve conversational history.

In [ ]:
```
qa_chain = RetrievalQA.from_chain_type(
    llm,
    retriever=vectordb.as_retriever()
)
```

In [ ]:
```
question = "Is probability a class topic?"
result = qa_chain({"query": question})
result["result"]
```

Out[ ]:
```
'Yes, probability is a class topic in the course being described. The instructor assumes familiarity with basic probabili
ty and statistics.'
```

In [ ]:
```
question = "why are those prerequesites needed?"
result = qa_chain({"query": question})
result["result"]
```

Out[ ]:
```
'The prerequisites for the class are needed because the course assumes that all students have a basic knowledge of comput
er science and computer skills, including understanding of big-O notation. These prerequisites are essential for students
to be able to grasp the concepts and materials covered in the machine learning course effectively.'
```

Note, The LLM response varies. Some responses **do** include a reference to probability which might be gleaned from referenced documents. The point is simply that the model does not have access to past questions or answers, this will be covered in the next section.

In [ ]:
```

```