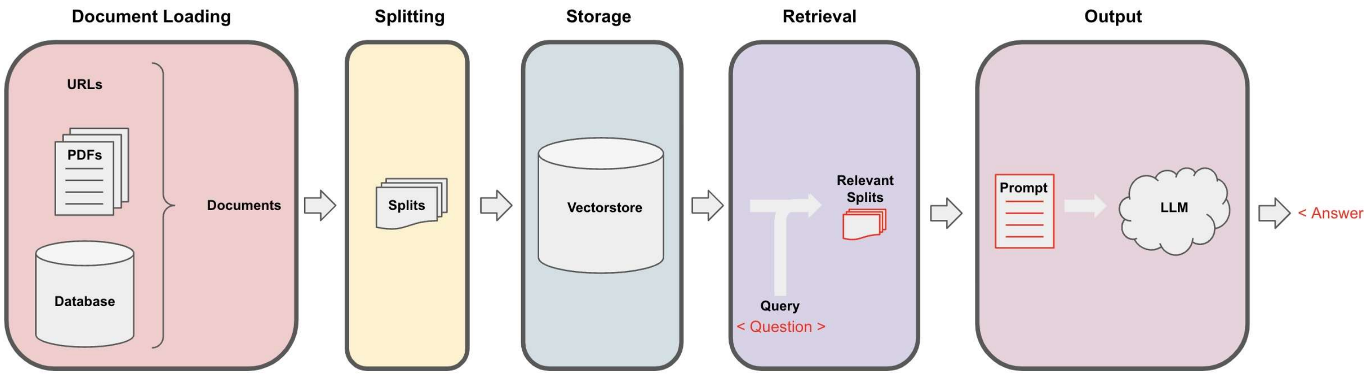
Question Answering

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 [1]:
```
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 [2]:
```
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-0301
```

In [3]:

```
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=em
```

In [4]:

```
print(vectordb._collection.count())
```

```
209
```

In [5]:

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

```
3
```

In [6]:

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

RetrievalQA chain

In [26]:

```
from langchain.chains import RetrievalQA
```

In [30]:

```
qa_chain = RetrievalQA.from_chain_type(
    llm,
    retriever=vectordb.as_retriever(),
    return_source_documents=True
)
```

In [31]:

```
result = qa_chain({"query": question})
```

In [10]:

```
result["result"]
```

```
'The major topic for this class is machine learning. Additionally, the class
may cover statistics and algebra as refreshers in the discussion sections. La
ter in the quarter, the discussion sections will also cover extensions for th
e material taught in the main lectures.'
```

In [37]:
```python
for doc in result['source_documents']:
    print(doc.page_content[:200])
```

```
So, see, most of us use learning algorithms half a dozen, a dozen, maybe doze
ns of times
without even knowing it.
And of course, learning algorithms are also  doing things like giving us a gr
owing
So, see, most of us use learning algorithms half a dozen, a dozen, maybe doze
ns of times
without even knowing it.
And of course, learning algorithms are also  doing things like giving us a gr
owing
really makes a difference between a good so lution and amazing solution. And
to give
everyone to just how we do points assignments, or what is it that causes a so
lution to get
full marks, or just ho
So in this class, we've tried to convey to you a broad set of principl es and
tools that will
be useful for doing many, many things. And ev ery time I teach this class, I
can actually
very confident
```

Prompt

In [11]:
```python
from langchain.prompts import PromptTemplate

# Build prompt
template = """Use the following pieces of context to answer the question at
{context}
Question: {question}
Helpful Answer:"""
QA_CHAIN_PROMPT = PromptTemplate.from_template(template)
```

In [12]:
```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 [13]:
```python
question = "Is probability a class topic?"
```

In [14]:
```python
result = qa_chain({"query": question})
```

```
In [15]: result["result"]
```

```
'Yes, probability is assumed to be a prerequisite for this class. The instruc
tor assumes familiarity with basic probability and statistics, and will go ov
er some of the prerequisites in the discussion sections as a refresher cours
e. Thanks for asking!'
```

```
In [16]: result["source_documents"][0]
```

```
Document(page_content="of this class will not be very program ming intensive,
although we will do some \nprogramming, mostly in either MATLAB or Octa ve.
I'll say a bit more about that later.  \nI also assume familiarity with basic
proba bility and statistics. So most undergraduate \nstatistics class, like S
tat 116 taught here at Stanford, will be more than enough. I'm gonna \nassume
all of you know what ra ndom variables are, that all of you know what expecta
tion \nis, what a variance or a random variable is. And in case of some of yo
u, it's been a while \nsince you've seen some of this material. At some of th
e discussion sections, we'll actually \ngo over some of the prerequisites, so
rt of as  a refresher course under prerequisite class. \nI'll say a bit more
about that later as well.  \nLastly, I also assume familiarity with basi c li
near algebra. And again, most undergraduate \nlinear algebra courses are more
than enough. So if you've taken courses like Math 51, \n103, Math 113 or CS20
5 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 \nm
ultiply 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, th
at'd be even better. \nBut if you don't quite know or if you're not qu ite su
re, 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 [17]: qa_chain_mr = RetrievalQA.from_chain_type(
             llm,
             retriever=vectordb.as_retriever(),
             chain_type="map_reduce"
         )
```

```
In [18]: result = qa_chain_mr({"query": question})
```

```
In [19]: result["result"]
```

```
'There is no clear answer to this question based on the given portion of the
document. The document mentions familiarity with basic probability and statis
tics as a prerequisite for the class, and there is a brief mention of probabi
lity in the text, but it is not clear if it is a main topic of the class.'
```

If you wish to experiment on the `LangChain plus platform` :

- Go to langchain plus platform (https://www.langchain.plus/) 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 [20]:

```
#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 [21]:

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

```
'There is no clear answer to this question based on the given portion of the
document. The document mentions familiarity with basic probability and statis
tics as a prerequisite for the class, and there is a brief mention of probabi
lity in the text, but it is not clear if it is a main topic of the class.'
```

In [22]:

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

```
"The class will have review sections to refresh the prerequisites, including
statistics and algebra. The instructor will also use the discussion sections
to go over extensions for the material taught in the main lectures. Machine l
earning is a vast field, and there are a few extensions that the instructor w
ants to teach but didn't have time to cover in the main lectures."
```

RetrievalQA limitations

QA fails to preserve conversational history.

In [23]:

```
qa_chain = RetrievalQA.from_chain_type(
    llm,
    retriever=vectordb.as_retriever()
)
```

In [24]:

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

```
'Yes, probability is a topic that will be assumed to be familiar to students
in this class. The instructor mentions that basic probability and statistics
are prerequisites for the class and assumes that most undergraduate statistic
s classes, like Stat 116 taught at Stanford, will be more than enough.'
```

In [25]:

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

```
'The prerequisites are needed because in this class, the instructor assumes t
hat all students have a basic knowledge of computer science and knowledge of
basic computer skills and principles. This includes knowledge of big-O notati
on and linear algebra, which are important concepts in machine learning. With
out this basic knowledge, it may be difficult for students to understand the
material covered in the class.'
```

Note, The LLM response varies. Some responses **do** include a reference to probability which might be gleanec 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 [ ]:

```
```