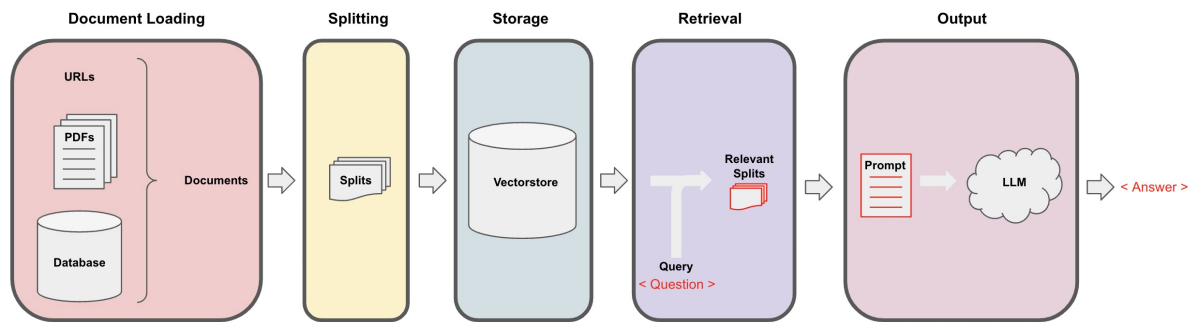


Vectorstores and Embeddings

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



```
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']
```

We just discussed Document Loading and Splitting.

```
In [ ]: from langchain.document_loaders import PyPDFLoader

# Load PDF
loaders = [
    # Duplicate documents on purpose - messy data
    PyPDFLoader("docs/cs229_lectures/MachineLearning-Lecture01.pdf"),
    PyPDFLoader("docs/cs229_lectures/MachineLearning-Lecture01.pdf"),
    PyPDFLoader("docs/cs229_lectures/MachineLearning-Lecture02.pdf"),
    PyPDFLoader("docs/cs229_lectures/MachineLearning-Lecture03.pdf")
]
docs = []
for loader in loaders:
    docs.extend(loader.load())
```

```
In [ ]: # Split
from langchain.text_splitter import RecursiveCharacterTextSplitter
text_splitter = RecursiveCharacterTextSplitter(
    chunk_size = 1500,
    chunk_overlap = 150
)
```

```
In [ ]: splits = text_splitter.split_documents(docs)
```

```
In [ ]: len(splits)
```

```
Out[ ]: 209
```

Embeddings

Let's take our splits and embed them.

```
In [ ]: from langchain.embeddings.openai import OpenAIEmbeddings
embedding = OpenAIEmbeddings()
```

```
In [ ]: sentence1 = "i like dogs"
sentence2 = "i like canines"
sentence3 = "the weather is ugly outside"
```

```
In [ ]: embedding1 = embedding.embed_query(sentence1)
embedding2 = embedding.embed_query(sentence2)
embedding3 = embedding.embed_query(sentence3)
```

```
In [ ]: import numpy as np
```

```
In [ ]: np.dot(embedding1, embedding2)
```

```
Out[ ]: 0.9631227602970008
```

```
In [ ]: np.dot(embedding1, embedding3)
```

```
Out[ ]: 0.7703256969056435
```

```
In [ ]: np.dot(embedding2, embedding3)
```

```
Out[ ]: 0.7591626802162056
```

Vectorstores

```
In [ ]: # ! pip install chromadb
```

```
In [ ]: from langchain.vectorstores import Chroma
```

```
In [ ]: persist_directory = 'docs/chroma/'
```

```
In [ ]: !rm -rf ./docs/chroma # remove old database files if any
```

```
In [ ]: vectordb = Chroma.from_documents(  
    documents=splits,  
    embedding=embedding,  
    persist_directory=persist_directory  
)
```

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

```
209
```

Similarity Search

```
In [ ]: question = "is there an email i can ask for help"
```

```
In [ ]: docs = vectordb.similarity_search(question,k=3)
```

```
In [ ]: len(docs)
```

```
Out[ ]: 3
```

```
In [ ]: docs[0].page_content
```

```
Out[ ]: "cs229-qa@cs.stanford.edu. This goes to an account that's read by all the TAs and me. So \nrather than sending us email  
individually, if you send email to this account, it will \nactually let us get back to you maximally quickly with answers  
to your questions. \nIf you're asking questions about homework problems, please say in the subject line which \nassignment  
and which question the email refers to, since that will also help us to route \nyour question to the appropriate TA or  
to me appropriately and get the response back to \nyou quickly. \nLet's see. Skipping ahead – let's see – for homework  
k, one midterm, one open and term \nproject. Notice on the honor code. So one thing that I think will help you to succeed  
and \ndo well in this class and even help you to enjoy this class more is if you form a study \ngroup. \nSo start looking  
around where you're sitting now or at the end of class today, mingle a \nlittle bit and get to know your classmates  
. I strongly encourage you to form study groups \nand sort of have a group of people to study with and have a group of your  
fellow students \nto talk over these concepts with. You can also post on the class news group if you want to \nuse that  
to try to form a study group. \nBut some of the problems sets in this class are reasonably difficult. People that have  
taken the class before may tell you they were very difficult. And just I bet it would be \nmore fun for you, and you  
would probably have a better learning experience if you form a"
```

Let's save this so we can use it later!

```
In [ ]: vectordb.persist()
```

Failure modes

This seems great, and basic similarity search will get you 80% of the way there very easily.

But there are some failure modes that can creep up.

Here are some edge cases that can arise - we'll fix them in the next class.

```
In [ ]: question = "what did they say about matlab?"
```

```
In [ ]: docs = vectordb.similarity_search(question,k=5)
```

Notice that we're getting duplicate chunks (because of the duplicate MachineLearning-Lecture01.pdf in the index).

Semantic search fetches all similar documents, but does not enforce diversity.

docs[0] and docs[1] are identical.

```
In [ ]: docs[0]
```

```
Out[ ]: Document(page_content='those homeworks will be done in either MATLAB or in Octave, which is sort of – I \nknow some people call it a free version of MATLAB, which it sort of is, sort of isn\'t. \nSo I guess for those of you that haven\'t seen MATLAB before, and I know most of you \nhave, MATLAB is I guess part of the programming language that makes it very easy to write codes using matrices, to write code for numerical routines, to move data around, to \nplot data. And it\'s sort of an extremely easy to learn tool to use for implementing a lot of \nlearning algorithms. \nAnd in case some of you want to work on your own home computer or something if you \ndon\'t have a MATLAB license, for the purposes of this class, there\'s also – [inaudible] \nwrite that down [inaudible] MATLAB – there\'s also a software package called Octave \nthat you can download for free off the Internet. And it has somewhat fewer features than MATLAB, but it\'s free, and for the purposes of this class, it will work for just about \neverything. \nSo actually I, well, so yeah, just a side comment for those of you that haven\'t seen \nMATLAB before I guess, once a colleague of mine at a different university, not at \nStanford, actually teaches another machine learning course. He\'s taught it for many years. \nSo one day, he was in his office, and an old student of his from, like, ten years ago came \ninto his office and he said, "Oh, professor, professor, thank you so much for your', metadata={'source': 'docs/cs229_lectures/MachineLearning-Lecture01.pdf', 'page': 8})
```

```
In [ ]: docs[1]
```

```
Out[ ]: Document(page_content='those homeworks will be done in either MATLAB or in Octave, which is sort of – I \nknow some people call it a free version of MATLAB, which it sort of is, sort of isn\'t. \nSo I guess for those of you that haven\'t seen MATLAB before, and I know most of you \nhave, MATLAB is I guess part of the programming language that makes it very easy to write codes using matrices, to write code for numerical routines, to move data around, to \nplot data. And it\'s sort of an extremely easy to learn tool to use for implementing a lot of \nlearning algorithms. \nAnd in case some of you want to work on your own home computer or something if you \ndon\'t have a MATLAB license, for the purposes of this class, there\'s also – [inaudible] \nwrite that down [inaudible] MATLAB – there\'s also a software package called Octave \nthat you can download for free off the Internet. And it has somewhat fewer features than MATLAB, but it\'s free, and for the purposes of this class, it will work for just about \neverything. \nSo actually I, well, so yeah, just a side comment for those of you that haven\'t seen \nMATLAB before I guess, once a colleague of mine at a different university, not at \nStanford, actually teaches another machine learning course. He\'s taught it for many years. \nSo one day, he was in his office, and an old student of his from, like, ten years ago came \ninto his office and he said, "Oh, professor, professor, thank you so much for your', metadata={'source': 'docs/cs229_lectures/MachineLearning-Lecture01.pdf', 'page': 8})
```

We can see a new failure mode.

The question below asks a question about the third lecture, but includes results from other lectures as well.

```
In [ ]: question = "what did they say about regression in the third lecture?"
```

```
In [ ]: docs = vectordb.similarity_search(question,k=5)
```

```
In [ ]: for doc in docs:
    print(doc.metadata)

{'source': 'docs/cs229_lectures/MachineLearning-Lecture03.pdf', 'page': 0}
{'source': 'docs/cs229_lectures/MachineLearning-Lecture03.pdf', 'page': 14}
{'source': 'docs/cs229_lectures/MachineLearning-Lecture02.pdf', 'page': 0}
{'source': 'docs/cs229_lectures/MachineLearning-Lecture03.pdf', 'page': 6}
{'source': 'docs/cs229_lectures/MachineLearning-Lecture01.pdf', 'page': 8}
```

```
In [ ]: print(docs[4].page_content)
```

into his office and he said, "Oh, professor, professor, thank you so much for your machine learning class. I learned so much from it. There's this stuff that I learned in your class, and I now use every day. And it's helped me make lots of money, and here's a picture of my big house."

So my friend was very excited. He said, "Wow. That's great. I'm glad to hear this machine learning stuff was actually useful. So what was it that you learned? Was it logistic regression? Was it the PCA? Was it the data networks? What was it that you learned that was so helpful?" And the student said, "Oh, it was the MATLAB."

So for those of you that don't know MATLAB yet, I hope you do learn it. It's not hard, and we'll actually have a short MATLAB tutorial in one of the discussion sections for those of you that don't know it.

Okay. The very last piece of logistical thing is the discussion sections. So discussion sections will be taught by the TAs, and attendance at discussion sections is optional, although they'll also be recorded and televised. And we'll use the discussion sections mainly for two things. For the next two or three weeks, we'll use the discussion sections to go over the prerequisites to this class or if some of you haven't seen probability or statistics for a while or maybe algebra, we'll go over those in the discussion sections as a refresher for those of you that want one.

Approaches discussed in the next lecture can be used to address both!

```
In [ ]:
```