In [1]:
```
%pip install --quiet --upgrade langchain-text-splitters langchain-community langgraph
```
```
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
import getpass
import os

os.environ["LANGSMITH_TRACING"] = "true"
os.environ["LANGSMITH_API_KEY"] = getpass.getpass()
```

In [3]:
```
pip install -qU "langchain[openai]"
```
```
Note: you may need to restart the kernel to use updated packages.
```

In [4]:
```
import getpass
import os

if not os.environ.get("OPENAI_API_KEY"):
  os.environ["OPENAI_API_KEY"] = getpass.getpass("Enter API key for OpenAI: ")

from langchain.chat_models import init_chat_model

llm = init_chat_model("gpt-4o-mini", model_provider="openai")
```

In [5]:
```
pip install -qU langchain-openai
```
```
Note: you may need to restart the kernel to use updated packages.
```

In [6]:
```
import getpass
import os

if not os.environ.get("OPENAI_API_KEY"):
  os.environ["OPENAI_API_KEY"] = getpass.getpass("Enter API key for OpenAI: ")

from langchain_openai import OpenAIEmbeddings

embeddings = OpenAIEmbeddings(model="text-embedding-3-large",
                             dimensions=1024)
```

In [7]:
```
pip install -qU langchain-pinecone
```
```
Note: you may need to restart the kernel to use updated packages.
```

In [8]:
```
from langchain_pinecone import PineconeVectorStore
from pinecone import Pinecone

pc = Pinecone(api_key = "pcsk_7A1TPN_MS8g1LuZcvvpXAFVN9sobPRDXSH1ipWhhMNPxfp3yLmLpq9azhYDTDXn5af1qXN")

idxName = "ebook-rag-idx"

if not pc.has_index(idxName):
    pc.create_index_for_model(
        name = idxName,
        cloud = "aws",
        region = "us-east-1",
        embed = {
            "model":"llama-text-embed-v2",
            "field_map":{"text": "chunk_text"}
        }
    )


idx = pc.Index(idxName)

vecStore = PineconeVectorStore(embedding=embeddings, index=idx)
```
```
/opt/anaconda3/lib/python3.13/site-packages/langchain_pinecone/__init__.py:3: LangChainDeprecationWarning: As of langchain-core 0.3.0, LangChain uses pydantic v2 internall
y. The langchain_core.pydantic_v1 module was a compatibility shim for pydantic v1, and should no longer be used. Please update the code to import from Pydantic directly.

For example, replace imports like: `from langchain_core.pydantic_v1 import BaseModel`
with: `from pydantic import BaseModel`
or the v1 compatibility namespace if you are working in a code base that has not been fully upgraded to pydantic 2 yet.         from pydantic.v1 import BaseModel

  from langchain_pinecone.vectorstores import Pinecone, PineconeVectorStore
```

In [9]:
```
%pip install -qU langchain-community pypdf
```
```
Note: you may need to restart the kernel to use updated packages.
```

In [10]:
```
import bs4
from langchain import hub
from langchain_community.document_loaders import PyPDFLoader
from langchain_core.documents import Document
from langchain_text_splitters import RecursiveCharacterTextSplitter
from langgraph.graph import START, StateGraph
from typing_extensions import List, TypedDict

filePath = "/Users/priyachandrashekar/Downloads/ADarkNightsWork.pdf"
loader = PyPDFLoader(
    filePath,
    mode = "single",
)

ebook = loader.load()
```

In [11]:
```
txtSplit = RecursiveCharacterTextSplitter(chunk_size=1000, chunk_overlap=200)
splits = txtSplit.split_documents(ebook)

_ = vecStore.add_documents(documents=splits)

prompt = hub.pull("rlm/rag-prompt")
```

In [12]:
```
class State(TypedDict):
    question: str
    context: List[Document]
    answer: str

def retrieve(s: State):
    docs = vecStore.similarity_search(s["question"])
    return {"context": docs}

def generate(s: State):
    content = "\n\n".join(doc.page_content for doc in s["context"])
    msg = prompt.invoke({"question": s["question"], "context": content})
    res = llm.invoke(msg)
    return {"answer": res.content}
```

In [13]:
```
graphBuilder = StateGraph(State).add_sequence([retrieve, generate])
graphBuilder.add_edge(START, "retrieve")
graph = graphBuilder.compile()
```

In [14]:
```
response = graph.invoke({"question": "What is the theme of the book?"})
print(response["answer"])
```
```
The theme of the book revolves around the moral struggles of individuals facing guilt, the pursuit of truth, and the weight of family responsibilities. It explores the conf
licts between personal desires, societal expectations, and the consequences of one's actions. The narrative delves into the complexity of human emotions and the search for
redemption.
```

In [15]:
```
response = graph.invoke({"question": "How does the book end?"})
print(response["answer"])
```
```
The book ends with Ellinor softly asking Miss Monro to take her home after a significant ceremony. This moment signifies her transition and emotional state as she is led ho
me, akin to being blind. The ending evokes themes of helplessness and the passage from youth to adulthood.
```

In [16]:
```
response = graph.invoke({"question": "Who is Miss Monro?"})
print(response["answer"])
```
```
Miss Monro is the daughter of a precentor of a cathedral and has a reluctance to work as a teacher for wealthy tradesmen, preferring to be associated with the canons' famil
ies. She is involved in arranging matters for Ellinor, who is in a depressed state, and has a keen interest in inviting social connections to their gatherings. Miss Monro a
lso has opportunities for patronage from the Chapter, including an offer for a small house at a nominal rent.
```

In [17]:
```
response = graph.invoke({"question": "Who is the author of the book?"})
print(response["answer"])
```
```
The author of the book "A Dark Night's Work" is Elizabeth Gaskell.
```

In [ ]:

In [ ]: