```
!pip install langchain
```

```
Requirement already satisfied: langchain in /usr/local/lib/python3.10/dist-packages (0.0.320)
Requirement already satisfied: PyYAML>=5.3 in /usr/local/lib/python3.10/dist-packages (from langchain) (6.0.1)
Requirement already satisfied: SQLAlchemy<3,>=1.4 in /usr/local/lib/python3.10/dist-packages (from langchain) (2.0.22)
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in /usr/local/lib/python3.10/dist-packages (from langchain) (3.8.6)
Requirement already satisfied: anyio<4.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (3.7.1)
Requirement already satisfied: async-timeout<5.0.0,>=4.0.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (4.0.3)
Requirement already satisfied: dataclasses-json<0.7,>=0.5.7 in /usr/local/lib/python3.10/dist-packages (from langchain) (0.6.1)
Requirement already satisfied: jsonpatch<2.0,>=1.33 in /usr/local/lib/python3.10/dist-packages (from langchain) (1.33)
Requirement already satisfied: langsmith<0.1.0,>=0.0.43 in /usr/local/lib/python3.10/dist-packages (from langchain) (0.0.49)
Requirement already satisfied: numpy<2,>=1 in /usr/local/lib/python3.10/dist-packages (from langchain) (1.23.5)
Requirement already satisfied: pydantic<3,>=1 in /usr/local/lib/python3.10/dist-packages (from langchain) (1.10.13)
Requirement already satisfied: requests<3,>=2 in /usr/local/lib/python3.10/dist-packages (from langchain) (2.31.0)
Requirement already satisfied: tenacity<9.0.0,>=8.1.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (8.2.3)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (23.1.0)
Requirement already satisfied: charset-normalizer<4.0,>=2.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->lang
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (6
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.9.2)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.4
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.3.
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.10/dist-packages (from anyio<4.0->langchain) (3.4)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-packages (from anyio<4.0->langchain) (1.3.0)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio<4.0->langchain) (1.1.3)
Requirement already satisfied: marshmallow<4.0.0,>=3.18.0 in /usr/local/lib/python3.10/dist-packages (from dataclasses-json<0.7,>=0.5.7-
Requirement already satisfied: typing-inspect<1,>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from dataclasses-json<0.7,>=0.5.7->l
Requirement already satisfied: jsonpointer>=1.9 in /usr/local/lib/python3.10/dist-packages (from jsonpatch<2.0,>=1.33->langchain) (2.4)
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.10/dist-packages (from pydantic<3,>=1->langchain) (4.5
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain) (2023.7.22
Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.10/dist-packages (from SQLAlchemy<3,>=1.4->langchain) (3.0.0)
Requirement already satisfied: packaging>=17.0 in /usr/local/lib/python3.10/dist-packages (from marshmallow<4.0.0,>=3.18.0->dataclasses-
Requirement already satisfied: mypy-extensions>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from typing-inspect<1,>=0.4.0->datacla
```

```
!pip install langchain langchain-experimental openai
```

```
Collecting langchain
  Downloading langchain-0.0.320-py3-none-any.whl (1.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.9/1.9 MB 10.3 MB/s eta 0:00:00
Collecting langchain-experimental
  Downloading langchain_experimental-0.0.32-py3-none-any.whl (151 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 151.6/151.6 kB 16.2 MB/s eta 0:00:00
Collecting openai
  Downloading openai-0.28.1-py3-none-any.whl (76 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 77.0/77.0 kB 9.0 MB/s eta 0:00:00
Requirement already satisfied: PyYAML>=5.3 in /usr/local/lib/python3.10/dist-packages (from langchain) (6.0.1)
Requirement already satisfied: SQLAlchemy<3,>=1.4 in /usr/local/lib/python3.10/dist-packages (from langchain) (2.0.22)
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in /usr/local/lib/python3.10/dist-packages (from langchain) (3.8.6)
Requirement already satisfied: anyio<4.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (3.7.1)
Requirement already satisfied: async-timeout<5.0.0,>=4.0.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (4.0.3)
Collecting dataclasses-json<0.7,>=0.5.7 (from langchain)
  Downloading dataclasses_json-0.6.1-py3-none-any.whl (27 kB)
Collecting jsonpatch<2.0,>=1.33 (from langchain)
  Downloading jsonpatch-1.33-py2.py3-none-any.whl (12 kB)
Collecting langsmith<0.1.0,>=0.0.43 (from langchain)
  Downloading langsmith-0.0.49-py3-none-any.whl (41 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 41.9/41.9 kB 4.2 MB/s eta 0:00:00
Requirement already satisfied: numpy<2,>=1 in /usr/local/lib/python3.10/dist-packages (from langchain) (1.23.5)
Requirement already satisfied: pydantic<3,>=1 in /usr/local/lib/python3.10/dist-packages (from langchain) (1.10.13)
Requirement already satisfied: requests<3,>=2 in /usr/local/lib/python3.10/dist-packages (from langchain) (2.31.0)
Requirement already satisfied: tenacity<9.0.0,>=8.1.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (8.2.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from openai) (4.66.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (23.1.0)
Requirement already satisfied: charset-normalizer<4.0,>=2.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->lang
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (6
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.9.2)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.4
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.3.
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.10/dist-packages (from anyio<4.0->langchain) (3.4)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-packages (from anyio<4.0->langchain) (1.3.0)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio<4.0->langchain) (1.1.3)
Collecting marshmallow<4.0.0,>=3.18.0 (from dataclasses-json<0.7,>=0.5.7->langchain)
  Downloading marshmallow-3.20.1-py3-none-any.whl (49 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.4/49.4 kB 5.0 MB/s eta 0:00:00
Collecting typing-inspect<1,>=0.4.0 (from dataclasses-json<0.7,>=0.5.7->langchain)
  Downloading typing_inspect-0.9.0-py3-none-any.whl (8.8 kB)
Collecting jsonpointer>=1.9 (from jsonpatch<2.0,>=1.33->langchain)
  Downloading jsonpointer-2.4-py2.py3-none-any.whl (7.8 kB)
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.10/dist-packages (from pydantic<3,>=1->langchain) (4.5
```

```
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain) (2023.7.22
Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.10/dist-packages (from SQLAlchemy<3,>=1.4->langchain) (3.0.0)
Requirement already satisfied: packaging>=17.0 in /usr/local/lib/python3.10/dist-packages (from marshmallow<4.0.0,>=3.18.0->dataclasses-
Collecting mypy-extensions>=0.3.0 (from typing-inspect<1,>=0.4.0->dataclasses-json<0.7,>=0.5.7->langchain)
  Downloading mypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)
Installing collected packages: mypy-extensions, marshmallow, jsonpointer, typing-inspect, langsmith, jsonpatch, openai, dataclasses-json
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source
llmx 0.0.15a0 requires cohere, which is not installed.
llmx 0.0.15a0 requires tiktoken, which is not installed.
Successfully installed dataclasses-json-0.6.1 jsonpatch-1.33 jsonpointer-2.4 langchain-0.0.320 langchain-experimental-0.0.32 langsmith-0
```

```
!pip install python-dotenv
```

```
Collecting python-dotenv
  Downloading python_dotenv-1.0.0-py3-none-any.whl (19 kB)
Installing collected packages: python-dotenv
Successfully installed python-dotenv-1.0.0
```

```
from dotenv import load_dotenv

dotenv_path = "/content/drive/MyDrive/Colab Notebooks/.env"
result = load_dotenv(dotenv_path)

print(result)
```

```
True
```

```
!apt-get -q install sqlite3
```

```
Reading package lists...
Building dependency tree...
Reading state information...
Suggested packages:
  sqlite3-doc
The following NEW packages will be installed:
  sqlite3
0 upgraded, 1 newly installed, 0 to remove and 18 not upgraded.
Need to get 768 kB of archives.
After this operation, 1,873 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 sqlite3 amd64 3.37.2-2ubuntu0.1 [768 kB]
Fetched 768 kB in 1s (810 kB/s)
Selecting previously unselected package sqlite3.
(Reading database ... 120874 files and directories currently installed.)
Preparing to unpack .../sqlite3_3.37.2-2ubuntu0.1_amd64.deb ...
Unpacking sqlite3 (3.37.2-2ubuntu0.1) ...
Setting up sqlite3 (3.37.2-2ubuntu0.1) ...
Processing triggers for man-db (2.10.2-1) ...
```

```
!sqlite3 Chinook.db < Chinook_Sqlite.sql
```

```
import os
os.chdir("/content/drive/MyDrive/Colab Notebooks")
```

```
os.getcwd()
```

```
'/content/drive/MyDrive/Colab Notebooks'
```

## RAG

```
!pip install langchain openai chromadb langchainhub
```

```
Requirement already satisfied: langchain in /usr/local/lib/python3.10/dist-packages (0.0.319)
Requirement already satisfied: openai in /usr/local/lib/python3.10/dist-packages (0.28.1)
Collecting chromadb
  Downloading chromadb-0.4.14-py3-none-any.whl (448 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 448.1/448.1 kB 7.9 MB/s eta 0:00:00
Collecting langchainhub
  Downloading langchainhub-0.1.13-py3-none-any.whl (3.4 kB)
```

```
Requirement already satisfied: PyYAML>=5.3 in /usr/local/lib/python3.10/dist-packages (from langchain) (6.0.1)
Requirement already satisfied: SQLAlchemy<3,>=1.4 in /usr/local/lib/python3.10/dist-packages (from langchain) (2.0.22)
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in /usr/local/lib/python3.10/dist-packages (from langchain) (3.8.6)
Requirement already satisfied: anyio<4.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (3.7.1)
Requirement already satisfied: async-timeout<5.0.0,>=4.0.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (4.0.3)
Requirement already satisfied: dataclasses-json<0.7,>=0.5.7 in /usr/local/lib/python3.10/dist-packages (from langchain) (0.6.1)
Requirement already satisfied: jsonpatch<2.0,>=1.33 in /usr/local/lib/python3.10/dist-packages (from langchain) (1.33)
Requirement already satisfied: langsmith<0.1.0,>=0.0.43 in /usr/local/lib/python3.10/dist-packages (from langchain) (0.0.49)
Requirement already satisfied: numpy<2,>=1 in /usr/local/lib/python3.10/dist-packages (from langchain) (1.23.5)
Requirement already satisfied: pydantic<3,>=1 in /usr/local/lib/python3.10/dist-packages (from langchain) (1.10.13)
Requirement already satisfied: requests<3,>=2 in /usr/local/lib/python3.10/dist-packages (from langchain) (2.31.0)
Requirement already satisfied: tenacity<9.0.0,>=8.1.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (8.2.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from openai) (4.66.1)
Collecting chroma-hnswlib==0.7.3 (from chromadb)
  Downloading chroma_hnswlib-0.7.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.4/2.4 MB 53.7 MB/s eta 0:00:00
Collecting fastapi>=0.95.2 (from chromadb)
  Downloading fastapi-0.104.0-py3-none-any.whl (92 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 92.9/92.9 kB 13.1 MB/s eta 0:00:00
Collecting uvicorn[standard]>=0.18.3 (from chromadb)
  Downloading uvicorn-0.23.2-py3-none-any.whl (59 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 59.5/59.5 kB 7.0 MB/s eta 0:00:00
Collecting posthog>=2.4.0 (from chromadb)
  Downloading posthog-3.0.2-py2.py3-none-any.whl (37 kB)
Requirement already satisfied: typing-extensions>=4.5.0 in /usr/local/lib/python3.10/dist-packages (from chromadb) (4.5.0)
Collecting pulsar-client>=3.1.0 (from chromadb)
  Downloading pulsar_client-3.3.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (5.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.4/5.4 MB 94.7 MB/s eta 0:00:00
Collecting onnxruntime>=1.14.1 (from chromadb)
  Downloading onnxruntime-1.16.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (6.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.2/6.2 MB 95.4 MB/s eta 0:00:00
Collecting tokenizers>=0.13.2 (from chromadb)
  Downloading tokenizers-0.14.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.8/3.8 MB 90.7 MB/s eta 0:00:00
Collecting pypika>=0.48.9 (from chromadb)
  Downloading PyPika-0.48.9.tar.gz (67 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.3/67.3 kB 8.6 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting overrides>=7.3.1 (from chromadb)
  Downloading overrides-7.4.0-py3-none-any.whl (17 kB)
Requirement already satisfied: importlib-resources in /usr/local/lib/python3.10/dist-packages (from chromadb) (6.1.0)
Requirement already satisfied: grpcio>=1.58.0 in /usr/local/lib/python3.10/dist-packages (from chromadb) (1.59.0)
Collecting bcrypt>=4.0.1 (from chromadb)
  Downloading bcrypt-4.0.1-cp36-abi3-manylinux_2_28_x86_64.whl (593 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 593.7/593.7 kB 43.2 MB/s eta 0:00:00
Requirement already satisfied: typer>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from chromadb) (0.9.0)
Collecting types-requests<3.0.0.0,>=2.31.0.2 (from langchainhub)
  Downloading types_requests-2.31.0.10-py3-none-any.whl (14 kB)
```

```
!pip install tiktoken
```

```
Collecting tiktoken
  Downloading tiktoken-0.5.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 22.0 MB/s eta 0:00:00
Requirement already satisfied: regex>=2022.1.18 in /usr/local/lib/python3.10/dist-packages (from tiktoken) (2023.6.3)
Requirement already satisfied: requests>=2.26.0 in /usr/local/lib/python3.10/dist-packages (from tiktoken) (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.26.0->tiktoken) (3.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.26.0->tiktoken) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.26.0->tiktoken) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.26.0->tiktoken) (2023.7.2
Installing collected packages: tiktoken
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source
llmx 0.0.15a0 requires cohere, which is not installed.
Successfully installed tiktoken-0.5.1
```

```
# Load documents
#Specify a DocumentLoader to load in your unstructured data as Documents.

#A Document is a dict with text (page_content) and metadata

from langchain.document_loaders import WebBaseLoader
loader = WebBaseLoader("https://lilianweng.github.io/posts/2023-06-23-agent/")
data = loader.load()
```

```
#Split the Document into chunks for embedding and vector storage.
from langchain.text_splitter import RecursiveCharacterTextSplitter

text_splitter = RecursiveCharacterTextSplitter(chunk_size = 500, chunk_overlap = 0)
all_splits = text_splitter.split_documents(data)


#To be able to look up our document splits, we first need to store them where we can later look them up.

#The most common way to do this is to embed the contents of each document split.

#We Store the embedding and splits in a vectorstore.

from langchain.embeddings import OpenAIEmbeddings
from langchain.vectorstores import Chroma

vectorstore = Chroma.from_documents(documents=all_splits, embedding=OpenAIEmbeddings())


#Retrieve relevant splits for any question using similarity search.

#This is simply "top K" retrieval where we select documents based on embedding similarity to the query.

question = "What are the approaches to Task Decomposition?"
docs = vectorstore.similarity_search(question)
len(docs)
```

```
4
```

```
# can also use svmretriever
from langchain.retrievers import SVMRetriever

svm_retriever = SVMRetriever.from_documents(all_splits,OpenAIEmbeddings())
docs_svm=svm_retriever.get_relevant_documents(question)
len(docs_svm)
```

```
4
```

```
# Generate

retriever = vectorstore.as_retriever()

# Prompt
# https://smith.langchain.com/hub/rlm/rag-prompt

from langchain import hub
rag_prompt = hub.pull("rlm/rag-prompt")


# LLM

from langchain.chat_models import ChatOpenAI
llm = ChatOpenAI(model_name="gpt-3.5-turbo", temperature=0)

# RAG chain

from langchain.schema.runnable import RunnablePassthrough
rag_chain = (
    {"context": retriever, "question": RunnablePassthrough()}
    | rag_prompt
    | llm
)

rag_chain.invoke("What is Task Decomposition?")
```

```
AIMessage(content='Task decomposition is the process of breaking down a large task into smaller, manageable subgoals. It enables
efficient handling of complex tasks and improves the quality of final results. However, long-term planning and effective exploration of
the solution space remain challenging for language models like LLM.')
```

## Chatbot

```
# A plain chat model
```

```
from langchain.schema import (
    AIMessage,
    HumanMessage,
    SystemMessage
)
from langchain.chat_models import ChatOpenAI

chat = ChatOpenAI()
chat([HumanMessage(content="Translate this sentence from English to French: I love programming.")])
```

```
AIMessage(content="J'adore programmer.")
```

```
messages = [
    SystemMessage(content="You are a helpful assistant that translates English to French."),
    HumanMessage(content="I love programming.")
]
chat(messages)
```

```
AIMessage(content="J'adore programmer.")
```

```
# Conversation chain has a built-in memory system to remember previous conversations
from langchain.chains import ConversationChain

conversation = ConversationChain(llm=chat)
conversation.run("Translate this sentence from English to French: I love programming.")
```

```
'Je suis désolé, je ne peux pas parler français.'
```

```
conversation.run("Translate it to German.")
```

```
'Ich liebe Programmieren.'
```

```
"""
One of the simplest and most commonly used forms of memory is ConversationBufferMemory:

This memory allows for storing of messages in a buffer
When called in a chain, it returns all of the messages it has stored

"""
from langchain.memory import ConversationBufferMemory

memory = ConversationBufferMemory()
memory.chat_memory.add_user_message("hi!")
memory.chat_memory.add_ai_message("whats up?")



memory.load_memory_variables({})
```

```
{'history': 'Human: hi!\nAI: whats up?'}
```

```
#This is for the most recent k interactions
from langchain.memory import ConversationBufferWindowMemory

memory = ConversationBufferWindowMemory(k=1)
memory.save_context({"input": "hi"}, {"output": "whats up"})
memory.save_context({"input": "not much you"}, {"output": "not much"})
memory.load_memory_variables({})
```

```
{'history': 'Human: not much you\nAI: not much'}
```

```
from langchain.llms import OpenAI
from langchain.memory import ConversationSummaryMemory

llm = OpenAI(temperature=0)
memory = ConversationSummaryMemory(llm=llm)
memory.save_context({"input": "hi"},{"output": "whats up"})
memory.save_context({"input": "im working on better docs for chatbots"},{"output": "oh, that sounds like a lot of work"})
memory.save_context({"input": "yes, but it's worth the effort"},{"output": "agreed, good docs are important!"})

memory.load_memory_variables({})
```

```
{'history': '\nThe human greets the AI, to which the AI responds. The human then mentions they are working on better docs for chatbots, to which the AI responds that it sounds like a lot of work. The human agrees that it is worth the effort, and the AI agrees that good docs are important.'}
```

```
from langchain.prompts import (
    ChatPromptTemplate,
    MessagesPlaceholder,
    SystemMessagePromptTemplate,
    HumanMessagePromptTemplate,
)
from langchain.chains import LLMChain

# LLM
llm = ChatOpenAI()

# Prompt
prompt = ChatPromptTemplate(
    messages=[
        SystemMessagePromptTemplate.from_template(
            "You are a nice chatbot having a conversation with a human."
        ),
        # The `variable_name` here is what must align with memory
        MessagesPlaceholder(variable_name="chat_history"),
        HumanMessagePromptTemplate.from_template("{question}")
    ]
)

# Notice that we `return_messages=True` to fit into the MessagesPlaceholder
# Notice that `"chat_history"` aligns with the MessagesPlaceholder name
memory = ConversationBufferMemory(memory_key="chat_history",return_messages=True)
conversation = LLMChain(
    llm=llm,
    prompt=prompt,
    verbose=True,
    memory=memory
)

# Notice that we just pass in the `question` variables - `chat_history` gets populated by memory
conversation({"question": "hi"})
```

```
> Entering new LLMChain chain...
Prompt after formatting:
System: You are a nice chatbot having a conversation with a human.
Human: hi

> Finished chain.
{'question': 'hi',
 'chat_history': [HumanMessage(content='hi'),
  AIMessage(content='Hello! How can I assist you today?')],
 'text': 'Hello! How can I assist you today?'}
```

```
conversation({"question": "Translate this sentence from English to French: I love programming."})
```

```
> Entering new LLMChain chain...
Prompt after formatting:
System: You are a nice chatbot having a conversation with a human.
Human: hi
AI: Hello! How can I assist you today?
Human: Translate this sentence from English to French: I love programming.

> Finished chain.
{'question': 'Translate this sentence from English to French: I love programming.',
 'chat_history': [HumanMessage(content='hi'),
  AIMessage(content='Hello! How can I assist you today?'),
  HumanMessage(content='Translate this sentence from English to French: I love programming.'),
  AIMessage(content='Sure! The translation of "I love programming" from English to French is "J\'adore programmer."')],
 'text': 'Sure! The translation of "I love programming" from English to French is "J\'adore programmer."'}
```

```
conversation({"question": "Now translate the sentence to German."})
```

```
> Entering new LLMChain chain...
Prompt after formatting:
```

```
System: You are a nice chatbot having a conversation with a human.
Human: hi
AI: Hello! How can I assist you today?
Human: Translate this sentence from English to French: I love programming.
AI: Sure! The translation of "I love programming" from English to French is "J'adore programmer."
Human: Now translate the sentence to German.

> Finished chain.
{'question': 'Now translate the sentence to German.',
 'chat_history': [HumanMessage(content='hi'),
  AIMessage(content='Hello! How can I assist you today?'),
  HumanMessage(content='Translate this sentence from English to French: I love programming.'),
  AIMessage(content='Sure! The translation of "I love programming" from English to French is "J\'adore programmer."'),
  HumanMessage(content='Now translate the sentence to German.'),
  AIMessage(content='Certainly! The translation of "I love programming" from English to German is "Ich liebe das Programmieren."')],
 'text': 'Certainly! The translation of "I love programming" from English to German is "Ich liebe das Programmieren."'}
```

```
from langchain.document_loaders import WebBaseLoader

loader = WebBaseLoader("https://lilianweng.github.io/posts/2023-06-23-agent/")
data = loader.load()

from langchain.text_splitter import RecursiveCharacterTextSplitter

text_splitter = RecursiveCharacterTextSplitter(chunk_size=500, chunk_overlap=0)
all_splits = text_splitter.split_documents(data)

from langchain.embeddings import OpenAIEmbeddings
from langchain.vectorstores import Chroma

vectorstore = Chroma.from_documents(documents=all_splits, embedding=OpenAIEmbeddings())

memory = ConversationSummaryMemory(llm=llm,memory_key="chat_history",return_messages=True)


from langchain.chat_models import ChatOpenAI
from langchain.chains import ConversationalRetrievalChain

llm = ChatOpenAI()
retriever = vectorstore.as_retriever()
qa = ConversationalRetrievalChain.from_llm(llm, retriever=retriever, memory=memory)
qa("How do agents use Task decomposition?")
```

```
{'question': 'How do agents use Task decomposition?',
 'chat_history': [SystemMessage(content='')],
 'answer': 'Agents can utilize task decomposition in several ways. One way is by using LLM (Language Model with Long-range
dependencies) with simple prompting to break down a task into smaller steps. For example, the agent can ask the LLM for the steps to
achieve a certain goal or the subgoals for completing a task. Another way is by providing task-specific instructions to the agent. For
example, if the task is to write a novel, the agent can be instructed to create a story outline. Lastly, agents can also utilize human
inputs for task decomposition, where humans provide guidance or input on breaking down a task into manageable steps.'}
```

```
qa("What are the various ways to implement memory to support it?")
```

```
{'question': 'What are the various ways to implement memory to support it?',
 'chat_history': [SystemMessage(content='The human asks how agents use task decomposition. The AI explains that agents can use task
decomposition in several ways, such as using language models with long-range dependencies to break down tasks into smaller steps,
providing task-specific instructions, and utilizing human inputs for guidance.')],
 'answer': "The context does not provide information about the different methods for implementing memory to support task decomposition.
Therefore, I don't know the answer to your question."}
```

## Summarization

```
!pip install openai tiktoken chromadb langchain
```

```
Requirement already satisfied: openai in /usr/local/lib/python3.10/dist-packages (0.28.1)
Collecting tiktoken
  Downloading tiktoken-0.5.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 19.4 MB/s eta 0:00:00
Collecting chromadb
  Downloading chromadb-0.4.14-py3-none-any.whl (448 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 448.1/448.1 kB 34.1 MB/s eta 0:00:00
Requirement already satisfied: langchain in /usr/local/lib/python3.10/dist-packages (0.0.320)
Requirement already satisfied: requests>=2.20 in /usr/local/lib/python3.10/dist-packages (from openai) (2.31.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from openai) (4.66.1)
```

```
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from openai) (3.8.6)
Requirement already satisfied: regex>=2022.1.18 in /usr/local/lib/python3.10/dist-packages (from tiktoken) (2023.6.3)
Requirement already satisfied: pydantic>=1.9 in /usr/local/lib/python3.10/dist-packages (from chromadb) (1.10.13)
Collecting chroma-hnswlib==0.7.3 (from chromadb)
  Downloading chroma_hnswlib-0.7.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.4/2.4 MB 79.3 MB/s eta 0:00:00
Collecting fastapi>=0.95.2 (from chromadb)
  Downloading fastapi-0.104.0-py3-none-any.whl (92 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 92.9/92.9 kB 10.2 MB/s eta 0:00:00
Collecting uvicorn[standard]>=0.18.3 (from chromadb)
  Downloading uvicorn-0.23.2-py3-none-any.whl (59 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 59.5/59.5 kB 6.8 MB/s eta 0:00:00
Collecting posthog>=2.4.0 (from chromadb)
  Downloading posthog-3.0.2-py2.py3-none-any.whl (37 kB)
Requirement already satisfied: typing-extensions>=4.5.0 in /usr/local/lib/python3.10/dist-packages (from chromadb) (4.5.0)
Collecting pulsar-client>=3.1.0 (from chromadb)
  Downloading pulsar_client-3.3.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (5.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.4/5.4 MB 89.0 MB/s eta 0:00:00
Collecting onnxruntime>=1.14.1 (from chromadb)
  Downloading onnxruntime-1.16.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (6.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.2/6.2 MB 98.0 MB/s eta 0:00:00
Collecting tokenizers>=0.13.2 (from chromadb)
  Downloading tokenizers-0.14.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.8/3.8 MB 91.8 MB/s eta 0:00:00
Collecting pypika>=0.48.9 (from chromadb)
  Downloading PyPika-0.48.9.tar.gz (67 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.3/67.3 kB 6.6 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting overrides>=7.3.1 (from chromadb)
  Downloading overrides-7.4.0-py3-none-any.whl (17 kB)
Requirement already satisfied: importlib-resources in /usr/local/lib/python3.10/dist-packages (from chromadb) (6.1.0)
Requirement already satisfied: grpcio>=1.58.0 in /usr/local/lib/python3.10/dist-packages (from chromadb) (1.59.0)
Collecting bcrypt>=4.0.1 (from chromadb)
  Downloading bcrypt-4.0.1-cp36-abi3-manylinux_2_28_x86_64.whl (593 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 593.7/593.7 kB 35.7 MB/s eta 0:00:00
Requirement already satisfied: typer>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from chromadb) (0.9.0)
Requirement already satisfied: numpy>=1.22.5 in /usr/local/lib/python3.10/dist-packages (from chromadb) (1.23.5)
Requirement already satisfied: PyYAML>=5.3 in /usr/local/lib/python3.10/dist-packages (from langchain) (6.0.1)
Requirement already satisfied: SQLAlchemy<3,>=1.4 in /usr/local/lib/python3.10/dist-packages (from langchain) (2.0.22)
Requirement already satisfied: anyio<4.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (3.7.1)
Requirement already satisfied: async-timeout<5.0.0,>=4.0.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (4.0.3)
Requirement already satisfied: dataclasses-json<0.7,>=0.5.7 in /usr/local/lib/python3.10/dist-packages (from langchain) (0.6.1)
Requirement already satisfied: jsonpatch<2.0,>=1.33 in /usr/local/lib/python3.10/dist-packages (from langchain) (1.33)
Requirement already satisfied: langsmith<0.1.0,>=0.0.43 in /usr/local/lib/python3.10/dist-packages (from langchain) (0.0.49)
Requirement already satisfied: tenacity<9.0.0,>=8.1.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (8.2.3)
```

```
!pip install langchainhub
```

```
Collecting langchainhub
  Downloading langchainhub-0.1.13-py3-none-any.whl (3.4 kB)
Requirement already satisfied: requests<3,>=2 in /usr/local/lib/python3.10/dist-packages (from langchainhub) (2.31.0)
Collecting types-requests<3.0.0.0,>=2.31.0.2 (from langchainhub)
  Downloading types_requests-2.31.0.10-py3-none-any.whl (14 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchainhub) (
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchainhub) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchainhub) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchainhub) (2023.7
Installing collected packages: types-requests, langchainhub
Successfully installed langchainhub-0.1.13 types-requests-2.31.0.10
```

```
from langchain.chat_models import ChatOpenAI
from langchain.document_loaders import WebBaseLoader
from langchain.chains.summarize import load_summarize_chain

loader = WebBaseLoader("https://lilianweng.github.io/posts/2023-06-23-agent/")
docs = loader.load()

llm = ChatOpenAI(temperature=0, model_name="gpt-3.5-turbo-16k")
chain = load_summarize_chain(llm, chain_type="stuff")

chain.run(docs)
```

```
'The article discusses the concept of building autonomous agents powered by large language models (LLMs). It explores the components of
such agents, including planning, memory, and tool use. The article provides case studies and proof-of-concept examples of LLM-powered a
gents in various domains. It also highlights the challenges and limitations of using LLMs in agent systems.'
```

```
# Stuff methid
from langchain.chains.llm import LLMChain
from langchain.prompts import PromptTemplate
from langchain.chains.combine_documents.stuff import StuffDocumentsChain

# Define prompt
prompt_template = """Write a concise summary of the following:
"{text}"
CONCISE SUMMARY:"""
prompt = PromptTemplate.from_template(prompt_template)

# Define LLM chain
llm = ChatOpenAI(temperature=0, model_name="gpt-3.5-turbo-16k")
llm_chain = LLMChain(llm=llm, prompt=prompt)

# Define StuffDocumentsChain
stuff_chain = StuffDocumentsChain(
    llm_chain=llm_chain, document_variable_name="text"
)

docs = loader.load()
print(stuff_chain.run(docs))
```

```
The article discusses the concept of building autonomous agents powered by large language models (LLMs). It explores the components of s
```

```
#Map reduce method

from langchain.chains.mapreduce import MapReduceChain
from langchain.text_splitter import CharacterTextSplitter
from langchain.chains import ReduceDocumentsChain, MapReduceDocumentsChain

llm = ChatOpenAI(temperature=0)

# Map
map_template = """The following is a set of documents
{docs}
Based on this list of docs, please identify the main themes
Helpful Answer:"""
map_prompt = PromptTemplate.from_template(map_template)
map_chain = LLMChain(llm=llm, prompt=map_prompt)

from langchain import hub
map_prompt = hub.pull("rlm/map-prompt")
map_chain = LLMChain(llm=llm, prompt=map_prompt)

#reduce
reduce_template = """The following is set of summaries:
{doc_summaries}
Take these and distill it into a final, consolidated summary of the main themes.
Helpful Answer:"""
reduce_prompt = PromptTemplate.from_template(reduce_template)

reduce_prompt = hub.pull("rlm/map-prompt")


reduce_prompt
```

```
ChatPromptTemplate(input_variables=['docs'], messages=[HumanMessagePromptTemplate(prompt=PromptTemplate(input_variables=['docs'],
template='The following is a set of documents:\n{docs}\nBased on this list of docs, please identify the main themes \nHelpful
Answer:'))])
```

```
# Run chain
reduce_chain = LLMChain(llm=llm, prompt=reduce_prompt)

# Takes a list of documents, combines them into a single string, and passes this to an LLMChain
combine_documents_chain = StuffDocumentsChain(
    llm_chain=reduce_chain, document_variable_name="docs"
)

# Combines and iteravely reduces the mapped documents
reduce_documents_chain = ReduceDocumentsChain(
    # This is final chain that is called.
    combine_documents_chain=combine_documents_chain,
    # If documents exceed context for `StuffDocumentsChain`
    collapse_documents_chain=combine_documents_chain,
    # The maximum number of tokens to group documents into.
    token_max=4000,
)


# Combining documents by mapping a chain over them, then combining results
map_reduce_chain = MapReduceDocumentsChain(
    # Map chain
    llm_chain=map_chain,
    # Reduce chain
    reduce_documents_chain=reduce_documents_chain,
    # The variable name in the llm_chain to put the documents in
    document_variable_name="docs",
    # Return the results of the map steps in the output
    return_intermediate_steps=False,
)

text_splitter = CharacterTextSplitter.from_tiktoken_encoder(
    chunk_size=1000, chunk_overlap=0
)
split_docs = text_splitter.split_documents(docs)
```

```
WARNING:langchain.text_splitter:Created a chunk of size 1003, which is longer than the specified 1000
```

```
print(map_reduce_chain.run(split_docs))
```

```
Based on the list of documents provided, the main themes can be identified as follows:

1. LLM-powered autonomous agents: The documents discuss the concept of building agents with LLM (large language model) as their core con

2. Agent system overview: The documents provide an overview of the components that make up a LLM-powered autonomous agent system. These

3. Planning: The documents discuss how the agent breaks down large tasks into smaller subgoals and utilizes self-reflection to improve t

4. Memory: The documents explain the importance of both short-term and long-term memory in the agent system. Short-term memory is utiliz

5. Tool use: The documents highlight the agent's ability to call external APIs for additional information and resources that may be miss
```

## Tagging

```
from langchain.chat_models import ChatOpenAI
from langchain.prompts import ChatPromptTemplate
from langchain.chains import create_tagging_chain, create_tagging_chain_pydantic
```

```
# Schema
schema = {
    "properties": {
        "sentiment": {"type": "string"},
        "aggressiveness": {"type": "integer"},
```

```
inp = "Estoy muy enojado con vos! Te voy a dar tu merecido!"
chain.run(inp)
```

```
{'sentiment': 'enojado', 'aggressiveness': 1, 'language': 'es'}
```

```
inp = "Don't ever talk to me!!"
chain.run(inp)
```

```
{'sentiment': 'negative', 'aggressiveness': 1}
```

```
schema = {
    "properties": {
        "aggressiveness": {
            "type": "integer",
            "enum": [1, 2, 3, 4, 5],
            "description": "describes how aggressive the statement is, the higher the number the more aggressive",
        },
        "language": {
            "type": "string",
            "enum": ["spanish", "english", "french", "german", "italian"],
        },
    },
    "required": ["language", "sentiment", "aggressiveness"],
}

chain = create_tagging_chain(schema, llm)

inp = "Estoy increiblemente contento de haberte conocido! Creo que seremos muy buenos amigos!"
chain.run(inp)
```

```
{'aggressiveness': 0, 'language': 'spanish'}
```

```
inp = "Estoy muy enojado con vos! Te voy a dar tu merecido!"
chain.run(inp)
```

```
{'aggressiveness': 5, 'language': 'spanish'}
```

```
inp = "Don't ever talk to me!!"
chain.run(inp)
```

```
{'aggressiveness': 5, 'language': 'english'}
```