**CHAPTER 5:**

**chain.py**

*import os*

*import time*

*from langchain.chains import MapReduceDocumentsChain, LLMChain, ReduceDocumentsChain, StuffDocumentsChain*

*from langchain.document\_loaders import NewsURLLoader*

*from langchain.llms import CTransformers*

*from langchain.prompts import PromptTemplate*

*from langchain.text\_splitter import RecursiveCharacterTextSplitter*

*def summarizer(article\_url):*

*# Load article*

*loader = NewsURLLoader([article\_url])*

*docs = loader.load()*

*# Load LLM*

*config = {'max\_new\_tokens': 4096, 'temperature': 0.3, 'context\_length': 4096}*

*llm = CTransformers(model="TheBloke/Mistral-7B-Instruct-v0.1-GGUF",*

*model\_file="mistral-7b-instruct-v0.1.Q4\_K\_M.gguf",*

*config=config,*

*threads=os.cpu\_count())*

*# Map template and chain*

*map\_template = """<s>[INST] The following is a part of an article from a website:*

*{docs}*

*Based on this, please identify the main theme and key points of it*

*Answer:  [/INST] </s>"""*

*map\_prompt = PromptTemplate.from\_template(map\_template)*

*map\_chain = LLMChain(llm=llm, prompt=map\_prompt)*

*# Reduce template and chain*

*reduce\_template = """<s>[INST] The following is set of summaries from the article:*

*{doc\_summaries}*

*Take these as a input and condense it into a final, consolidated summary of all the main points.*

*Construct it as a well organized summary of the main points and it should be between 2 and 6 paragraphs.*

*Answer:  [/INST] </s>"""*

*reduce\_prompt = PromptTemplate.from\_template(reduce\_template)*

*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="doc\_summaries"*

*)*

*# Combines and iteratively 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=True,*

*)*

*# Split documents into chunks*

*text\_splitter = RecursiveCharacterTextSplitter(*

*chunk\_size=4000, chunk\_overlap=100*

*)*

*split\_docs = text\_splitter.split\_documents(docs)*

*# Run the chain*

*start\_time = time.time()*

*result = map\_reduce\_chain.\_\_call\_\_(split\_docs, return\_only\_outputs=True)*

*time\_taken = time.time() - start\_time*

*return result['output\_text'], time\_taken*

**main.py**

*import streamlit as st*

*from PIL import Image*

*from chain import summarizer*

*def add\_logo(logo\_path, width, height):*

*logo = Image.open(logo\_path)*

*modified\_logo = logo.resize((width, height))*

*return modified\_logo*

*def main():*

*# Set page title*

*st.set\_page\_config(page\_title="Summarizer App", page\_icon="📜", layout="wide")*

*# Set title*

*st.title("Summarizer", anchor=False)*

*st.header("Summarize Articles using Mistral", anchor=False)*

*st.sidebar.image(add\_logo("C:/Users/srika/LLM/Summarization/logo.png", 250, 250))*

*st.sidebar.title("Navigation")*

*st.sidebar.markdown("- Home")*

*st.sidebar.markdown("- About")*

*st.sidebar.markdown("- Contact")*

*# Input URL*

*st.divider()*

*url = st.text\_input("Enter URL of any article", value="")*

*# Download audio*

*st.divider()*

*if url:*

*with st.status("Processing...", state="running", expanded=True) as status:*

*st.write("Summarizing Article...")*

*summary, time\_taken = summarizer(url)*

*status.update(label=f"Finished - Time Taken: {time\_taken} seconds", state="complete")*

*# Show Summary*

*st.subheader("Summary:", anchor=False)*

*st.write(summary)*

*if \_\_name\_\_=="\_\_main\_\_":*

*main()*