```
# Import Langchaine to google colab.
!pip install --quiet langchain
!pip install --quiet langchain-google-genai
!pip install --quiet chromadb
!pip install langchain-community
```

Show hidden output

```
# Import Google Key.
import os
import getpass
os.environ['GOOGLE_API_KEY'] = getpass.getpass('GeminiAPI_key')
```

Show hidden output

```
# import LangChain libraries
from langchain import PromptTemplate
from langchain import hub
from langchain.docstore.document import Document
from langchain.document_loaders import WebBaseLoader
from langchain.schema import StrOutputParser
from langchain.schema.prompt_template import format_document
from langchain.schema.runnable import RunnablePassthrough
from langchain.vectorstores import Chroma
```

Show hidden output

```
# Import the dataset from the web (Star Wars: The Last Jedi)
loader = WebBaseLoader("https://en.wikipedia.org/wiki/Star_Wars:_The_Last_Jedi")
docs = loader.load()


# Extract the text from the website data content.
text_content = docs[0].page_content

# To select the required content.
text_content_0 = text_content.split("==References==")[0]
text_content_1 = text_content_0.split("==External Links==")[0]
text_content_2 = text_content_1.split("==See also==")[0]
final_text = text_content_2.split("==Notes==")[0]
print(final_text)

# Convert the text to LangChain's to the document format.
docs = [Document(page_content=final_text, metadata={"source":"Star Wars: The Last Jedi"})]
```

```
Star Wars: The Last Jedi - Wikipedia
```

```
# Import Gemini chatbot.
from langchain_google_genai import GoogleGenerativeAIEmbeddings

gemini_embeddings = GoogleGenerativeAIEmbeddings(model="models/text-embedding-004")


# Save to disk.
VectorStore = Chroma.from_documents(
    documents=docs,
    embedding=gemini_embeddings,
    persist_directory="./chroma_db"
)


from re import search
# Load from disk.
VectorStore_disk = Chroma(
    persist_directory="./chroma_db",
    embedding_function=gemini_embeddings
)

retriever = VectorStore_disk.as_retriever(search_kwargs={"k": 1})
print(len(retriever.get_relevant_documents("MMLU")))
```

```
1
```

```
from langchain_google_genai import ChatGoogleGenerativeAI

llm = ChatGoogleGenerativeAI(model="gemini-pro", temparature=0.1, top_p=0.8, top_k=40)


from langchain_core.prompts import PromptTemplate

# Prompt template to query Gemini.
llm_prompt_template = """ You are an assistant for question-answering tasks.
Use the following pieces of retrieved context to answer the question.
If you don't know the answer, just say that you don't know, don't try to make up an answer.
Use three sentences maximum. Keep the answer as concise as possible.\n
Question: {question} \nContext: {context} \nAnswer:"""

# Use keyword arguments to initialize PromptTemplate
llm_prompt = PromptTemplate(template=llm_prompt_template, input_variables=["question", "context"])

print(llm_prompt)
```

```
input_variables=['context', 'question'] input_types={} partial_variables={} template=" You are an assistant for question-answering tasks
```

```
# Combine data from documents to readable string format.
def format_docs(docs):
  return "\n\n".join(doc.page_content for doc in docs)

rag_chain = (
    {"context": retriever | format_docs, "question": RunnablePassthrough()}
    | llm_prompt
    | llm
    | StrOutputParser()
)



# Prompt the model. I am ready to ask question to Gemini about Star Wars: The Last Jedi
rag_chain.invoke("What is the movie about?")
```

```
'The movie is about the resistance forces fighting against Kylo Ren and the First Order while General Leia Organa, Finn, and Poe Damero
```

```
rag_chain.invoke("What is the main character in the movie?")
```

```
rag_chain.invoke("Where was the movie shot?")
```

```
rag_chain.invoke("Who is the producer and the director of the movie?")
```

```
rag_chain.invoke("What was the rating of the movie?")
```

```
'The rating of the movie is 91% fresh on Rotten Tomatoes and 84 out of 100 on Metacritic.'
```

```
rag_chain.invoke("Did the movie have any special effects?")
```

```
'The movie did not have any special effects.'
```

```
rag_chain.invoke("What are the names of the characters of the movie?")
```

```
'1. Mark Hamill\n2. Carrie Fisher\n3. Adam Driver\n4. Daisy Ridley\n5. John Boyega\n6. Oscar Isaac\n7. Andy Serkis\n8. Lupita Nyong'o\n
9. Domhnall Gleeson\n10. Anthony Daniels\n11. Gwendoline Christie\n12. Kelly Marie Tran\n13. Laura Dern\n14. Benicio del Toro'
```