```
!pip install langchain sentence-transformers chromadb llama-cpp-python langc
```

Show hidden output

```
from google.colab import drive
drive. mount ("/content/drive")
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, ca
```

```
from langchain_community.document_loaders import PyPDFDirectoryLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain_community.embeddings import SentenceTransformerEmbeddings
from langchain.vectorstores import Chroma
from langchain_community.llms import LlamaCpp
from langchain.chains import RetrievalQA, LLMChain
```

```
loader = PyPDFDirectoryLoader("/content/drive/MyDrive/Colab Notebooks")
docs = loader.load()
```

```
len(docs)
```

```
95
```

```
text_splitter = RecursiveCharacterTextSplitter(chunk_size=300, chunk_overlap
chunks = text_splitter.split_documents (docs)
```

```
len(chunks)
```

```
585
```

```
import os
```

```
os. environ ['HUGGINGFACEHUB_API_TOKEN' ] = "" #Enter your API key
```

```
embedding=SentenceTransformerEmbeddings (model_name="NeuML/pubmedbert-base-
```

Show hidden output

```
vectorstore = Chroma.from_documents(chunks,embedding)
```

```
query = "Who is at risk of heart disease?"
search_results = vectorstore.similarity_search(query)
```

```
search_results
```

Show hidden output

```
retriever = vectorstore.as_retriever(search_kwargs={'k':5})
retriever.get_relevant_documents (query)
```

Show hidden output

```
llm = LlamaCpp (
  model_path="/content/drive/MyDrive/BioMistral-7B.Q2_K.gguf",
  temperature=0.2,
  max_tokens = 2048,
  top_p=1
)
```

```
load_tensors: layer  28 assigned to device CPU
load_tensors: layer  29 assigned to device CPU
load_tensors: layer  30 assigned to device CPU
load_tensors: layer  31 assigned to device CPU
load_tensors: layer  32 assigned to device CPU
load_tensors: tensor 'token_embd.weight' (q2_K) (and 290 others) cannot be
load_tensors:   CPU_Mapped model buffer size =  2592.57 MiB
llama_init_from_model: n_batch is less than GGML_KQ_MASK_PAD - increasing t
llama_init_from_model: n_seq_max     = 1
llama_init_from_model: n_ctx         = 512
llama_init_from_model: n_ctx_per_seq = 512
llama_init_from_model: n_batch       = 32
llama_init_from_model: n_ubatch      = 8
llama_init_from_model: flash_attn    = 0
llama_init_from_model: freq_base     = 10000.0
llama_init_from_model: freq_scale    = 1
llama_init_from_model: n_ctx_per_seq (512) < n_ctx_train (32768) -- the ful
llama_kv_cache_init: kv_size = 512, offload = 1, type_k = 'f16', type_v = '
llama_kv_cache_init: layer 0: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
llama_kv_cache_init: layer 1: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
llama_kv_cache_init: layer 2: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
llama_kv_cache_init: layer 3: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
llama_kv_cache_init: layer 4: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
llama_kv_cache_init: layer 5: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
llama_kv_cache_init: layer 6: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
llama_kv_cache_init: layer 7: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
llama_kv_cache_init: layer 8: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
```

```
 llama_kv_cache_init: layer 9: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 10: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 11: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 12: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 13: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 14: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 15: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 16: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 17: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 18: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 19: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 20: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 21: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 22: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 23: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 24: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 25: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 26: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 27: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 28: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 29: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 30: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init: layer 31: n_embd_k_gqa = 1024, n_embd_v_gqa = 1024
 llama_kv_cache_init:        CPU KV buffer size =    64.00 MiB
 llama_init_from_model: KV self size  =   64.00 MiB, K (f16):   32.00 MiB, V
 llama_init_from_model:        CPU  output buffer size =     0.12 MiB
 llama_init_from_model:        CPU compute buffer size =     1.31 MiB
 llama_init_from_model: graph nodes  = 1030
 llama_init_from_model: graph splits = 1
CPU : SSE3 = 1 | SSSE3 = 1 | AVX = 1 | AVX2 = 1 | F16C = 1 | FMA = 1 | LLAM
Model metadata: {'tokenizer.chat_template': "{{ bos_token }}{% for message
Available chat formats from metadata: chat_template.default
Guessed chat format: mistral-instruct
```

```
template = """
<|context|>
You are an Medical Assistant that follows the instructions and generate the
response based
on the query and
the
context provided.
Please be truthful and give direct answers.
</S>
<|user |>
{query}
</s>
<|assistant |>
"""
```

```
from langchain.schema. runnable import RunnablePassthrough
from langchain.schema.output_parser import StrOutputParser
from langchain.prompts import ChatPromptTemplate


prompt = ChatPromptTemplate.from_template(template)


rag_chain = (
  {"context": retriever, "query": RunnablePassthrough()}
  |prompt
  | llm
  | StrOutputParser()
)
```

```
response=rag_chain.invoke(query)
```

```
llama_perf_context_print:        load time =   29148.87 ms
llama_perf_context_print: prompt eval time =   29148.73 ms /    78 tokens (
llama_perf_context_print:        eval time =   33531.99 ms /    68 runs   (
llama_perf_context_print:       total time =   62754.14 ms /   146 tokens
```

```
response
```

```
'The risk of heart disease is higher in men than women. The risk increases
with age and is highest in people over 65 years old. People who are overwe
ight or obese have a greater risk of heart disease. Smoking, high blood pr
```

```
import sys
while True:
  user_input = input (f"Input query: ")
  if user_input == 'exit':
    print ("Exiting...")
    sys.exit()
  if user_input=="":
    continue
  result = rag_chain. invoke(user_input)
  print("Answer: ", result)
```

```
Input query: What are the diseases that affect heart health?
Llama.generate: 59 prefix-match hit, remaining 20 prompt tokens to eval
llama_perf_context_print:        load time =   29148.87 ms
llama_perf_context_print: prompt eval time =    6213.97 ms /    20 tokens (
llama_perf_context_print:        eval time =   22797.68 ms /    44 runs   (
llama_perf_context_print:       total time =   29063.81 ms /    64 tokens
Answer:  The heart is a vital organ that pumps blood throughout the body. I
Input query: What are the diseases that affect heart health?
Llama.generate: 78 prefix-match hit, remaining 1 prompt tokens to eval
llama_perf_context_print:        load time =   29148.87 ms
llama_perf_context_print: prompt eval time =       0.00 ms /     1 tokens (
llama_perf_context_print:        eval time =   62640.66 ms /   127 runs   (
llama_perf_context_print:       total time =   62783.49 ms /   128 tokens
Answer:  The heart is a vital organ that pumps blood throughout the body. T
Input query: preventive measures from COVID 19
Llama.generate: 59 prefix-match hit, remaining 19 prompt tokens to eval
llama_perf_context_print:        load time =   29148.87 ms
llama_perf_context_print: prompt eval time =    6261.06 ms /    19 tokens (
llama_perf_context_print:        eval time =   76712.78 ms /   155 runs   (
llama_perf_context_print:       total time =   83165.13 ms /   174 tokens
Answer:  The best way to prevent COVID-19 is by following the instructions
Input query: exit
Exiting...
An exception has occurred, use %tb to see the full traceback.

SystemExit

/usr/local/lib/python3.11/dist-packages/IPython/core/interactiveshell.py:35
  warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```