

```
!pip install -U langchain langchain-community langchain-huggingface faiss-cpu tra
!pip install -U langchain-huggingface
```

```

⇒ Requirement already satisfied: langchain in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: langchain-community in /usr/local/lib/python3
Requirement already satisfied: langchain-huggingface in /usr/local/lib/pytho
Requirement already satisfied: faiss-cpu in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: transformers in /usr/local/lib/python3.11/dis
Requirement already satisfied: accelerate in /usr/local/lib/python3.11/dist-
Requirement already satisfied: PyMuPDF in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: langchain-core<1.0.0,>=0.3.66 in /usr/local/l
Requirement already satisfied: langchain-text-splitters<1.0.0,>=0.3.8 in /us
Requirement already satisfied: langsmith>=0.1.17 in /usr/local/lib/python3.1
Requirement already satisfied: pydantic<3.0.0,>=2.7.4 in /usr/local/lib/pyth
Requirement already satisfied: SQLAlchemy<3,>=1.4 in /usr/local/lib/python3.
Requirement already satisfied: requests<3,>=2 in /usr/local/lib/python3.11/d
Requirement already satisfied: PyYAML>=5.3 in /usr/local/lib/python3.11/dist
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in /usr/local/lib/pytho
Requirement already satisfied: tenacity!=8.4.0,<10,>=8.1.0 in /usr/local/lib
Requirement already satisfied: dataclasses-json<0.7,>=0.5.7 in /usr/local/li
Requirement already satisfied: pydantic-settings<3.0.0,>=2.4.0 in /usr/local
Requirement already satisfied: httpx-sse<1.0.0,>=0.4.0 in /usr/local/lib/pyt
Requirement already satisfied: numpy>=1.26.2 in /usr/local/lib/python3.11/di
Requirement already satisfied: tokenizers>=0.19.1 in /usr/local/lib/python3.
Requirement already satisfied: huggingface-hub>=0.30.2 in /usr/local/lib/pyt
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.1
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: psutil in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: torch>=2.0.0 in /usr/local/lib/python3.11/dis
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/pyt
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.11
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/di
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.1
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.1
Requirement already satisfied: marshmallow<4.0.0,>=3.18.0 in /usr/local/lib/
Requirement already satisfied: typing-inspect<1,>=0.4.0 in /usr/local/lib/py
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.11
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/
Requirement already satisfied: hf-xet<2.0.0,>=1.1.2 in /usr/local/lib/python
Requirement already satisfied: jsonpatch<2.0,>=1.33 in /usr/local/lib/python
Requirement already satisfied: httpx<1,>=0.23.0 in /usr/local/lib/python3.11
Requirement already satisfied: orjson<4.0.0,>=3.9.14 in /usr/local/lib/pytho
Requirement already satisfied: requests-toolbelt<2.0.0,>=1.0.0 in /usr/local

```

```

Requirement already satisfied: zstandard<0.24.0,>=0.23.0 in /usr/local/lib/p
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/pyth
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/pytho
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/py
Requirement already satisfied: python-dotenv>=0.21.0 in /usr/local/lib/pytho
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/py
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dis
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.
Requirement already satisfied: greenlet>=1 in /usr/local/lib/python3.11/dist
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.4.127 in /usr/loca

```

Load and parse PDFs using PyMuPDF via LangChain

```

from langchain_community.document_loaders import PyMuPDFLoader
import os

all_documents = []

for file_name in os.listdir("papers"):
    if file_name.endswith(".pdf"):
        path = os.path.join("papers", file_name)
        loader = PyMuPDFLoader(path)
        docs = loader.load()
        all_documents.extend(docs)


print(f"Loaded {len(all_documents)} text chunks from PDFs.")

from langchain.text_splitter import CharacterTextSplitter

# Split large chunks into smaller pieces
text_splitter = CharacterTextSplitter(chunk_size=400, chunk_overlap=50)
split_documents = text_splitter.split_documents(all_documents)


print(f"Split into {len(split_documents)} chunks")

```

 Loaded 2 text chunks from PDFs.
 Split into 2 chunks

```
# Embed documents using HuggingFace MiniLM
from langchain_huggingface import HuggingFaceEmbeddings
from langchain.vectorstores import FAISS

embedding_model = HuggingFaceEmbeddings(model_name="all-MiniLM-L6-v2")
db = FAISS.from_documents(split_documents, embedding_model)
```

 /usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: Use
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access private repositories.
warnings.warn(

```
# Load a Hugging Face LLM (Flan-T5)
from langchain_huggingface import HuggingFacePipeline
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM, pipeline

model_id = "google/flan-t5-base"

tokenizer = AutoTokenizer.from_pretrained(model_id)
model = AutoModelForSeq2SeqLM.from_pretrained(model_id)

pipe = pipeline(
    "text2text-generation",
    model=model,
    tokenizer=tokenizer,
    max_length=512,
    truncation=True
)
llm = HuggingFacePipeline(pipeline=pipe)
```

 Device set to use cpu

```
from langchain.prompts import PromptTemplate

prompt_template = PromptTemplate(
    input_variables=["context", "question"],
    template="question: {question}\ncontext: {context}"
)

from langchain.chains import RetrievalQA

qa_chain = RetrievalQA.from_chain_type(
    llm=llm,
    retriever=db.as_retriever(search_kwargs={"k": 1}),
    chain_type="stuff",
    chain_type_kwargs={"prompt": prompt_template}
)

# Ask your question!
query = "explain in 100 words What was different about the public's attitude toward
response = qa_chain.invoke(query)

print("Answer:")
print(response['result'])
```

➡ Answer:
Rather than targeting doctors for potentially spreading SARS-CoV-2 or for fal

