RAG_chat_bot

December 16, 2025

[22]:
```
from langchain_community.document_loaders import WebBaseLoader
from langchain_text_splitters import RecursiveCharacterTextSplitter
from langchain_huggingface import HuggingFaceEmbeddings
from langchain_community.vectorstores import Chroma

from langchain_groq import ChatGroq
from langchain_core.runnables import RunnablePassthrough
from langchain_core.output_parsers import StrOutputParser
from langchain import hub
```

[24]:
```
from langchain_community.document_loaders import WebBaseLoader

loader = WebBaseLoader(
    web_paths=["https://artelus.com/"]
)

docs = loader.load()
print(docs[0].page_content[:500])
```

```
Artelus - Empowering Healthcare with Artificial
IntelligenceHomeAboutProductsAwardsPublicationsMediaCareersContactUsing
Artificial Intelligence to prevent Preventable BlindnessBook a Call Now
WHAT IS DRISTiDRISTiDiabetic Retinopathy Screening (DRISTi) (CE Class 1) is an
AI product designed to detect the early presence of Diabetic Retinopathy (DR) in
patients during eye check up screening process instantaneously. By cutting the
cord, and creating a first of its kind, AI on a Chip we have
```

[25]:
```
from langchain.text_splitter import RecursiveCharacterTextSplitter

text_splitter = RecursiveCharacterTextSplitter(
    chunk_size=1000,
    chunk_overlap=200
)

splits = text_splitter.split_documents(docs)
print(len(splits))
```

```
3
```

[26]:
```
from langchain_huggingface import HuggingFaceEmbeddings
from langchain_community.vectorstores import Chroma


embeddings = HuggingFaceEmbeddings(
    model_name="sentence-transformers/all-MiniLM-L6-v2"
)

vectorstore = Chroma.from_documents(
    documents=splits,
    embedding=embeddings,
    persist_directory="./chromaDB",
    collection_name="mental_health"
)

vectorstore.persist()

print("Stored successfully in Chroma")
```

```
Stored successfully in Chroma
```

```
/var/folders/wj/ctyq3nfn137dfmg143h05mr80000gn/T/ipykernel_20798/1141385086.py:1
6: LangChainDeprecationWarning: Since Chroma 0.4.x the manual persistence method
is no longer supported as docs are automatically persisted.
  vectorstore.persist()
```

[27]:
```
retriever = vectorstore.as_retriever()
```

[28]:
```
from langchain import hub

prompt = hub.pull("rlm/rag-prompt")
```

[36]:
```
from langchain_groq import ChatGroq
#from dotenv import load_dotenv
#import os

#load_dotenv()

llm = ChatGroq(
    groq_api_key="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",
    model_name="llama-3.1-8b-instant"
)
print("Testing LLM")
response = llm.invoke("What is Retrieval Augmented Generation?")
print(response.content)
```

```
Testing LLM
Retrieval Augmented Generation (RAG) is a type of AI model architecture that
combines the strengths of both retrieval-based and generative models. It was
```

first introduced in the paper "Retrieval-Augmented Generation for Knowledge-Intensive Tasks" by Luan et al. in 2021.

In traditional generative models, AI systems are trained to generate text or other outputs from scratch, using only the input prompt or context. However, this approach can be limited by the model's ability to recall specific knowledge or details from its training data.

Retrieval-Augmented Generation addresses this limitation by incorporating a retrieval component into the model architecture. The retrieval component is trained to search for relevant information in a knowledge base or a large corpus of text, and then uses this information to augment the generated output.

The RAG model architecture typically consists of two main components:

1. **Retrieval component**: This component is trained to retrieve relevant information from a knowledge base or a large corpus of text. This can be done using techniques such as dense retrieval or sparse retrieval, depending on the specific implementation.
2. **Generative component**: This component is trained to generate text or other outputs from scratch, using the input prompt or context, as well as the retrieved information from the retrieval component.

When a user inputs a prompt or context, the retrieval component searches for relevant information in the knowledge base or corpus, and returns a set of relevant documents or passages. The generative component then uses this retrieved information to generate a final output, which is a combination of the input prompt or context, and the relevant information retrieved from the knowledge base or corpus.

RAG models have been shown to outperform traditional generative models on a range of knowledge-intensive tasks, such as question answering, text summarization, and conversational dialogue. This is because the retrieval component can provide the model with access to a vast amount of relevant information, which can be used to generate more accurate and informative outputs.

Overall, Retrieval-Augmented Generation is an exciting development in the field of natural language processing, and has the potential to enable AI systems to perform a wide range of knowledge-intensive tasks with greater accuracy and effectiveness.

[30]:
```python
from langchain_core.runnables import RunnablePassthrough
from langchain_core.output_parsers import StrOutputParser

def format_docs(docs):
    return "\n".join(doc.page_content for doc in docs)
```

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

[31]:
```
rag_chain.invoke("about Artelus")
```

[31]:
```
'Artelus is a company that uses artificial intelligence to empower healthcare by
preventing preventable blindness through early detection of Diabetic Retinopathy
(DR). Their AI product, DRISTi, is a CE Class 1 device that can detect early
signs of DR in patients during eye check-ups instantaneously. Artelus aims to
bring point-of-care diagnostics to remote areas of the globe, where access to
healthcare is limited.'
```

[34]:
```
rag_chain.invoke("Capital of India")
```

[34]:
```
"I don't know. \n\nThe provided context is about Artelus, an AI-based healthcare
company, and its products for detecting diabetic retinopathy and other retinal
diseases. \n\nThe information about the capital of India is not present in the
given context."
```

[35]:
```
rag_chain.invoke("Mention all the products developed by Artelus")
```

[35]:
```
'Artelus develops the following product: \n\n1. DRISTi (Diabetic Retinopathy
Screening) - an AI product designed to detect Diabetic Retinopathy in patients
during eye check-up screening process. \n2. Fundus AI - a system that can
capture 4 levels of Diabetic Retinopathy (DR) and Diabetic Macular Edema (DME).
\n3. OCT AI - a system that accurately identifies 7 retinal diseases.'
```