Value-Based Healthcare (VBHC) in Breast Reconstruction Surgery

AI in Healthcare - High Risk Project Submission

presented by

Suryanarayana Jayanthi, Alphonse Chander

The University of Texas at Austin

VBC context for Breast reconstruction surgery

VBC in nutshell

- U.S. healthcare spending accounted for **17.3%** of the nation's Gross Domestic Product (GDP) in 2022, significantly higher than other high-income countries. (Source: Centers for Medicare & Medicaid Services - CMS). Value Based Care (VBC) is transforming US health care industry to reduce cost and improve patient outcomes.

Here are some key aspects of VBC:

- **Shift in Payment Incentives:** FFS rewards the *volume* of services provided (more tests, visits, procedures = more payment), whereas VBC links payment to the *quality* and *outcomes* of care, incentivizing providers to keep patients healthy and manage costs effectively.
- **Focus on Population Health & Coordination:** VBC encourages a proactive approach, focusing on preventative care, managing chronic conditions for entire patient populations, and coordinating care across different providers and settings, rather than the reactive, fragmented care often seen in FFS.
- **Alignment of Risk and Goals:** VBC models often involve providers sharing financial risk and reward based on achieving specific quality metrics and cost targets, aiming to align provider incentives with the goals of improving patient health and reducing overall healthcare expenditures.

VBC context for our project breast reconstruction surgery

While breast reconstruction surgery improves outcomes of the patients, a lot of information about cost and quality are not readily available. Our project focus is how to use LLM so we can understand the cost related information better for breast reconstruction surgery that can help create better value based care plans and payments.

Problem Statement & Methodology Overview

Problem Statement

Breast reconstruction improves quality of life after mastectomy, yet synthesizing cost and clinical evidence for Value Based Care surgical decision-making remains challenging due to:

- High volume of unstructured, and primarily available as PDFs on platforms like PubMed
- Variability in outcome measures (cost, complications, **BREAST-Q**, QALY)
- Time consuming manual reviews that **lack scalability**

Objective

To develop a question answer system and evaluate a **Retrieval-Augmented Generation (RAG)** pipeline using **GPT-4** to automate summarization of clinical literature.
The system extracts structured insights from breast reconstruction studies, focusing on:

Complications

Cost-effectiveness (e.g., ICER, QALY)

Patient satisfaction (BREAST-Q)

Methodology

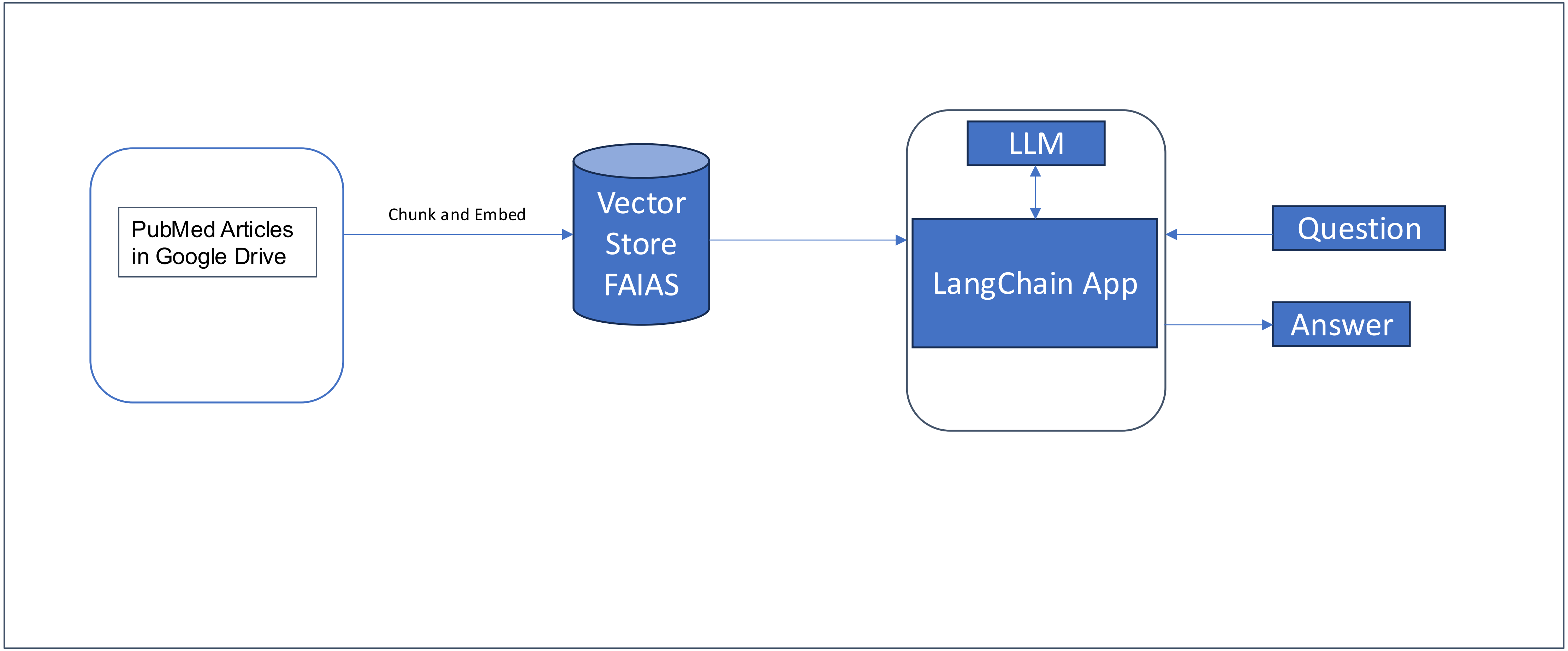
Pipeline with LLMs with RAG pattern

- To leverage the **semantic understanding and reasoning** abilities of large language models to interpret medical evidence and support **personalized, value-based surgical decisions** through:
- Accurate, explainable answers to clinical questions
- Context-aware synthesis from peer-reviewed full-text studies

Pipeline Summary

- **PDF Ingestion (PubMed) -> Text Chunking -> Embedding (OpenAI) -> Vector Indexing (FAISS) -> Clinical Query Input -> Retrieval (Top-K Chunks) -> GPT-4 Summarization -> Evidence-based Outp**

High level solution design



Data Preparation

This project introduces a **Retrieval-Augmented Generation (RAG) pipeline** to automate literature summarization for breast reconstruction outcomes, using **PubMed articles** as input.

GOAL

- Ingest and preprocess clinical PDFs covering breast reconstruction techniques (e.g., DIEP, TRAM, implants)
- Segment documents into context-preserving chunks for retrieval
- Embed and index these chunks in a searchable vector store
- Generate evidence-based summaries using GPT-4 for key clinical questions

PIPELINE

Integrates the following components:

- **PubMed PDFs** sourced manually from systematic reviews and cost-effectiveness studies
- **PyPDF and LangChain splitters** for parsing and chunking full-text clinical literature
- **OpenAI Embeddings + FAISS** for semantic indexing and document retrieval
- **GPT-4 via LangChain** for summarizing cost, complications, and patient-reported outcomes (e.g., BREAST-Q, QALYs)

.

A strong emphasis is placed on **accuracy, transparency**, and the ability to **update insights dynamically** as new literature becomes available

PubMed Dataset Overview for Breast Reconstruction Summarization

Data Source:

Peer-reviewed PDFs from **PubMed-indexed articles** related to breast reconstruction

Types of Studies Used:

- **Cost-effectiveness analyses** (e.g., ICER, QALY comparisons between DIEP, TRAM, and implant-based reconstruction)
- **Complication profiles** by flap type or implant technique
- **BREAST-Q patient-reported outcomes** across surgical pathways
- **Length-of-stay impact** on costs and satisfaction
- **Systematic reviews** and multi-institutional cohort studies

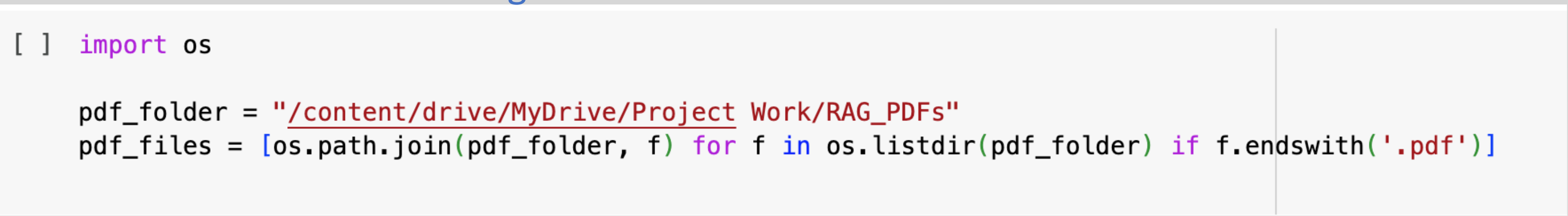
Technology Use Cases

Chunk-based **text segmentation and embedding**

Evidence-grounded **question answering** using vector similarity search

Clinical query evaluation of surgical decision factors

Code that lists PDF names ingested in the Drive



```
[ ]  import os

     pdf_folder = "/content/drive/MyDrive/Project Work/RAG_PDFs"
     pdf_files = [os.path.join(pdf_folder, f) for f in os.listdir(pdf_folder) if f.endswith('.pdf')]
```

PDF Ingestion & Preprocessing Pipeline

Ingestion of PubMed Docs

- Loads **real clinical studies** (systematic reviews, cost analyses, PROM studies) in PDF format
- Parses PDFs and splits into overlapping chunks using **LangChain text splitter**
- Prepares chunks for **semantic search** using OpenAI embeddings
- Runs efficiently in **Google Colab**, leveraging local or cloud storage (e.g., Google Drive)

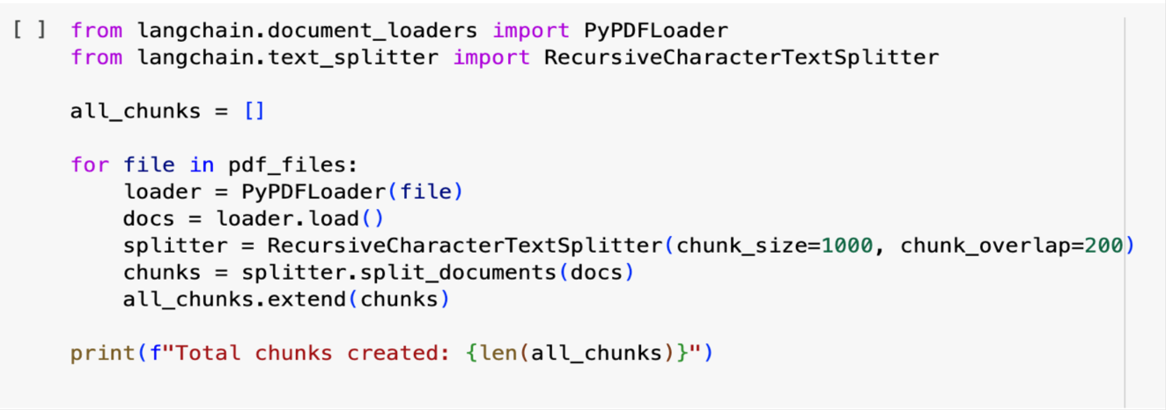
Chunk-based **text segmentation and embedding**

Preprocessing Pipeline

Enable downstream tasks such as:

- **Structured summarization** of surgical outcomes, complications, and cost metrics
- **Clinical question answering** using top-K chunk retrieval
- **Exploration of outcomes across techniques** (e.g., DIEP vs TRAM vs implant-based

Code



```
[ ] from langchain.document_loaders import PyPDFLoader
    from langchain.text_splitter import RecursiveCharacterTextSplitter

    all_chunks = []

    for file in pdf_files:
        loader = PyPDFLoader(file)
        docs = loader.load()
        splitter = RecursiveCharacterTextSplitter(chunk_size=1000, chunk_overlap=200)
        chunks = splitter.split_documents(docs)
        all_chunks.extend(chunks)

    print(f"Total chunks created: {len(all_chunks)}")
```

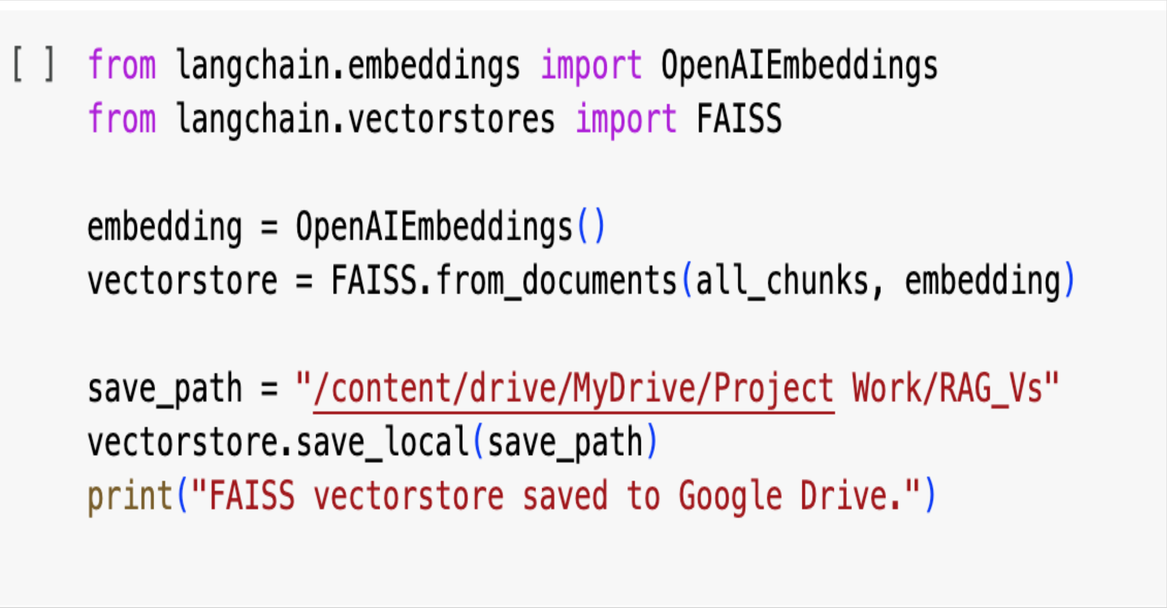
Vectorizing PubMed Chunks for Surgical Outcomes Summarization

This preprocessing step converts parsed **PubMed PDF chunks** into embeddings using **OpenAI** and stores them in a **FAISS vectorstore**, enabling semantic retrieval for downstream clinical question answering.

Features

- Parses and chunks full-text PDFs related to:
    - DIEP vs TRAM flaps
    - Pre-pectoral vs sub-pectoral implants
    - Cost-utility and BREAST-Q outcomes
- Embeds chunks using **OpenAIEmbedding API**
- Saves vectorized representations in **FAISS format**
- Stores the searchable index to **Google Drive** for persistence and later use

Code



```python
from langchain.embeddings import OpenAIEmbeddings
from langchain.vectorstores import FAISS

embedding = OpenAIEmbeddings()
vectorstore = FAISS.from_documents(all_chunks, embedding)

save_path = "/content/drive/MyDrive/Project Work/RAG_Vs"
vectorstore.save_local(save_path)
print("FAISS vectorstore saved to Google Drive.")
```

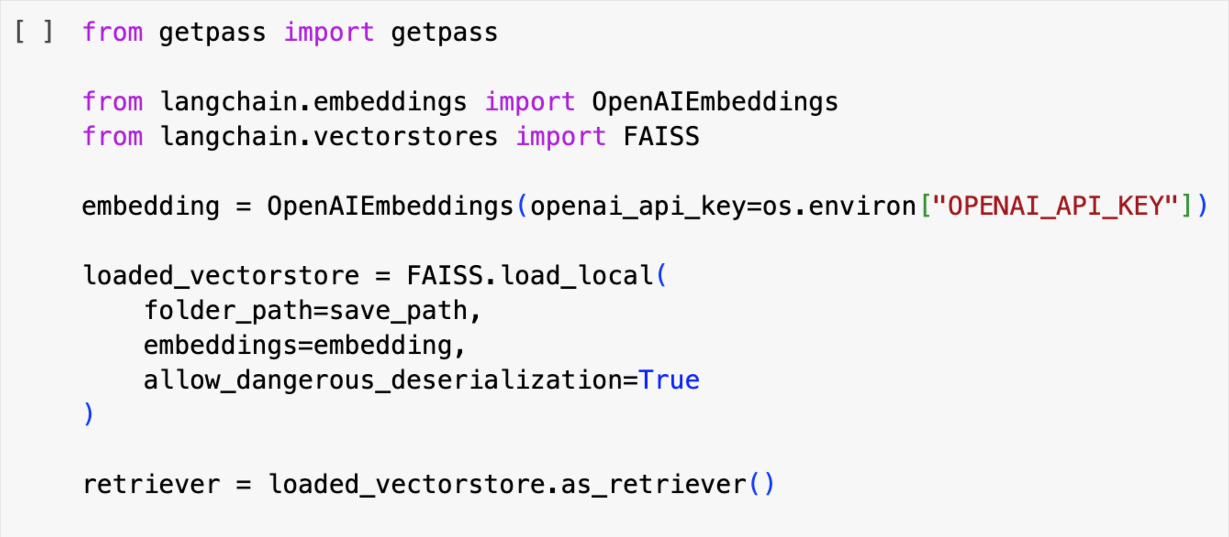
Query Construction: Connecting Vector store to GPT-4

This step loads the FAISS vectorstore containing embedded PubMed chunks and connects it to GPT-4 using **LangChain's RetrievalQA** for clinical summarization.

Steps

- Loading the saved **FAISS vector index** from Google Drive
- Initializing with a secure **OpenAI API key**
- Wrapping the vectorstore into a **retriever object**
- Executing structured queries (e.g., cost, complications, BREAST-Q outcomes) through GPT-4 with context-aware chunking

Code



```
[ ]  from getpass import getpass

     from langchain.embeddings import OpenAIEmbeddings
     from langchain.vectorstores import FAISS

     embedding = OpenAIEmbeddings(openai_api_key=os.environ["OPENAI_API_KEY"])

     loaded_vectorstore = FAISS.load_local(
         folder_path=save_path,
         embeddings=embedding,
         allow_dangerous_deserialization=True
     )

     retriever = loaded_vectorstore.as_retriever()
```

Query Execution Pipeline: Evidence-Grounded QA with GPT-4 and FAISS

Connecting Retrieval to Generation

This step connects semantic retrieval from FAISS with GPT-4-based generation using LangChain’s RetrievalQA interface and a custom prompt template. The system enables structured clinical questions to be answered using embedded chunks from full-text biomedical literature.

Features

- **Custom Prompt Design:** Instructs GPT-4 to base its answers strictly on retrieved text and to note uncertainty when evidence is incomplete.
- **Retriever Integration:** Uses a FAISS vector store to semantically fetch the most relevant text chunks for each query.
- **Model Configuration:** Leverages OpenAI’s gpt-4 via ChatOpenAI,

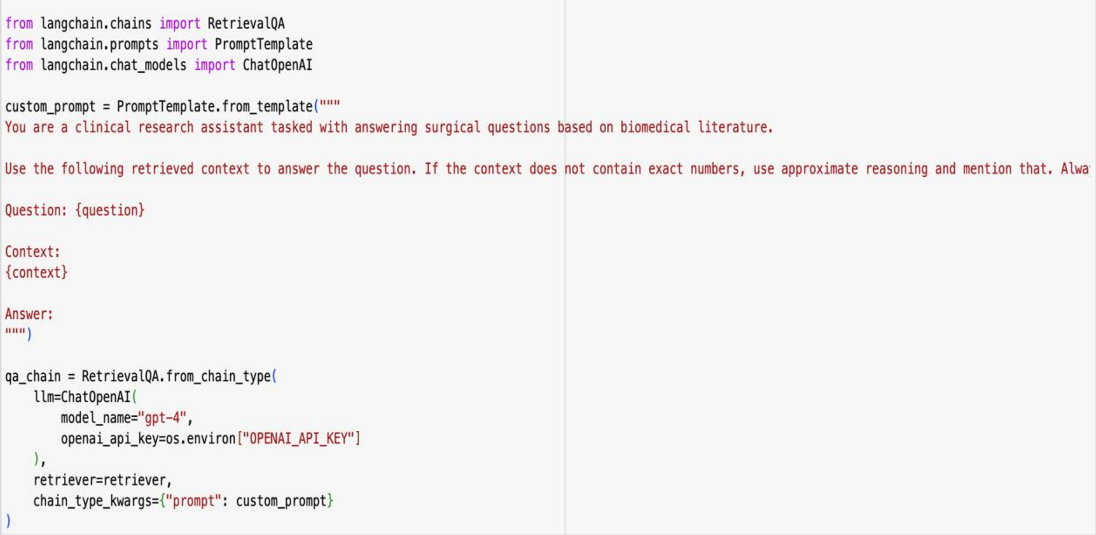
secured with an API key and deployed with LangChain.

- **Automated Response Generation:** Returns concise,

evidence-linked summaries from multi-document sources

with embedded citations where available.

Code



```
from langchain.chains import RetrievalQA
from langchain.prompts import PromptTemplate
from langchain.chat_models import ChatOpenAI

custom_prompt = PromptTemplate.from_template("""
You are a clinical research assistant tasked with answering surgical questions based on biomedical literature.

Use the following retrieved context to answer the question. If the context does not contain exact numbers, use approximate reasoning and mention that. Alwa

Question: {question}

Context:
{context}

Answer:
""")

qa_chain = RetrievalQA.from_chain_type(
    llm=ChatOpenAI(
        model_name="gpt-4",
        openai_api_key=os.environ["OPENAI_API_KEY"]
    ),
    retriever=retriever,
    chain_type_kwargs={"prompt": custom_prompt}
)
```

This pipeline enables GPT-4 to synthesize **multi-study clinical insights** – supporting comparative questions about surgical techniques, complications, patient satisfaction, or cost-effectiveness – without requiring clinicians to read every source.

Results – Examples for query

"How do patient-reported outcomes compare between implant-based and autologous (tissue-based) breast reconstruction?"

- According to the systematic review, patient-reported outcomes were generally higher for autologous (tissue-based) breast reconstruction compared to implant-based reconstruction. Using the BREAST-Q validated measurement tool, patients who underwent autologous reconstruction reported higher satisfaction with their breasts and greater psychosocial well-being than those who underwent implant-based reconstruction. Differences in physical well-being between the two groups were less significant and the least significant difference was noted for sexual well-being. The EORTC-QLQ-BR23/C30 PROMs also noted similar trends. The SF-36 measure, however, noted virtually no difference between the two methods of reconstruction regarding similar quality of life domains. Therefore, from the patient perspective, autologous reconstruction is either equal to or superior to implant-based reconstruction. The context does not provide exact numbers for these outcomes.

Clinical Query Execution: Sample Responses for Breast Reconstruction Questions

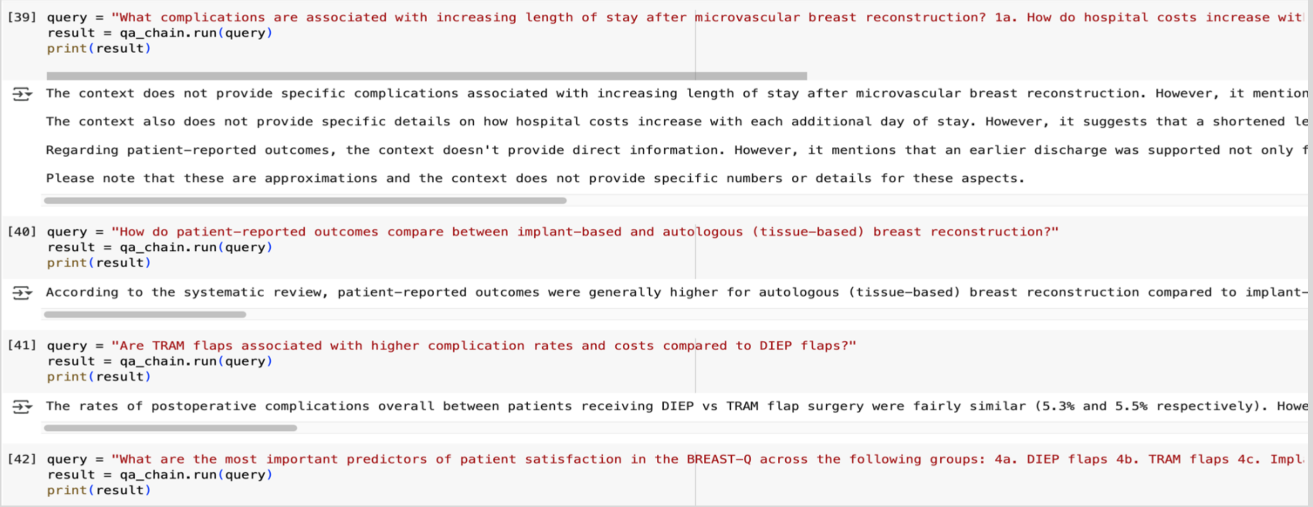
Running Structured Prompts Through GPT-4

This step showcases examples of clinical questions posed to the system and how GPT-4 responds using evidence retrieved from embedded PubMed articles.

Steps

- Structured question formulated by clinician or researcher
- Top-K relevant document chunks retrieved from FAISS
- GPT-4 summarizes answer using in-context clinical evidence
- Responses printed or logged for downstream use

Code



```
[39] query = "What complications are associated with increasing length of stay after microvascular breast reconstruction? 1a. How do hospital costs increase wit
     result = qa_chain.run(query)
     print(result)

     The context does not provide specific complications associated with increasing length of stay after microvascular breast reconstruction. However, it mention

     The context also does not provide specific details on how hospital costs increase with each additional day of stay. However, it suggests that a shortened le

     Regarding patient-reported outcomes, the context doesn't provide direct information. However, it mentions that an earlier discharge was supported not only f

     Please note that these are approximations and the context does not provide specific numbers or details for these aspects.

[40] query = "How do patient-reported outcomes compare between implant-based and autologous (tissue-based) breast reconstruction?"
     result = qa_chain.run(query)
     print(result)

     According to the systematic review, patient-reported outcomes were generally higher for autologous (tissue-based) breast reconstruction compared to implant-

[41] query = "Are TRAM flaps associated with higher complication rates and costs compared to DIEP flaps?"
     result = qa_chain.run(query)
     print(result)

     The rates of postoperative complications overall between patients receiving DIEP vs TRAM flap surgery were fairly similar (5.3% and 5.5% respectively). Howe

[42] query = "What are the most important predictors of patient satisfaction in the BREAST-Q across the following groups: 4a. DIEP flaps 4b. TRAM flaps 4c. Impl
     result = qa_chain.run(query)
     print(result)
```

These real-world prompts and responses form the **basis for evaluating the pipeline's clinical utility**, allowing transparent, grounded summaries tailored to surgical planning.

Conclusion

This project presents a **scalable and transparent AI pipeline** for summarizing clinical literature related to breast reconstruction outcomes using **Retrieval-Augmented Generation (RAG)** and GPT-4.

Key Contributions

- Developed an end-to-end pipeline from: **PubMed PDFs -> Chunking -> Embedding -> Vector Search -> GPT-4 Summarization**
- The system answers structured clinical queries regarding:
- **Cost-effectiveness** (e.g., ICERs, QALYs)
- **Complications** by technique (e.g., TRAM vs DIEP)
- **Patient-reported outcomes** (e.g., BREAST-Q domains)
- Pipeline integrates **LangChain**, **OpenAI embeddings**, and **FAISS** for efficient retrieval

Impact & Future Directions:

This pipeline serves as a **lightweight, real-time alternative to manual systematic reviews** for reconstructive surgery literature.

Next steps

- Scaling ingestion to **hundreds of PubMed articles**
- Extending to domains like **ADM use**, **radiation timing**, and **immediate vs delayed reconstruction**
- Deploying via a **clinician-facing interface** for on-demand evidence lookup

Resources & Implementation

This project leverages **full-text PubMed articles** related to breast reconstruction, processed into a structured format for **RAG-based summarization** using GPT-4. Implementation was conducted in **Google Colab** with **OpenAI's APIs**, and data/artifact storage handled via **Google Drive**.

PubMed Dataset

Relevant articles (n = 12) were sourced from:

- **Systematic reviews**
- **Cost-effectiveness analyses**
- **Complication rate comparisons**
- **BREAST-Q patient-reported outcomes studies**
- All PDFs were uploaded to a shared **Google Drive folder** and parsed using LangChain.

Notebook Implementation

Run the model in Colab