Example Case: Financial Statements Analysis Using LLMs Powered by RAG (Retrieval-Augmented Generation)

1. **Scenario:**
   Assume you are a financial analyst tasked with analyzing financial statements. The data is stored in diverse formats like PDFs, Excel sheets, or other document types.
2. **Challenge:**
   Traditional methods involve manual extraction, spreadsheet analysis, and complex formula-driven insights. A natural language-driven approach is preferred to make the process more intuitive and efficient.
3. **Solution: LLMs with Retrieval-Augmented Generation (RAG):**
   - **Data Ingestion:**
     Use OCR tools or specialized parsers to extract relevant data from documents like PDFs or scanned images.
   - **Knowledge Base Creation:**
     Store structured data in a vector database after embedding it using techniques like sentence transformers or similar models.
   - **Query and Analysis:**
     Deploy an LLM fine-tuned for financial terminologies and concepts. The model interacts with the vector database, fetching only the most relevant snippets for user queries.
   - **Natural Language Interaction:**
     Users can ask questions like:
     - "What are the key financial ratios for Company X over the past 3 years?"
     - "Summarize the cash flow trends in this document."
       - The LLM retrieves the relevant context, processes it, and provides a concise, accurate response.
   - **Enhanced Insights:**
     The system can generate charts, comparisons, and forecasts based on the extracted data, making the analysis process even more actionable.
4. **Advantages:**
   - **Time Efficiency:** Reduces manual data processing significantly.
   - **Intuitive Interaction:** Natural language queries eliminate the need for technical expertise in financial modeling.
   - **Contextual Relevance:** Retrieval-based systems ensure responses are backed by specific, accurate data.
   - **Scalability:** Can handle a vast volume of documents across multiple formats.
   - **Enhanced Visualization:** Integrates with tools to generate charts, reports, and trend analyses.
5. **Disadvantages:**
   - **Data Extraction Challenges:** OCR and parser tools may fail with poorly scanned or non-standardized documents.
   - **Model Accuracy:** LLMs may occasionally misinterpret queries or generate incorrect insights without adequate fine-tuning.
   - **Dependency on Pre-existing Data:** Results are limited to the quality and comprehensiveness of the knowledge base.
   - **Cost:** Setting up and maintaining RAG systems, including vector databases and fine-tuned LLMs, can be expensive.
   - **Security Risks:** Sensitive financial data may require robust encryption and access controls to prevent breaches.
   - **Limited Reasoning:** While LLMs excel in summarizing and retrieving information, they may lack advanced reasoning for complex financial scenarios.
   - **Continuous Maintenance:** Regular updates and monitoring are needed to ensure the model remains accurate with new regulations and data changes.

```
!pip install langchain[all]
```

```
!pip install langchain==0.3.0 --quiet
!pip install langchain_core==0.3.15 --quiet
!pip install langchain_community==0.3.0 --quiet
!pip install langchain_text_splitters==0.3.0 --quiet
!pip install langchain_experimental==0.3.0 --quiet
!pip install langchain_openai==0.2.0 --quiet
!pip install httpx==0.27.2 --quiet
!pip install faiss-cpu==1.8.0 --quiet
!pip install pdfplumber==0.11.0 --quiet
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.0/1.0 MB 12.7 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 408.7/408.7 kB 7.5 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.3/2.3 MB 38.8 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.5/49.5 kB 3.5 MB/s eta 0:00:00
```

```
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 206.9/206.9 kB 3.8 MB/s eta 0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 51.5/51.5 kB 2.8 MB/s eta 0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 21.5 MB/s eta 0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 27.0/27.0 MB 46.8 MB/s eta 0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 48.5/48.5 kB 3.0 MB/s eta 0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.4/56.4 kB 3.5 MB/s eta 0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.6/5.6 MB 60.5 MB/s eta 0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.8/2.8 MB 66.3 MB/s eta 0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 76.4/76.4 kB 2.9 MB/s eta 0:00:00
```

Section A: Initial Setup - Load Libraries, Keys...

```
!pip install --upgrade langchain langchain-openai
```

```
OPENAI_API_KEY = #OPENAI_API_KEY
embeddings = OpenAIEmbeddings(model="text-embedding-ada-002", openai_api_key=OPENAI_API_KEY)
```

```
<ipython-input-5-20cf088b964d>:1: LangChainDeprecationWarning: The class `OpenAIEmbeddings` was deprecated in LangC
  embeddings = OpenAIEmbeddings(model="text-embedding-ada-002", openai_api_key=openai_api_key)
```

```
review_text = "The product is amazing!"
review_embedding = embeddings.embed_query(review_text)
review_embedding
```

```
from langchain_openai import OpenAI
```

```
EMBEDDING_MODEL  = "text-embedding-3-small"
GENERATION_MODEL = "gpt-3.5-turbo-instruct"

llm = openai.OpenAI(model=GENERATION_MODEL)
embed_model = OpenAIEmbeddings(model=EMBEDDING_MODEL)
```

Section B: testing GPT 3.5 it work or not

```
llm.invoke("What is the Capital of India")
```

```
llm.invoke("What is a Quarterly Revenue of Infosys in USD for recent quarter?")
```

Section C : Augment with 10K/10-Q (PDF)

```
pdfFile = "/media/kavi/EFLabs/RAG_MODEL/InfosysPressReleaseQ32024.pdf"

from langchain_community.document_loaders import PDFPlumberLoader
loader = PDFPlumberLoader(pdfFile)
docs = loader.load()

# Check the number of pages
print("Number of pages in the PDF:",len(docs))

# Load the random page content
print(docs[2].page_content)  # Sample content
```

Show hidden output

Section D: Pre-processing of Data

Step D1. Split the document into Chunks

- The SemanticChunker splits text into chunks based on semantic similarity, ensuring that related content stays together in the same

chunk.

```
from langchain_experimental.text_splitter import SemanticChunker
from langchain.embeddings import HuggingFaceEmbeddings

text_splitter = SemanticChunker(HuggingFaceEmbeddings())
documents = text_splitter.split_documents(docs)
```

Show hidden output

```
print(len(documents))  # Number of Chunks
```

```
13
```

```
# Now Look at the content of Second Document post Chucking - the contents will be different
print(documents[2].page_content)
```

```
IFRS – USD
Press Release
1. Client wins & Testimonials
• Infosys announced that it has entered into a long-term collaboration with Metro Bank to
enhance some of its IT and support functions, while digitally transforming the bank’s business
operations. Daniel Frumkin, Metro Bank Chief Executive Officer, said, “This collaboration
with a world class provider like Infosys builds on the solid foundations we have already laid,
unleashing our true potential, and creating a sustainably profitable and scalable organization
that is fit for the future. At the end of this transformation, we will be a very different business,
but the true essence of Metro Bank will remain the same – a high-quality service organization
putting customers centre-stage. Metro Bank expects to deliver £80m of annualized cost
savings this year across multiple initiatives, as it progresses towards the target of reaching mid-
to-high teen Return on Tangible Equity by 2027. Our vision for Metro Bank in 2025 and beyond,
places our store network firmly at its heart, as we continue with our plans to open new stores
and bring the Metro Bank experience to the north of England."
• Infosys announced a strategic collaboration with Proximus to help unlock new business
opportunities. Antonietta Mastroianni, Chief Digital & IT Officer at Proximus, said: “We are
delighted to strengthen our long-standing collaboration with Infosys. By leveraging Infosys’
global reach and our expertise in CPaaS and DI Solutions, the collaboration will drive
innovation and deliver superior customer experiences for our joint customers. We are confident
that our mutual deep expertise and proven track record will be instrumental in this two-way
partnership."
• Infosys announced its collaboration with TDC Net to help them transform from a traditional
infrastructure company to a leading customer-centric technology company. Campbell Fraser,
CTIO, TDC Net said, "At TDC Net, we are committed to delivering exceptional value to our
customers through a transformation in our IT landscape. Our collaboration with Infosys will
enable us to leverage industry-standard processes and platform to create better customer
experiences.
```

Step D2. Create embeddings for each text chunk

- Text (Unstructured Data) Converted to Numeric Representation
- Store in specialised / purpose build Database - Vector Database

```
from langchain_community.embeddings import HuggingFaceEmbeddings
from langchain_community.vectorstores import FAISS

# Instantiate the embedding model
embedder = HuggingFaceEmbeddings()

# Create the vector store
vector = FAISS.from_documents(documents, embedder)
```

```
<ipython-input-23-dc2eb2477ddd>:5: LangChainDeprecationWarning: Default values for HuggingFaceEmbeddings.model_name
  embedder = HuggingFaceEmbeddings()
```

Step D3. Test with Sample Retrieval of Data from the vector database

```
# Input
retriever = vector.as_retriever(search_type="similarity", search_kwargs={"k": 3})
retrieved_docs = retriever.invoke("What is name of the CFO?")
```

```
# Look at the name of CFO in this chuck of Text at the bottom or near end of the chunk
print(retrieved_docs[0].page_content)
```

```
Free cash flow for Q2 was at $839 million, growing 25.2% year
```

```
Free cash flow for Q2 was at $839 million, growing 25.2% year
on year. TCV of large deal wins was $2.4 billion, 41% being net new. H1 revenues grew at 2.9% year over year in con
based with good momentum in financial services. This stems from our strength in industry expertise,
market leading capabilities in cloud with Cobalt and generative AI with Topaz, resulting in growing client
preference to partner with us", said Salil Parekh, CEO and MD. "Our large deals at $2.4 billion in Q2
reflect our differentiated position. I am grateful to our employees for their unwavering commitment to our
client as we further strengthen our market leadership" he added. 3.1% QoQ 21.1% 4.7% YoY $2.4 Bn $839 Mn
3.3% YoY Operating EPS Increase Large Deal Free
CC Growth Margin (₹ terms) TCV Cash Flow
Guidance for FY25:
• Revenue growth of 3.75%-4.50% in constant currency
• Operating margin of 20%-22%
Key highlights:
For the quarter ended September 30, 2024 For six months ended September 30, 2024
• Revenues in CC terms grew by 3.3% YoY and • Revenues in CC terms grew by 2.9% YoY
3.1% QoQ
• Reported revenues at $4,894 million, growth of • Reported revenues at $9,608 million, growth of
3.7% YoY 2.9% YoY
• Operating margin at 21.1%, decline of 0.1% • Operating margin at 21.1%, growth of 0.1%
YoY and flat QoQ YoY
• Basic EPS at $0.19, growth of 3.4% YoY • Basic EPS at $0.37, growth of 4.4% YoY
• FCF at $839 million, growth of 25.2% YoY; • FCF at $1,933 million, growth of 41.2% YoY;
FCF conversion at 107.8% of net profit FCF conversion at 125.3% of net profit
"We continue to focus on accelerating revenue growth with a sharp focus on margin performance. Operating margins fo
and utilization despite higher employee payouts. Our focus on cash generation resulted in another
quarter of over 100% Free Cash Flow conversion to net profits" said Jayesh Sanghrajka, CFO. "The
Board announced an interim dividend of `21 per share, 16.7% increase from last year" he added. Infosys Limited – Pr
```

Section E: Augmentation

```
from langchain.chains import RetrievalQA
from langchain.chains.llm import LLMChain
from langchain.chains.combine_documents.stuff import StuffDocumentsChain
from langchain.prompts import PromptTemplate
```

```
prompt = """
1. Use the following pieces of context to answer the question at the end.
2. If you don't know the answer, just say that "I don't know" but don't make up an answer on your own.\n
3. Keep the answer crisp and limited to 3,4 sentences.

Context: {context}

Question: {question}

Helpful Answer:"""


QA_CHAIN_PROMPT = PromptTemplate.from_template(prompt)

llm_chain = LLMChain(
                  llm=llm,
                  prompt=QA_CHAIN_PROMPT,
                  callbacks=None,
                  verbose=False)

document_prompt = PromptTemplate(
    input_variables=["page_content", "source"],
    template="Context:\ncontent:{page_content}\nsource:{source}",
)

combine_documents_chain = StuffDocumentsChain(
                  llm_chain=llm_chain,
                  document_variable_name="context",
                  document_prompt=document_prompt,
                  callbacks=None,
              )
```

```
<ipython-input-28-010463b0fa9e>:3: LangChainDeprecationWarning: The class `LLMChain` was deprecated in LangChain 0.
  llm_chain = LLMChain(
<ipython-input-28-010463b0fa9e>:14: LangChainDeprecationWarning: This class is deprecated. Use the `create_stuff_do
  combine_documents_chain = StuffDocumentsChain(
```

```
qa = RetrievalQA(
                  combine_documents_chain=combine_documents_chain,
                  verbose=False,
                  retriever=retriever,
                  return_source_documents=False,
              )
```

```
,
```

```
<ipython-input-29-2e0f2d5987ae>:1: LangChainDeprecationWarning: This class is deprecated. Use the `create_retrieval
  qa = RetrievalQA(
```

Section F: testing!!

```
# Input Prompt
# Note : Ignore warnings if you are getting the response
print(qa("What is name of the CFO?")["result"])
```

```
<ipython-input-30-e54365ff8fe3>:2: LangChainDeprecationWarning: The method `Chain.__call__` was deprecated in langc
  print(qa("What is name of the CFO?")["result"])
 The name of the CFO mentioned in the context is Jayesh Sanghrajka.
```

```
pprint(qa("What is a Quarterly Revenue of Infosys in USD for recent quarter?"))
```

```
{'query': 'What is a Quarterly Revenue of Infosys in USD for recent quarter?',
 'result': ' Infosys reported a quarterly revenue of $4,894 million in USD for '
           'the recent quarter, with a sequential growth of 3.1% and a '
           'year-on-year growth of 3.3% in constant currency. This information '
           'was extracted from the audited condensed consolidated Balance '
           'sheet and Statement of Comprehensive Income for the quarter and '
           'six months ended September 30, 2024, which have been taken on '
           'record at the Board meeting held on October 17, 2024.'}
```