In [1]:
```
# 1. Imports
from langchain.chat_models import ChatOpenAI
from langchain.text_splitter import CharacterTextSplitter
from langchain.embeddings import OpenAIEmbeddings
from langchain.vectorstores import FAISS
from langchain.retrievers.document_compressors import LLMChainExtractor
from langchain.schema.document import Document
from langchain.prompts import PromptTemplate
from langchain.chains import LLMChain

import os
from dotenv import load_dotenv
load_dotenv()
os.environ["OPENAI_API_KEY"] = os.getenv("OPENAI_API_KEY")
# 2. Setup LLM and Embeddings
llm = ChatOpenAI(temperature=0, model="gpt-3.5-turbo")
embedding = OpenAIEmbeddings()

# 3. Load your document
with open("sample.txt", "r", encoding="utf-8") as f:
    raw_text = f.read()

# 4. Split it into chunks
splitter = CharacterTextSplitter(separator="\n", chunk_size=300, chunk_overl
chunks = splitter.split_text(raw_text)
documents = [Document(page_content=chunk) for chunk in chunks]

# 5. Compress the documents using LLMChainExtractor
compressor = LLMChainExtractor.from_llm(llm)
compressed_docs = compressor.compress_documents(documents, query="Summarize

print("✅ Compressed Summary:")
for doc in compressed_docs:
    print("-", doc.page_content)

# 6. Now ask a question about the compressed content using a custom LLMChain
qa_prompt = PromptTemplate.from_template(
    "Given the context below, answer the question:\n\nContext:\n{context}\n\
)
qa_chain = LLMChain(llm=llm, prompt=qa_prompt)

response = qa_chain.invoke({
    "context": "\n".join([doc.page_content for doc in compressed_docs]),
    "question": "What is the main idea of the document?"
})

print("\n💡 Answer to your question:")
print(response["text"])
```

```
✅ Compressed Summary:
- LangChain is a framework for building applications with LLMs.LangChain was
created by Harrison Chase.LangChain supports RAG, agents, memory, tools, and
more.It’s commonly used in chatbots, document Q&A, and AI workflow
```

```
💡 Answer to your question:
The main idea of the document is to introduce LangChain as a framework create
d by Harrison Chase for building applications with LLMs. It highlights the fe
atures and uses of LangChain, such as supporting RAG, agents, memory, tools,
and its common applications in chatbots, document Q&A, and AI workflow.
```

In [ ]:

```

```