```
!pip install -q langchain_experimental langchain_core google-generativeai==0.3.1 google-ai-generativelanguage==0.4.0 langchain-google-ge
!pip install -q "langchain[docarray]"
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 167.0/167.0 kB 4.4 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 241.2/241.2 kB 12.6 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 146.6/146.6 kB 13.6 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 815.9/815.9 kB 18.0 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 55.4/55.4 kB 5.7 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.7/1.7 MB 28.2 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.4/49.4 kB 4.1 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 215.3/215.3 kB 5.2 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 139.0/139.0 kB 7.6 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
  Building wheel for hnswlib (pyproject.toml) ... done
```

```
!pip -q install langchain huggingface_hub openai google-search-results tiktoken chromadb lark
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 226.7/226.7 kB 6.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 28.2 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 509.0/509.0 kB 17.3 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 111.7/111.7 kB 9.7 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 75.9/75.9 kB 8.0 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.4/2.4 MB 40.7 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 92.1/92.1 kB 10.4 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 60.8/60.8 kB 6.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 41.1/41.1 kB 4.4 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.4/5.4 MB 56.2 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.8/6.8 MB 79.1 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 57.9/57.9 kB 6.2 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 105.6/105.6 kB 11.5 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.3/67.3 kB 8.0 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 698.9/698.9 kB 42.1 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 59.6 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.6/67.6 kB 7.6 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 71.5/71.5 kB 8.6 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 76.9/76.9 kB 8.4 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.3/58.3 kB 5.7 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 46.0/46.0 kB 5.2 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 50.8/50.8 kB 5.6 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 341.4/341.4 kB 30.2 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.4/3.4 MB 87.0 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.3/1.3 MB 61.2 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 130.2/130.2 kB 13.2 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 86.8/86.8 kB 10.5 MB/s eta 0:00:00
  Building wheel for google-search-results (setup.py) ... done
  Building wheel for pypika (pyproject.toml) ... done
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the sou
lida 0.0.10 requires kaleido, which is not installed.
lida 0.0.10 requires python-multipart, which is not installed.
llmx 0.0.15a0 requires cohere, which is not installed.
```

```
!pip -q install sentence_transformers
!pip -q install -U FlagEmbedding
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 132.8/132.8 kB 4.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 536.6/536.6 kB 8.6 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 280.0/280.0 kB 9.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 38.3/38.3 MB 28.3 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 13.8 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.8/134.8 kB 13.0 MB/s eta 0:00:00
  Building wheel for FlagEmbedding (setup.py) ... done
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the sou
ibis-framework 7.1.0 requires pyarrow<15,>=2, but you have pyarrow 15.0.0 which is incompatible.
```

```
import os

os.environ['GOOGLE_API_KEY']=""
```

```
!pip show langchain-google-genai
```

```
Name: langchain-google-genai
Version: 0.0.9
Summary: An integration package connecting Google's genai package and LangChain
Home-page: https://github.com/langchain-ai/langchain
Author:
```

```
Author-email:
License: MIT
Location: /usr/local/lib/python3.10/dist-packages
Requires: google-generativeai, langchain-core
Required-by:
```

```
!mkdir -p Data
!unzip -q /content/langchain_blog_posts.zip -d Data
```

```
import os
import google.generativeai as genai

genai.configure(api_key="")
```

```
from langchain.vectorstores import FAISS

from langchain.schema import Document
from langchain.vectorstores import Chroma

from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.storage import InMemoryByteStore
from langchain.document_loaders import TextLoader

from langchain_google_genai.embeddings import GoogleGenerativeAIEmbeddings
from langchain_google_genai.chat_models import ChatGoogleGenerativeAI

embeddings = GoogleGenerativeAIEmbeddings(model = "models/embedding-001")
```

```
loaders = [
    TextLoader('/content/Data/blog.langchain.dev_announcing-langsmith_.txt'),
    TextLoader('/content/Data/blog.langchain.dev_benchmarking-question-answering-over-csv-data_.txt'),
]
docs = []
for l in loaders:
    docs.extend(l.load())
```

```
len(docs)
```

```
2
```

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

texts = text_splitter.split_documents(docs)
```

```
# Helper function for printing docs

def pretty_print_docs(docs):
    print(f"\n{'-' * 100}\n".join([f"Document {i+1}:\n\n" + d.page_content for i, d in enumerate(docs)]))
```

```
!pip install -q faiss-cpu
```

```
                                        ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.6/17.6 MB 45.2 MB/s eta 0:00:00
```

```
retriever = FAISS.from_documents(
    texts,
    embeddings,
).as_retriever()

docs = retriever.get_relevant_documents("What is LangSmith?")

pretty_print_docs(docs)
```

```
Document 1:

“Because we are building financial products, the bar for accuracy, personalization, and security is particularly high. LangSmith hel

We can’t wait to bring these benefits to more teams. And we’ve got a long list of features on the roadmap like analytics, playgrounc
----------------------------------------------------------------------------------------------------
Document 2:

URL: https://blog.langchain.dev/announcing-langsmith/
Title: Announcing LangSmith, a unified platform for debugging, testing, evaluating, and monitoring your LLM applications

LangChain exists to make it as easy as possible to develop LLM-powered applications.
```

```
We started with an open-source Python package when the main blocker for building LLM-powered applications was getting a simple proto
----------------------------------------------------------------------------------------------------
Document 3:

“Thanks to Langchain smith we were able to analyze our LLM calls, understand the performance of the different chain methods ( stuff



A unified platform

While each of these product areas provide unique value, often at a specific point in time in the development process, we believe a g
----------------------------------------------------------------------------------------------------
Document 4:

The blocker has now changed. While it’s easy to build a prototype of an application in ~5 lines of LangChain code, it’s still decept

Today, we’re introducing LangSmith, a platform to help developers close the gap between prototype and production. It’s designed for

LangSmith is now in closed beta. So if you’re looking for a robust, unified, system for debugging, testing, evaluating, and monitori



How did we get here?
```

LLMChainExtractor + Contextual compression

```
from langchain_google_genai.chat_models import ChatGoogleGenerativeAI
from langchain.retrievers import ContextualCompressionRetriever
from langchain.retrievers.document_compressors import LLMChainExtractor

llm = ChatGoogleGenerativeAI(
    model="gemini-pro",
    temperature=0.8,
    convert_system_message_to_human=True
)

compressor = LLMChainExtractor.from_llm(llm)

compression_retriever = ContextualCompressionRetriever(
    base_compressor=compressor,
    base_retriever=retriever
)
```

```
compressor.llm_chain.prompt
```

```
PromptTemplate(input_variables=['context', 'question'], output_parser=NoOutputParser(), template='Given the following question and
context, extract any part of the context *AS IS* that is relevant to answer the question. If none of the context is relevant return
NO_OUTPUT. \n\nRemember, *DO NOT* edit the extracted parts of the context.\n\n> Question: {question}\n>
Context:\n>>>\n{context}\n>>>\nExtracted relevant parts:')
```

```
compressed_docs = compression_retriever.get_relevant_documents("What is Langsmith?")

pretty_print_docs(compressed_docs)
```

```
/usr/local/lib/python3.10/dist-packages/langchain/chains/llm.py:316: UserWarning: The predict_and_parse method is deprecated, instea
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/langchain/chains/llm.py:316: UserWarning: The predict_and_parse method is deprecated, instea
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/langchain/chains/llm.py:316: UserWarning: The predict_and_parse method is deprecated, instea
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/langchain/chains/llm.py:316: UserWarning: The predict_and_parse method is deprecated, instea
  warnings.warn(
Document 1:

Today, we’re introducing LangSmith, a platform to help developers close the gap between prototype and production. It’s designed for

LangSmith is now in closed beta. So if you’re looking for a robust, unified, system for debugging, testing, evaluating, and monitori
```

LLMChainFilter

```
print("""Uses an LLM chain to select out the queries to show the final LLM - This could be shown to a model fine tuned to do this

"YES" we show it or "NO" we don't show it""")
```

```
Uses an LLM chain to select out the queries to show the final LLM - This could be shown to a model fine tuned to do this
```

```
"YES" we show it or "NO" we don't show it
```

```
from langchain.retrievers.document_compressors import LLMChainFilter

filter_ = LLMChainFilter.from_llm(llm)

filter_.llm_chain.prompt
```

```
PromptTemplate(input_variables=['context', 'question'], output_parser=BooleanOutputParser(), template="Given the following question
and context, return YES if the context is relevant to the question and NO if it isn't.\n\n> Question: {question}\n>
Context:\n>>>\n{context}\n>>>\n> Relevant (YES / NO):")
```

```
compression_retriever = ContextualCompressionRetriever(base_compressor=filter_, base_retriever=retriever)

compressed_docs = compression_retriever.get_relevant_documents("What is LangSmith")
pretty_print_docs(compressed_docs)
```

```
/usr/local/lib/python3.10/dist-packages/langchain/chains/llm.py:316: UserWarning: The predict_and_parse method is deprecated, instea
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/langchain/chains/llm.py:316: UserWarning: The predict_and_parse method is deprecated, instea
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/langchain/chains/llm.py:316: UserWarning: The predict_and_parse method is deprecated, instea
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/langchain/chains/llm.py:316: UserWarning: The predict_and_parse method is deprecated, instea
  warnings.warn(
Document 1:

“Because we are building financial products, the bar for accuracy, personalization, and security is particularly high. LangSmith hel

We can’t wait to bring these benefits to more teams. And we’ve got a long list of features on the roadmap like analytics, playgrounc
----------------------------------------------------------------------------------------------------
Document 2:

URL: https://blog.langchain.dev/announcing-langsmith/
Title: Announcing LangSmith, a unified platform for debugging, testing, evaluating, and monitoring your LLM applications

LangChain exists to make it as easy as possible to develop LLM-powered applications.

We started with an open-source Python package when the main blocker for building LLM-powered applications was getting a simple proto
----------------------------------------------------------------------------------------------------
Document 3:

The blocker has now changed. While it’s easy to build a prototype of an application in ~5 lines of LangChain code, it’s still decept

Today, we’re introducing LangSmith, a platform to help developers close the gap between prototype and production. It’s designed for

LangSmith is now in closed beta. So if you’re looking for a robust, unified, system for debugging, testing, evaluating, and monitori



How did we get here?
```

EmbeddingsFilter

```
from langchain_google_genai.embeddings import GoogleGenerativeAIEmbeddings
from langchain_google_genai.chat_models import ChatGoogleGenerativeAI

embeddings = GoogleGenerativeAIEmbeddings(model = "models/embedding-001")
```

```
from langchain.retrievers.document_compressors import EmbeddingsFilter

embeddings_filter = EmbeddingsFilter(embeddings=embeddings, similarity_threshold=0.16)
compression_retriever = ContextualCompressionRetriever(base_compressor=embeddings_filter, base_retriever=retriever)

compressed_docs = compression_retriever.get_relevant_documents("What is LangSmith")
pretty_print_docs(compressed_docs)
```

```
Document 1:

“Because we are building financial products, the bar for accuracy, personalization, and security is particularly high. LangSmith hel

We can’t wait to bring these benefits to more teams. And we’ve got a long list of features on the roadmap like analytics, playgrounc
----------------------------------------------------------------------------------------------------
Document 2:

“Thanks to Langchain smith we were able to analyze our LLM calls, understand the performance of the different chain methods ( stuff



A unified platform
```

```
While each of these product areas provide unique value, often at a specific point in time in the development process, we believe a g
----------------------------------------------------------------------------------------------------
Document 3:

URL: https://blog.langchain.dev/announcing-langsmith/
Title: Announcing LangSmith, a unified platform for debugging, testing, evaluating, and monitoring your LLM applications

LangChain exists to make it as easy as possible to develop LLM-powered applications.

We started with an open-source Python package when the main blocker for building LLM-powered applications was getting a simple proto
----------------------------------------------------------------------------------------------------
Document 4:

The blocker has now changed. While it's easy to build a prototype of an application in ~5 lines of LangChain code, it's still decept

Today, we're introducing LangSmith, a platform to help developers close the gap between prototype and production. It's designed for

LangSmith is now in closed beta. So if you're looking for a robust, unified, system for debugging, testing, evaluating, and monitori



How did we get here?
```

Pipelines

```
from langchain.document_transformers import EmbeddingsRedundantFilter
from langchain.retrievers.document_compressors import DocumentCompressorPipeline
from langchain.text_splitter import CharacterTextSplitter

splitter = CharacterTextSplitter(chunk_size=500, chunk_overlap=20, separator=". ")

redundant_filter = EmbeddingsRedundantFilter(embeddings=embeddings)
relevant_filter = EmbeddingsFilter(embeddings=embeddings, similarity_threshold=0.16)


pipeline_compressor = DocumentCompressorPipeline(
    transformers=[splitter, redundant_filter, relevant_filter]
)
```

```
compression_retriever = ContextualCompressionRetriever(base_compressor=pipeline_compressor,
                                                       base_retriever=retriever)

compressed_docs = compression_retriever.get_relevant_documents("What is LangSmith")
pretty_print_docs(compressed_docs)
```

```
Document 1:

URL: https://blog.langchain.dev/announcing-langsmith/
Title: Announcing LangSmith, a unified platform for debugging, testing, evaluating, and monitoring your LLM applications

LangChain exists to make it as easy as possible to develop LLM-powered applications.

We started with an open-source Python package when the main blocker for building LLM-powered applications was getting a simple proto
----------------------------------------------------------------------------------------------------
Document 2:

It's designed for building and iterating on products that can harness the power–and wrangle the complexity–of LLMs.

LangSmith is now in closed beta. So if you're looking for a robust, unified, system for debugging, testing, evaluating, and monitori



How did we get here?
----------------------------------------------------------------------------------------------------
Document 3:

"Thanks to Langchain smith we were able to analyze our LLM calls, understand the performance of the different chain methods ( stuff
----------------------------------------------------------------------------------------------------
Document 4:

We are consistently using it to improve our prompt engineering and look forward to the new features," said Stan Girard, Head of GenA



A unified platform

While each of these product areas provide unique value, often at a specific point in time in the development process, we believe a g
----------------------------------------------------------------------------------------------------
Document 5:
```

```
We see teams with all kinds of Rube Goldberg-machine-like processes for managing their LLM applications, and we want to make that a
----------------------------------------------------------------------------------------------------
Document 6:

“Because we are building financial products, the bar for accuracy, personalization, and security is particularly high. LangSmith hel

We can’t wait to bring these benefits to more teams. And we’ve got a long list of features on the roadmap like analytics, playgrounc
----------------------------------------------------------------------------------------------------
Document 7:

The blocker has now changed. While it’s easy to build a prototype of an application in ~5 lines of LangChain code, it’s still decept

Today, we’re introducing LangSmith, a platform to help developers close the gap between prototype and production
----------------------------------------------------------------------------------------------------
Document 8:

We remember seeing Nat Friedman tweet in late 2022 that there was “not enough tinkering happening.” The LangChain open-source packag
```

```
pipeline_compressor = DocumentCompressorPipeline(
    transformers=[splitter, compressor, redundant_filter, relevant_filter]
)
```

```
compression_retriever = ContextualCompressionRetriever(base_compressor=pipeline_compressor,
                                                       base_retriever=retriever)

compressed_docs = compression_retriever.get_relevant_documents("What is LangSmith")
pretty_print_docs(compressed_docs)
```

```
/usr/local/lib/python3.10/dist-packages/langchain/chains/llm.py:316: UserWarning: The predict_and_parse method is deprecated, instea
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/langchain/chains/llm.py:316: UserWarning: The predict_and_parse method is deprecated, instea
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/langchain/chains/llm.py:316: UserWarning: The predict_and_parse method is deprecated, instea
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/langchain/chains/llm.py:316: UserWarning: The predict_and_parse method is deprecated, instea
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/langchain/chains/llm.py:316: UserWarning: The predict_and_parse method is deprecated, instea
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/langchain/chains/llm.py:316: UserWarning: The predict_and_parse method is deprecated, instea
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/langchain/chains/llm.py:316: UserWarning: The predict_and_parse method is deprecated, instea
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/langchain/chains/llm.py:316: UserWarning: The predict_and_parse method is deprecated, instea
  warnings.warn(
Document 1:

LangSmith is now in closed beta. So if you’re looking for a robust, unified, system for debugging, testing, evaluating, and monitori
----------------------------------------------------------------------------------------------------
Document 2:

A unified platform

While each of these product areas provide unique value, often at a specific point in time in the development process, we believe a g
----------------------------------------------------------------------------------------------------
Document 3:

Today, we’re introducing LangSmith, a platform to help developers close the gap between prototype and production
----------------------------------------------------------------------------------------------------
Document 4:

“Thanks to Langchain smith we were able to analyze our LLM calls, understand the performance of the different chain methods ( stuff
```

Start coding or generate with AI.