

```
"RAG demo"
```

```
# Imports
```

```
from openai import OpenAI
import numpy as np
import pandas as pd
```

```
# Functions
```

```
def get_embeddings(chunks):
    response = OpenAI().embeddings.create(input=chunks, model="text-embedding-3-large")
    return [np.array(record.embedding) for record in response.data]
```

```
def cosine_similarity(a, b):
    return np.dot(a, b) / (np.linalg.norm(a) * np.linalg.norm(b))
```

```
def distance(a, b):
    return 1 - cosine_similarity(a, b)
```

```
def get_response(prompt):
    response = OpenAI().chat.completions.create(
        model = 'gpt-4-0125-preview',
        messages=[
            {"role": "user", "content": prompt}
        ]
    )
    return response.choices[0].message.content
```

```
# Snippet from a recent Financial Stability Report (5 paragraphs)
```

```
text = ""
A summary of the developments in the four broad categories of vulnerabilities since the last report is as follows:
```

1. Asset valuations. Equity prices grew faster than expected earnings, pushing the forward price-to-earnings ratio into the upper ranges of its historical distribution. Risk premiums in corporate bond markets narrowed somewhat and remained near the middle of their historical distributions. Prices of residential and commercial properties remained high relative to fundamentals (see Section 1, Asset Valuations).

2. Borrowing by businesses and households. Balance sheets of many nonfinancial businesses and households remained solid. Growth of business debt continued to decline through the first half of the year, although business debt remained high when measured relative to gross domestic product (GDP) or business assets. Measures of the ability of firms to service their debt remained strong. Household debt remained at modest levels relative to GDP, with most of that debt owed by households with strong credit histories or considerable home equity (see Section 2, Borrowing by Businesses and Households).

3. Leverage in the financial sector. The banking sector remains sound and resilient overall, and most banks continued to report capital levels well above regulatory requirements. That said, the increase in interest rates over the past two years has contributed to declines in the fair value of longer-maturity, fixed-rate assets that, for some banks, were sizable. Outside the banking sector, available data suggest that hedge fund leverage remained somewhat elevated, especially for the largest hedge funds. Leverage at life insurance companies remained near the middle of its historical range, while broker-dealer leverage remained historically low (see Section 3, Leverage in the Financial Sector).

4. Funding risks. Most domestic banks have ample liquidity and limited reliance on short-term wholesale funding; nevertheless, some banks continued to face funding strains, likely owing to vulnerabilities associated with high levels of uninsured deposits and declines in the fair value of assets. The Bank Term Funding Program (BTFP) helped mitigate these strains. Structural vulnerabilities remained in other short-term funding markets. Prime and tax-exempt money market funds (MMFs), as well as other cash-investment vehicles and stablecoins, remained vulnerable to runs. Bond and loan funds that hold assets that can become illiquid during periods of stress remained susceptible to large redemptions. Life insurers continued to rely on a higher-than-average share of runnable liabilities (see Section 4, Funding Risks).

```
"""  
  
# Step 1: split text into chunks
```

```
chunks = text.split('\n\n')  
print(f1) Text split into {len(chunks)} chunks')
```

```
# Step 2: compute embeddings for each chunk  
embeddings = get_embeddings(chunks)  
print(f2) Computed {len(embeddings)} embeddings of size {len(embeddings[0])}')
```

```
# Step 3: select relevant embeddings  
query = 'How has household debt evolved lately?'  
query_embedding = get_embeddings(query)[0]
```

```
for i, chunk in enumerate(chunks):  
    d = distance(query_embedding, embeddings[i])  
    print(f' - Distance of chunk {i} to query: {d:4.2f}')
```

```
distances = [distance(query_embedding, e) for e in embeddings]  
best_i = np.argmin(distances)  
context = chunks[best_i]  
print(f3) Best chunk is number {best_i} (" {context[:30]}...")')
```

```
# Step 4: query the LLM
```

```
prompt = f'''
```

```
You have been tasked to extract information from a report.
```

```
Text excerpts for this examination have been attached at the end of this text, after the word "CONTEXT:".
```

```
Please answer the following question: {query}
```

```
CONTEXT:
```

```
{context}
```

```
'''
```

```
print(f4) LLM answer:')  
answer = get_response(prompt)  
print(answer)
```

```
# Step 5: query the LLM to retrieve Stata-ready numeric output
```

```
def get_json_response(prompt):  
    response = OpenAI().chat.completions.create(  
        model = 'gpt-4-0125-preview',
```

```
response_format = { "type": "json_object" },
messages=[
    {"role": "system", "content": "You are a helpful assistant designed to output JSON."},
    {"role": "user", "content": prompt}
]
)
return response.choices[0].message.content
```

```
prompt = f"
```

You have been tasked to extract information from a report. Text excerpts for this examination have been attached at the end of this text, after the word "CONTEXT:".

Please answer the following question: {query}

Please provide your response using two JSON fields, the first one named 'success' with values True or False, the second named 'answer', with values 1 to 5, with 1 meaning "great" and 5 meaning "terrible".

CONTEXT:

```
{context}
```

```
"
```

```
print(f'LLM answer:')
answer = get_response(prompt)
print(answer)
```