

Lesson 4: Constructing a Knowledge Graph from Text Documents

Note: This notebook takes about 30 seconds to be ready to use. Please wait until the "Kernel starting, please wait..." message clears from the top of the notebook before running any cells. You may start the video while you wait.

Import packages and set up Neo4j

```
In [1]: ▶ from dotenv import load_dotenv
import os

# Common data processing
import json
import textwrap

# Langchain
from langchain_community.graphs import Neo4jGraph
from langchain_community.vectorstores import Neo4jVector
from langchain_openai import OpenAIEmbeddings
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.chains import RetrievalQAWithSourcesChain
from langchain_openai import ChatOpenAI

# Warning control
import warnings
warnings.filterwarnings("ignore")
```

```
In [2]: ▶ # Load from environment
load_dotenv('.env', override=True)
NEO4J_URI = os.getenv('NEO4J_URI')
NEO4J_USERNAME = os.getenv('NEO4J_USERNAME')
NEO4J_PASSWORD = os.getenv('NEO4J_PASSWORD')
NEO4J_DATABASE = os.getenv('NEO4J_DATABASE') or 'neo4j'
OPENAI_API_KEY = os.getenv('OPENAI_API_KEY')
# Note the code below is unique to this course environment, and not
# standard part of Neo4j's integration with OpenAI. Remove if running
# in your own environment.
OPENAI_ENDPOINT = os.getenv('OPENAI_BASE_URL') + '/embeddings'

# Global constants
VECTOR_INDEX_NAME = 'form_10k_chunks'
VECTOR_NODE_LABEL = 'Chunk'
VECTOR_SOURCE_PROPERTY = 'text'
VECTOR_EMBEDDING_PROPERTY = 'textEmbedding'
```

Take a look at a Form 10-K json file

- Publicly traded companies are required to fill a form 10-K each year with the Securities and Exchange Commission (SEC)

- You can search these filings using the SEC's [EDGAR database \(https://www.sec.gov/edgar/search/\)](https://www.sec.gov/edgar/search/)
- For the next few lessons, you'll work with a single 10-K form for a company called [NetApp \(https://www.netapp.com/\)](https://www.netapp.com/)

```
In [3]: first_file_name = "./data/form10k/0000950170-23-027948.json"
```

```
In [4]: first_file_as_object = json.load(open(first_file_name))
```

```
In [5]: type(first_file_as_object)
```

dict

```
In [6]: for k,v in first_file_as_object.items():
        print(k, type(v))
```

```
item1 <class 'str'>
item1a <class 'str'>
item7 <class 'str'>
item7a <class 'str'>
cik <class 'str'>
cusip6 <class 'str'>
cusip <class 'list'>
names <class 'list'>
source <class 'str'>
```

```
In [7]: item1_text = first_file_as_object['item1']
```

```
In [8]: item1_text[0:1500]
```

```
'>Item 1. \nBusiness\n\n\nOverview\n\n\nNetApp, Inc. (NetApp, we, us or
the Company) is a global cloud-led, data-centric software company. We wer
e incorporated in 1992 and are headquartered in San Jose, California. Bui
lding on more than three decades of innovation, we give customers the fre
edom to manage applications and data across hybrid multicloud environmen
ts. Our portfolio of cloud services, and storage infrastructure, powered b
y intelligent data management software, enables applications to run faste
r, more reliably, and more securely, all at a lower cost.\n\n\nOur opport
unity is defined by the durable megatrends of data-driven digital and clo
ud transformations. NetApp helps organizations meet the complexities crea
ted by rapid data and cloud growth, multi-cloud management, and the adopt
ion of next-generation technologies, such as AI, Kubernetes, and modern d
atabases. Our modern approach to hybrid, multicloud infrastructure and da
ta management, which we term 'evolved cloud', provides customers the abil
ity to leverage data across their entire estate with simplicity, securit
y, and sustainability which increases our relevance and value to our cust
omers.\n\n\nIn an evolved cloud state, the cloud is fully integrated into
an organization's architecture and operations. Data centers and clouds ar
e seamlessly united and hybrid multicloud operations are simplified, with
consistency and observability across environments. The key benefits NetAp
p brings to an organization's hybrid multicloud enviro'
```

Split Form 10-K sections into chunks

- Set up text splitter using LangChain

```
In [9]: text_splitter = RecursiveCharacterTextSplitter(
        chunk_size = 2000,
        chunk_overlap = 200,
        length_function = len,
        is_separator_regex = False,
    )
```

```
In [10]: item1_text_chunks = text_splitter.split_text(item1_text)
```

```
In [11]: type(item1_text_chunks)
```

list

```
In [12]: len(item1_text_chunks)
```

254

```
In [13]: item1_text_chunks[0]
```

'>Item 1. \nBusiness\n\n\nOverview\n\n\n\nNetApp, Inc. (NetApp, we, us or the Company) is a global cloud-led, data-centric software company. We were incorporated in 1992 and are headquartered in San Jose, California. Building on more than three decades of innovation, we give customers the freedom to manage applications and data across hybrid multicloud environments. Our portfolio of cloud services, and storage infrastructure, powered by intelligent data management software, enables applications to run faster, more reliably, and more securely, all at a lower cost.\n\n\nOur opportunity is defined by the durable megatrends of data-driven digital and cloud transformations. NetApp helps organizations meet the complexities created by rapid data and cloud growth, multi-cloud management, and the adoption of next-generation technologies, such as AI, Kubernetes, and modern databases. Our modern approach to hybrid, multicloud infrastructure and data management, which we term 'evolved cloud', provides customers the ability to leverage data across their entire estate with simplicity, security, and sustainability which increases our relevance and value to our customers.\n\n\nIn an evolved cloud state, the cloud is fully integrated into an organization's architecture and operations. Data centers and clouds are seamlessly united and hybrid multicloud operations are simplified, with consistency and observability across environments. The key benefits NetApp brings to an organization's hybrid multicloud environment are:\n\n\n•\nOperational simplicity: NetApp's use of open source, open architectures and APIs, microservices, and common capabilities and data services facilitate the creation of applications that can run anywhere.\n\n\n•\nFlexibility and consistency: NetApp makes moving data and applications between environments seamless through a common storage foundation across on-premises and multicloud environments.'

- Set up helper function to chunk all sections of the Form 10-K
- You'll limit the number of chunks in each section to 20 to speed things up

```

In [14]: ▶ def split_form10k_data_from_file(file):
    chunks_with_metadata = [] # use this to accumulate chunk records
    file_as_object = json.load(open(file)) # open the json file
    for item in ['item1', 'item1a', 'item7', 'item7a']: # pull these keys
        print(f'Processing {item} from {file}')
        item_text = file_as_object[item] # grab the text of the item
        item_text_chunks = text_splitter.split_text(item_text) # split into chunks
        chunk_seq_id = 0
        for chunk in item_text_chunks[:20]: # only take the first 20 chunks
            form_id = file[file.rindex('/') + 1:file.rindex('.')] # get form id
            # finally, construct a record with metadata and the chunk
            chunks_with_metadata.append({
                'text': chunk,
                # metadata from looping...
                'f10kItem': item,
                'chunkSeqId': chunk_seq_id,
                # constructed metadata...
                'formId': f'{form_id}', # pulled from the filename
                'chunkId': f'{form_id}-{item}-chunk{chunk_seq_id:04d}',
                # metadata from file...
                'names': file_as_object['names'],
                'cik': file_as_object['cik'],
                'cusip6': file_as_object['cusip6'],
                'source': file_as_object['source'],
            })
            chunk_seq_id += 1
        print(f'\tSplit into {chunk_seq_id} chunks')
    return chunks_with_metadata

```

```

In [15]: ▶ first_file_chunks = split_form10k_data_from_file(first_file_name)

```

Processing item1 from ./data/form10k/0000950170-23-027948.json

Split into 20 chunks

Processing item1a from ./data/form10k/0000950170-23-027948.json

Split into 1 chunks

Processing item7 from ./data/form10k/0000950170-23-027948.json

Split into 1 chunks

Processing item7a from ./data/form10k/0000950170-23-027948.json

Split into 1 chunks

In [16]: `first_file_chunks[0]`

```
{'text': '>Item 1. \nBusiness\n\n\nOverview\n\n\nNetApp, Inc. (NetApp, we, us or the Company) is a global cloud-led, data-centric software company. We were incorporated in 1992 and are headquartered in San Jose, California. Building on more than three decades of innovation, we give customers the freedom to manage applications and data across hybrid multicloud environments. Our portfolio of cloud services, and storage infrastructure, powered by intelligent data management software, enables applications to run faster, more reliably, and more securely, all at a lower cost.\n\n\nOur opportunity is defined by the durable megatrends of data-driven digital and cloud transformations. NetApp helps organizations meet the complexities created by rapid data and cloud growth, multi-cloud management, and the adoption of next-generation technologies, such as AI, Kubernetes, and modern databases. Our modern approach to hybrid, multicloud infrastructure and data management, which we term ‘evolved cloud’, provides customers the ability to leverage data across their entire estate with simplicity, security, and sustainability which increases our relevance and value to our customers.\n\n\nIn an evolved cloud state, the cloud is fully integrated into an organization’s architecture and operations. Data centers and clouds are seamlessly united and hybrid multicloud operations are simplified, with consistency and observability across environments. The key benefits NetApp brings to an organization’s hybrid multicloud environment are:\n\n\n•\nOperational simplicity: NetApp’s use of open source, open architectures and APIs, microservices, and common capabilities and data services facilitate the creation of applications that can run anywhere.\n\n\n•\nFlexibility and consistency: NetApp makes moving data and applications between environments seamless through a common storage foundation across on-premises and multicloud environments.',
'f10kItem': 'item1',
'chunkSeqId': 0,
'formId': '0000950170-23-027948',
'chunkId': '0000950170-23-027948-item1-chunk0000',
'names': ['Netapp Inc', 'NETAPP INC'],
'cik': '1002047',
'cusip6': '64110D',
'source': 'https://www.sec.gov/Archives/edgar/data/1002047/000095017023027948/0000950170-23-027948-index.htm'}
```

In [17]: `first_file_chunks[1]`

```
{'text': "\nFlexibility and consistency: NetApp makes moving data and applications between environments seamless through a common storage foundation across on-premises and multicloud environments.\n\n\n•\nCyber resilience: NetApp unifies monitoring, data protection, security, governance, and compliance for total cyber resilience - with consistency and automation across environments. \n\n\n•\nContinuous operations: NetApp uses AI-driven automation for continuous optimization to service applications and store stateless and stateful applications at the lowest possible costs.\n\n\n•\nSustainability: NetApp has industry-leading tools to audit consumption, locate waste, and set guardrails to stop overprovisioning.\n\n\nProduct, Solutions and Services Portfolio\n\n\n\nNetApp's portfolio of cloud services and storage infrastructure is powered by intelligent data management software. Our operations are organized into two segments: Hybrid Cloud and Public Cloud.\n\n\n\n\nHybrid Cloud\n\n\n\nHybrid Cloud \noffers a portfolio of storage management and infrastructure solutions that help customers recast their traditional data centers into modern data centers with the power of the cloud. Our hybrid cloud portfolio is designed to operate with public clouds to unlock the potential of hybrid, multi-cloud operations. We offer a broad portfolio of cloud-connected all-flash, hybrid-flash, and object storage systems, powered by intelligent data management software. Hybrid Cloud is composed of software, hardware, and related support, as well as professional and other services.\n\n\n\nIntelligent data management software",
'f10kItem': 'item1',
'chunkSeqId': 1,
'formId': '0000950170-23-027948',
'chunkId': '0000950170-23-027948-item1-chunk0001',
'names': ['Netapp Inc', 'NETAPP INC'],
'cik': '1002047',
'cusip6': '64110D',
'source': 'https://www.sec.gov/Archives/edgar/data/1002047/000095017023027948/0000950170-23-027948-index.htm'}
```

Create graph nodes using text chunks

```
In [18]: merge_chunk_node_query = """
MERGE(mergedChunk:Chunk {chunkId: $chunkParam.chunkId})
ON CREATE SET
    mergedChunk.names = $chunkParam.names,
    mergedChunk.formId = $chunkParam.formId,
    mergedChunk.cik = $chunkParam.cik,
    mergedChunk.cusip6 = $chunkParam.cusip6,
    mergedChunk.source = $chunkParam.source,
    mergedChunk.f10kItem = $chunkParam.f10kItem,
    mergedChunk.chunkSeqId = $chunkParam.chunkSeqId,
    mergedChunk.text = $chunkParam.text
RETURN mergedChunk
"""
```

- Set up connection to graph instance using LangChain

```
In [19]: kg = Neo4jGraph(
    url=NEO4J_URI, username=NEO4J_USERNAME, password=NEO4J_PASSWORD,
)
```

- Create a single chunk node for now

```
In [20]: ► kg.query(merge_chunk_node_query,
                params={'chunkParam':first_file_chunks[0]})

[{'mergedChunk': {'formId': '0000950170-23-027948',
                  'f10kItem': 'item1',
                  'names': ['Netapp Inc', 'NETAPP INC'],
                  'cik': '1002047',
                  'cusip6': '64110D',
                  'source': 'https://www.sec.gov/Archives/edgar/data/1002047/00009501702
3027948/0000950170-23-027948-index.htm',
                  'text': '>Item 1. \nBusiness\n\n\nOverview\n\n\n\nNetApp, Inc. (NetApp,
we, us or the Company) is a global cloud-led, data-centric software compa
ny. We were incorporated in 1992 and are headquartered in San Jose, Calif
ornia. Building on more than three decades of innovation, we give custome
rs the freedom to manage applications and data across hybrid multicloud e
nvironments. Our portfolio of cloud services, and storage infrastructure,
powered by intelligent data management software, enables applications to
run faster, more reliably, and more securely, all at a lower cost.\n\n\nO
ur opportunity is defined by the durable megatrends of data-driven digita
l and cloud transformations. NetApp helps organizations meet the complexi
ties created by rapid data and cloud growth, multi-cloud management, and
the adoption of next-generation technologies, such as AI, Kubernetes, and
modern databases. Our modern approach to hybrid, multicloud infrastruclur
e and data management, which we term 'evolved cloud', provides customers
the ability to leverage data across their entire estate with simplicity,
security, and sustainability which increases our relevance and value to o
ur customers.\n\n\nIn an evolved cloud state, the cloud is fully integrat
ed into an organization's architecture and operations. Data centers and c
louds are seamlessly united and hybrid multicloud operations are simplifi
ed, with consistency and observability across environments. The key benef
its NetApp brings to an organization's hybrid multicloud environment ar
e:\n\n\n•\nOperational simplicity: NetApp's use of open source, open arch
itectures and APIs, microservices, and common capabilities and data servi
ces facilitate the creation of applications that can run anywhere.\n\n\n•
\nFlexibility and consistency: NetApp makes moving data and applications
between environments seamless through a common storage foundation across
on-premises and multicloud environments.'},
  {'chunkId': '0000950170-23-027948-item1-chunk0000',
   'chunkSeqId': 0}]}
```

- Create a uniqueness constraint to avoid duplicate chunks

```
In [21]: ► kg.query("""
CREATE CONSTRAINT unique_chunk IF NOT EXISTS
FOR (c:Chunk) REQUIRE c.chunkId IS UNIQUE
""")
```

```
[]
```

```
In [22]: kg.query("SHOW INDEXES")
```

```
[{'id': 1,
  'name': 'index_343aff4e',
  'state': 'ONLINE',
  'populationPercent': 100.0,
  'type': 'LOOKUP',
  'entityType': 'NODE',
  'labelsOrTypes': None,
  'properties': None,
  'indexProvider': 'token-lookup-1.0',
  'owningConstraint': None,
  'lastRead': None,
  'readCount': 0},
 {'id': 2,
  'name': 'index_f7700477',
  'state': 'ONLINE',
  'populationPercent': 100.0,
  'type': 'LOOKUP',
  'entityType': 'RELATIONSHIP',
  'labelsOrTypes': None,
  'properties': None,
  'indexProvider': 'token-lookup-1.0',
  'owningConstraint': None,
  'lastRead': None,
  'readCount': 0},
 {'id': 3,
  'name': 'unique_chunk',
  'state': 'ONLINE',
  'populationPercent': 100.0,
  'type': 'RANGE',
  'entityType': 'NODE',
  'labelsOrTypes': ['Chunk'],
  'properties': ['chunkId'],
  'indexProvider': 'range-1.0',
  'owningConstraint': 'unique_chunk',
  'lastRead': None,
  'readCount': None}]
```

- Loop through and create nodes for all chunks
- Should create 23 nodes because you set a limit of 20 chunks in the text splitting function above


```
In [23]: ▶ node_count = 0
for chunk in first_file_chunks:
    print(f"Creating `:Chunk` node for chunk ID {chunk['chunkId']}")
    kg.query(merge_chunk_node_query,
              params={
                  'chunkParam': chunk
              })
    node_count += 1
print(f"Created {node_count} nodes")
```

```
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0000
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0001
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0002
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0003
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0004
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0005
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0006
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0007
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0008
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0009
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0010
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0011
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0012
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0013
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0014
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0015
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0016
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0017
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0018
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1-chunk0019
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item1a-chunk0000
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item7-chunk0000
Creating `:Chunk` node for chunk ID 0000950170-23-027948-item7a-chunk0000
Created 23 nodes
```

```
In [24]: ▶ kg.query("""
MATCH (n)
RETURN count(n) as nodeCount
""")
```

```
[{'nodeCount': 23}]
```

Create a vector index

```
In [25]: ▶ kg.query("""
CREATE VECTOR INDEX `form_10k_chunks` IF NOT EXISTS
FOR (c:Chunk) ON (c.textEmbedding)
OPTIONS { indexConfig: {
    `vector.dimensions`: 1536,
    `vector.similarity_function`: 'cosine'
}}
""")
```

```
[]
```

In [26]: `kg.query("SHOW INDEXES")`

```
[{'id': 5,
  'name': 'form_10k_chunks',
  'state': 'ONLINE',
  'populationPercent': 100.0,
  'type': 'VECTOR',
  'entityType': 'NODE',
  'labelsOrTypes': ['Chunk'],
  'properties': ['textEmbedding'],
  'indexProvider': 'vector-1.0',
  'owningConstraint': None,
  'lastRead': None,
  'readCount': None},
 {'id': 1,
  'name': 'index_343aff4e',
  'state': 'ONLINE',
  'populationPercent': 100.0,
  'type': 'LOOKUP',
  'entityType': 'NODE',
  'labelsOrTypes': None,
  'properties': None,
  'indexProvider': 'token-lookup-1.0',
  'owningConstraint': None,
  'lastRead': None,
  'readCount': 0},
 {'id': 2,
  'name': 'index_f7700477',
  'state': 'ONLINE',
  'populationPercent': 100.0,
  'type': 'LOOKUP',
  'entityType': 'RELATIONSHIP',
  'labelsOrTypes': None,
  'properties': None,
  'indexProvider': 'token-lookup-1.0',
  'owningConstraint': None,
  'lastRead': None,
  'readCount': 0},
 {'id': 3,
  'name': 'unique_chunk',
  'state': 'ONLINE',
  'populationPercent': 100.0,
  'type': 'RANGE',
  'entityType': 'NODE',
  'labelsOrTypes': ['Chunk'],
  'properties': ['chunkId'],
  'indexProvider': 'range-1.0',
  'owningConstraint': 'unique_chunk',
  'lastRead': neo4j.time.DateTime(2024, 3, 18, 9, 51, 10, 234000000, tzinfo=fo=UTC),
  'readCount': 67}]
```

Calculate embedding vectors for chunks and populate index

- This query calculates the embedding vector and stores it as a property called `textEmbedding` on each `Chunk` node.

```
In [27]: ▶ kg.query("""
MATCH (chunk:Chunk) WHERE chunk.textEmbedding IS NULL
WITH chunk, genai.vector.encode(
    chunk.text,
    "OpenAI",
    {
        token: $openAiApiKey,
        endpoint: $openAiEndpoint
    }) AS vector
CALL db.create.setNodeVectorProperty(chunk, "textEmbedding", vec
""",
params={"openAiApiKey":OPENAI_API_KEY, "openAiEndpoint": OPENAI_
```

[]

```
In [28]: ▶ kg.refresh_schema()
print(kg.schema)
```

Node properties are the following:

Chunk {textEmbedding: LIST, f10kItem: STRING, chunkSeqId: INTEGER, text: STRING, cik: STRING, cusip6: STRING, names: LIST, formId: STRING, source: STRING, chunkId: STRING}

Relationship properties are the following:

The relationships are the following:

Use similarity search to find relevant chunks

- Setup a help function to perform similarity search using the vector index

```
In [29]: ▶
    token: $openAiApiKey,
    endpoint: $openAiEndpoint
}) AS question_embedding
CALL db.index.vector.queryNodes($index_name, $top_k, question_en
RETURN score, node.text AS text
"""
similar = kg.query(vector_search_query,
                    params={
                        'question': question,
                        'openAiApiKey':OPENAI_API_KEY,
                        'openAiEndpoint': OPENAI_ENDPOINT,
                        'index_name':VECTOR_INDEX_NAME,
                        'top_k': 10})

return similar
```

- Ask a question!

```
In [30]: ▶ search_results = neo4j_vector_search(
    'In a single sentence, tell me about Netapp.'
)
```

In [31]: `search_results[0]`

```
{'score': 0.9358915090560913,
 'text': '>Item 1. \nBusiness\n\n\nOverview\n\n\nNetApp, Inc. (NetApp, we, us or the Company) is a global cloud-led, data-centric software company. We were incorporated in 1992 and are headquartered in San Jose, California. Building on more than three decades of innovation, we give customers the freedom to manage applications and data across hybrid multicloud environments. Our portfolio of cloud services, and storage infrastructure, powered by intelligent data management software, enables applications to run faster, more reliably, and more securely, all at a lower cost.\n\n\nOur opportunity is defined by the durable megatrends of data-driven digital and cloud transformations. NetApp helps organizations meet the complexities created by rapid data and cloud growth, multi-cloud management, and the adoption of next-generation technologies, such as AI, Kubernetes, and modern databases. Our modern approach to hybrid, multicloud infrastructure and data management, which we term 'evolved cloud', provides customers the ability to leverage data across their entire estate with simplicity, security, and sustainability which increases our relevance and value to our customers.\n\n\nIn an evolved cloud state, the cloud is fully integrated into an organization's architecture and operations. Data centers and clouds are seamlessly united and hybrid multicloud operations are simplified, with consistency and observability across environments. The key benefits NetApp brings to an organization's hybrid multicloud environment are:\n\n\n•\nOperational simplicity: NetApp's use of open source, open architectures and APIs, microservices, and common capabilities and data services facilitate the creation of applications that can run anywhere.\n\n\n•\nFlexibility and consistency: NetApp makes moving data and applications between environments seamless through a common storage foundation across on-premises and multicloud environments.'}
```

Set up a LangChain RAG workflow to chat with the form

```
In [32]: neo4j_vector_store = Neo4jVector.from_existing_graph(
    embedding=OpenAIEmbeddings(),
    url=NEO4J_URI,
    username=NEO4J_USERNAME,
    password=NEO4J_PASSWORD,
    index_name=VECTOR_INDEX_NAME,
    node_label=VECTOR_NODE_LABEL,
    text_node_properties=[VECTOR_SOURCE_PROPERTY],
    embedding_node_property=VECTOR_EMBEDDING_PROPERTY,
)
```

```
In [33]: retriever = neo4j_vector_store.as_retriever()
```

- Set up a RetrievalQAWithSourcesChain to carry out question answering
- You can check out the LangChain documentation for this chain [here](https://api.python.langchain.com/en/latest/chains/langchain.chains.qa_with_sources.retrieval.Retrieve) (https://api.python.langchain.com/en/latest/chains/langchain.chains.qa_with_sources.retrieval.Retrieve)

```
In [34]: chain = RetrievalQAWithSourcesChain.from_chain_type(
          ChatOpenAI(temperature=0),
          chain_type="stuff",
          retriever=retriever
        )

In [35]: def prettychain(question: str) -> str:
          """Pretty print the chain's response to a question"""
          response = chain({"question": question},
                           return_only_outputs=True,)
          print(textwrap.fill(response['answer'], 60))
```

- Ask a question!

```
In [36]: question = "What is Netapp's primary business?"
```

```
In [37]: prettychain(question)
```

NetApp's primary business is enterprise storage and data management, cloud storage, and cloud operations.

```
In [38]: prettychain("Where is Netapp headquartered?")
```

Netapp is headquartered in San Jose, California.

```
In [39]: prettychain("""
          Tell me about Netapp.
          Limit your answer to a single sentence.
          """)
```

NetApp is a global cloud-led, data-centric software company that provides customers with the freedom to manage applications and data across hybrid multicloud environments.

```
In [40]: prettychain("""
          Tell me about Apple.
          Limit your answer to a single sentence.
          """)
```

Apple is a global cloud-led, data-centric software company headquartered in San Jose, California, that provides customers with the freedom to manage applications and data across hybrid multicloud environments.

```
In [41]: prettychain("""
          Tell me about Apple.
          Limit your answer to a single sentence.
          If you are unsure about the answer, say you don't know.
          """)
```

I don't know.

Ask you own question!

- Add your own question to the call to prettychain below to find out more about NetApp

- Here is NetApp's website if you want some inspiration: <https://www.netapp.com/>
(<https://www.netapp.com/>)

```
In [42]: ► prettychain("""  
          ADD YOUR OWN QUESTION HERE  
          """)
```

I don't know.