Vector Retriever

:order: 2
:type: lesson
:branch: main

In this lesson, you will create a vector retriever to retrieve relevant data from Neo4j.

A retriever is a component that takes unstructured data (typically a users query) and retrieves relevant data.

You will create a vector retriever that find similar movies based on a movie plot.
The retriever will use the `moviePlots` vector index you used to search for similar movies using Cypher.

To find similar movies using a retriever you need to:

. Connect to a Neo4j database
. Create an _embedder_ to convert users queries into vectors
. Create a _retriever_ that uses the `moviePlots` vector index
. Use the retriever to search for similar movies using the users query
. Parse the results

Open the `genai-fundamentals/vector_retriever.py` file and review the program:

[source,python]
.vector_retriever.py
----
include::{repository-raw}/{branch}/genai-fundamentals/vector_retriever.py[tag=**]

The programs includes the code to connect to the Neo4j database using the link:https://neo4j.com/docs/python-manual/current/[Neo4j Python driver^].

[TIP]
You can learn more about the Neo4j Python driver in the Graph Academy link:https://graphacademy.neo4j.com/courses/drivers-python/[Using Neo4j with Python^] course.

Embedder

Create the embedder that will convert the user's query into a vector:

[source,python]
----
include::{repository-raw}/{branch}/genai-fundamentals/solutions/vector_retriever.py[tag=import-embedder]

include::{repository-raw}/{branch}/genai-fundamentals/solutions/vector_retriever.py[tag=embedder]

== Retriever

Create the retriever that will use the `moviePlots` vector index:

[source,python]
----
include::{repository-raw}/{branch}/genai-fundamentals/solutions/vector_ret
riever.py[tag=import-retriever]

include::{repository-raw}/{branch}/genai-fundamentals/solutions/vector_ret
riever.py[tag=retriever]

The retriever allows you to specify what properties to return from the
nodes that match the query.

== Search

You can use the retriever to search the vector index by passing a query
and the number of results to return.
The retriever will use the embedder to convert the query into a vector to
use in the search.

Search for similar movies:

[source,python]
----
include::{repository-raw}/{branch}/genai-fundamentals/solutions/vector_ret
riever.py[tag=search]

The `search` method returns a list of `items` that match the query.

Iterate over the items and print the results:

[source,python]
----
include::{repository-raw}/{branch}/genai-fundamentals/solutions/vector_ret
riever.py[tag=print-results]

[%collapsible]
.Click to view the complete code
====
[source,python]
----
include::{repository-raw}/{branch}/genai-fundamentals/solutions/vector_ret
riever.py[tag=**]

====

Run the program to search for similar movies based on a query.

You should see movie `titles`, `plots`, and the similarity `score` for the
times found.

[%collapsible]
.Click to reveal a typical output
====
[source]

{'title': 'Toy Story', 'plot': "A cowboy doll is profoundly threatened and
jealous when a new spaceman figure supplants him as top toy in a boy's
room."} 0.9099578857421875
{'title': 'Pinocchio', 'plot': 'A living puppet, with the help of a
cricket as his conscience, must prove himself worthy to become a real
boy.'} 0.9085540771484375
{'title': 'Adventures of Pinocchio, The', 'plot': "One of puppet-maker
Geppetto's creations comes magically to life. This puppet, Pinocchio, has
one major desire and that is to become a real boy someday. In order to
accomplish this goal he ..."} 0.9070587158203125
{'title': 'Jumanji', 'plot': 'When two kids find and play a magical board
game, they release a man trapped for decades in it and a host of dangers
that can only be stopped by finishing the game.'} 0.9043426513671875
{'title': 'Secret Adventures of Tom Thumb, The', 'plot': 'A boy born the
size of a small doll is kidnapped by a genetic lab and must find a way
back to his father in this inventive adventure filmed using stop motion
animation techniques. Tom meets...'} 0.903472900390625

====

Experiment with different queries to find different movies.

== Check Your Understanding

include::questions/1-embedder.adoc[leveloffset=+1]

[.summary]
== Lesson Summary

In this lesson, you learned how to create a vector retriever using the

`neo4j-graphrag` package.

In the next module, you will build this retriever into a simple RAG pipeline that will use an LLM to answer questions using the retrieved data.