Keyword and Semantic Searches with ReRank

Khoi Duong
Prof. Chang
CS589
4/10/2024

1. Keyword and Semantic Searches with ReRank
    - Project Implementation
        - ReRank
            - ReRank
    - Process for the project documentation
        - Step 1: Adding the project to your portofolio
            1. Please use Google Slides to document the project
                - Copy from a Google Slides file and mofigy the file, but still keep the original Google Slides file.
            2. Please link your presentation on GitHub using this structure

```
Generative AI
  - Fine-Tuning
    + Keyword and Semantic Searches with ReRank
```

        - Step 2: Submit
            1. The URLs of the Google Slides and GitHub web pages related to this project.
            2. A PDF file of your Google Slides

Project Implementation

- Step 1: Import two libraries: cohere and weaviate
- Step 2: Apply Dense Retrieval to a query
- Step 3: Improving Keyword Search with ReRank
- Step 4: Improving Dense Retrieval with ReRank

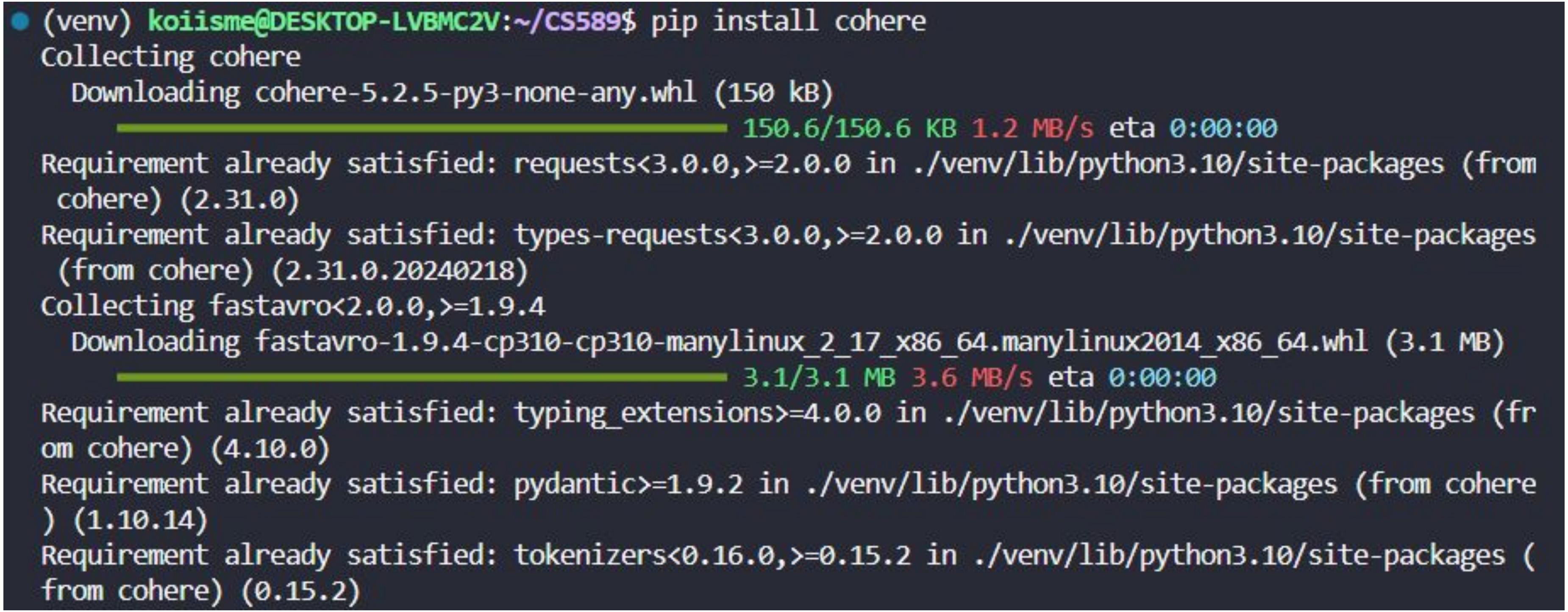
Get env variable needed for ReRank

Before we start, we need to set up the environment and get the env variable for the program, including WEAVIATE_API_KEY, WEAVIATE_API_URL, and COHERE_API_KEY

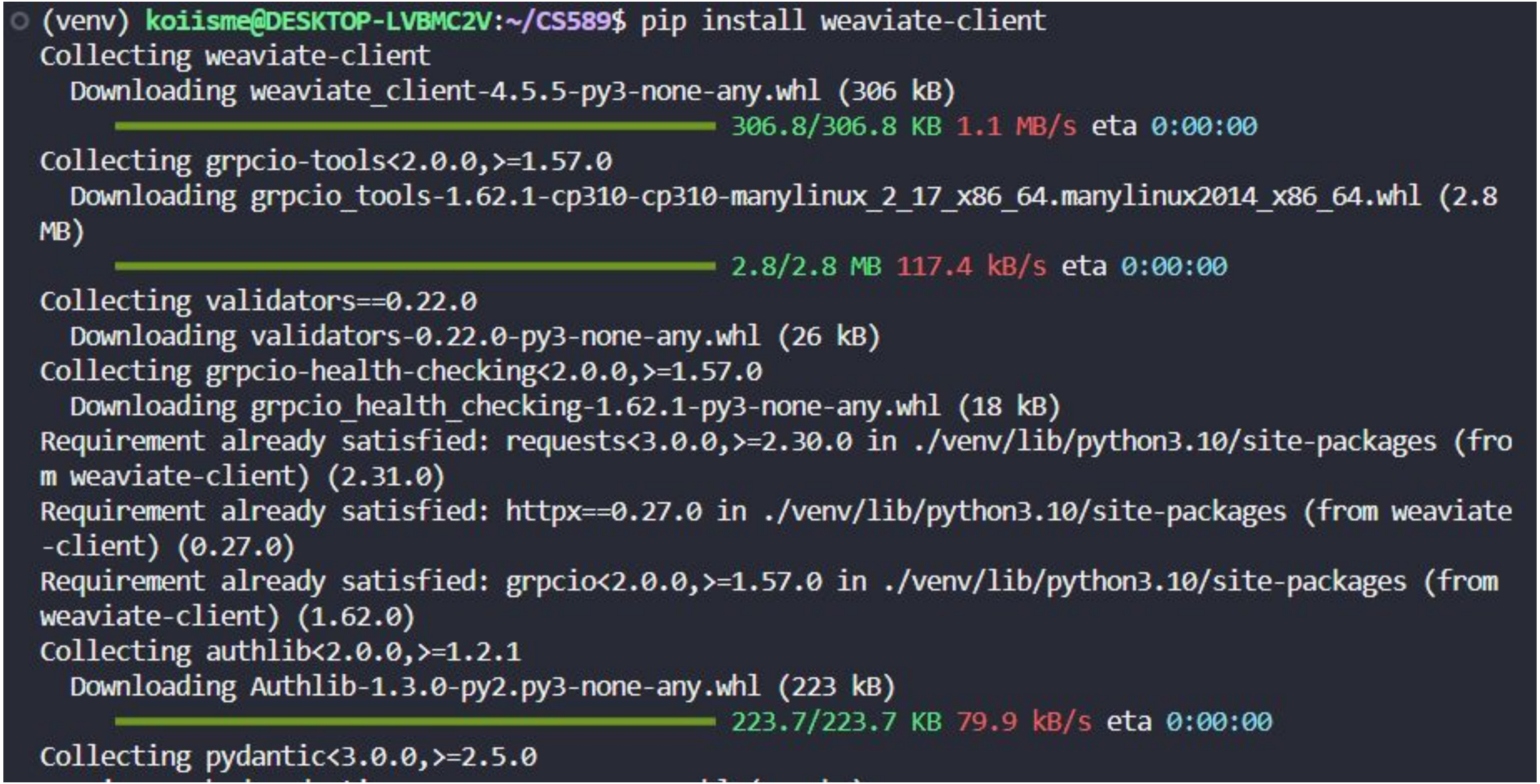
We start by installing cohere and weaviate-client with pip:

- pip install cohere
- pip install weaviate-client

We can also use pip to install any missing modules later on when running the program.



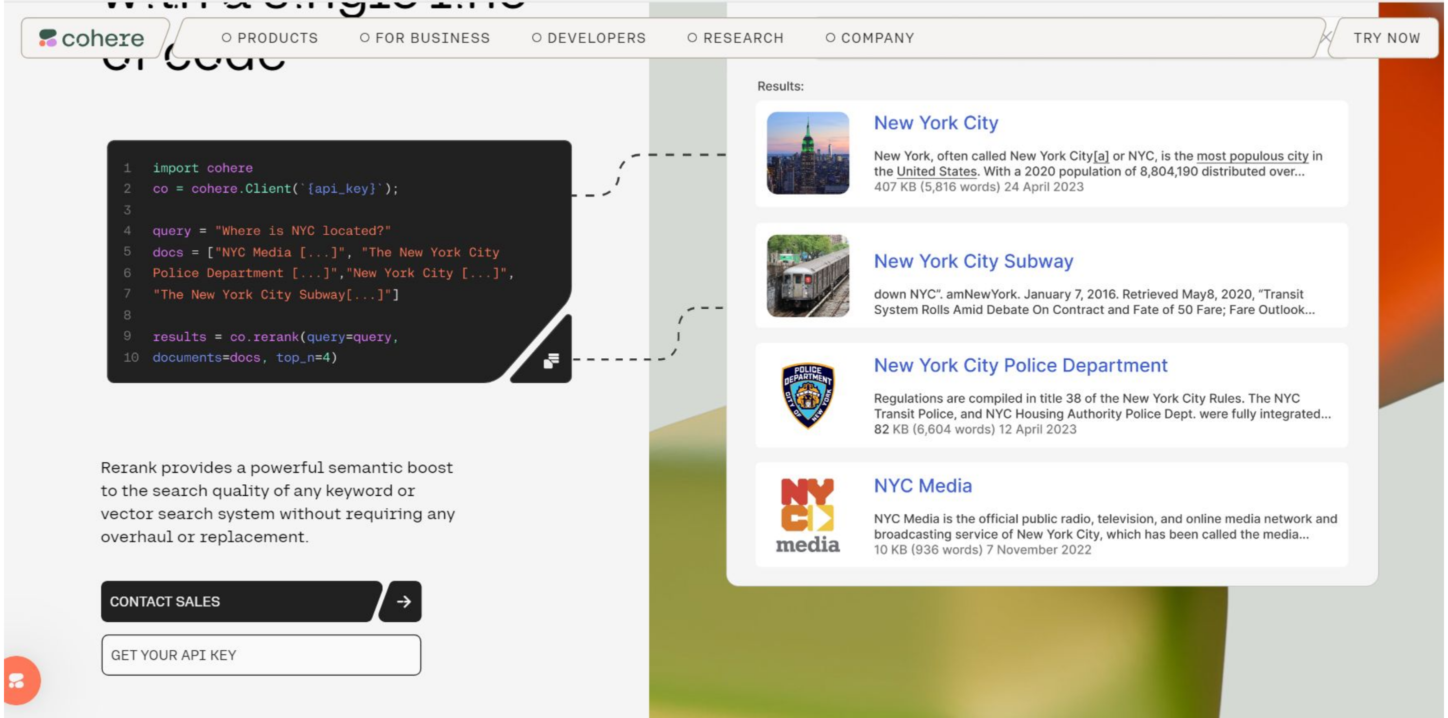
```
● (venv) koiisme@DESKTOP-LVBMC2V:~/CS589$ pip install cohere
  Collecting cohere
    Downloading cohere-5.2.5-py3-none-any.whl (150 kB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 150.6/150.6 KB 1.2 MB/s eta 0:00:00
  Requirement already satisfied: requests<3.0.0,>=2.0.0 in ./venv/lib/python3.10/site-packages (from
   cohere) (2.31.0)
  Requirement already satisfied: types-requests<3.0.0,>=2.0.0 in ./venv/lib/python3.10/site-packages
   (from cohere) (2.31.0.20240218)
  Collecting fastavro<2.0.0,>=1.9.4
    Downloading fastavro-1.9.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.1 MB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.1/3.1 MB 3.6 MB/s eta 0:00:00
  Requirement already satisfied: typing_extensions>=4.0.0 in ./venv/lib/python3.10/site-packages (fr
  om cohere) (4.10.0)
  Requirement already satisfied: pydantic>=1.9.2 in ./venv/lib/python3.10/site-packages (from cohere
  ) (1.10.14)
  Requirement already satisfied: tokenizers<0.16.0,>=0.15.2 in ./venv/lib/python3.10/site-packages (
  from cohere) (0.15.2)
```



```
○ (venv) koiisme@DESKTOP-LVBMC2V:~/CS589$ pip install weaviate-client
  Collecting weaviate-client
    Downloading weaviate_client-4.5.5-py3-none-any.whl (306 kB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 306.8/306.8 KB 1.1 MB/s eta 0:00:00
  Collecting grpcio-tools<2.0.0,>=1.57.0
    Downloading grpcio_tools-1.62.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.8
  MB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.8/2.8 MB 117.4 kB/s eta 0:00:00
  Collecting validators==0.22.0
    Downloading validators-0.22.0-py3-none-any.whl (26 kB)
  Collecting grpcio-health-checking<2.0.0,>=1.57.0
    Downloading grpcio_health_checking-1.62.1-py3-none-any.whl (18 kB)
  Requirement already satisfied: requests<3.0.0,>=2.30.0 in ./venv/lib/python3.10/site-packages (fro
  m weaviate-client) (2.31.0)
  Requirement already satisfied: httpx==0.27.0 in ./venv/lib/python3.10/site-packages (from weaviate
  -client) (0.27.0)
  Requirement already satisfied: grpcio<2.0.0,>=1.57.0 in ./venv/lib/python3.10/site-packages (from
  weaviate-client) (1.62.0)
  Collecting authlib<2.0.0,>=1.2.1
    Downloading Authlib-1.3.0-py2.py3-none-any.whl (223 kB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 223.7/223.7 KB 79.9 kB/s eta 0:00:00
  Collecting pydantic<3.0.0,>=2.5.0
```

Getting API key from Cohere and Weaviate

For Cohere, go to this link https://cohere.com/rerank to sign up an account



```
import cohere
co = cohere.Client(`{api_key}`);

query = "Where is NYC located?"
docs = ["NYC Media [...]", "The New York City
Police Department [...]","New York City [...]",
"The New York City Subway[...]"]

results = co.rerank(query=query,
documents=docs, top_n=4)
```

Results:

New York City

New York, often called New York City[a] or NYC, is the most populous city in the United States. With a 2020 population of 8,804,190 distributed over...
407 KB (5,816 words) 24 April 2023

New York City Subway

down NYC". amNewYork. January 7, 2016. Retrieved May8, 2020, "Transit System Rolls Amid Debate On Contract and Fate of 50 Fare; Fare Outlook...

New York City Police Department

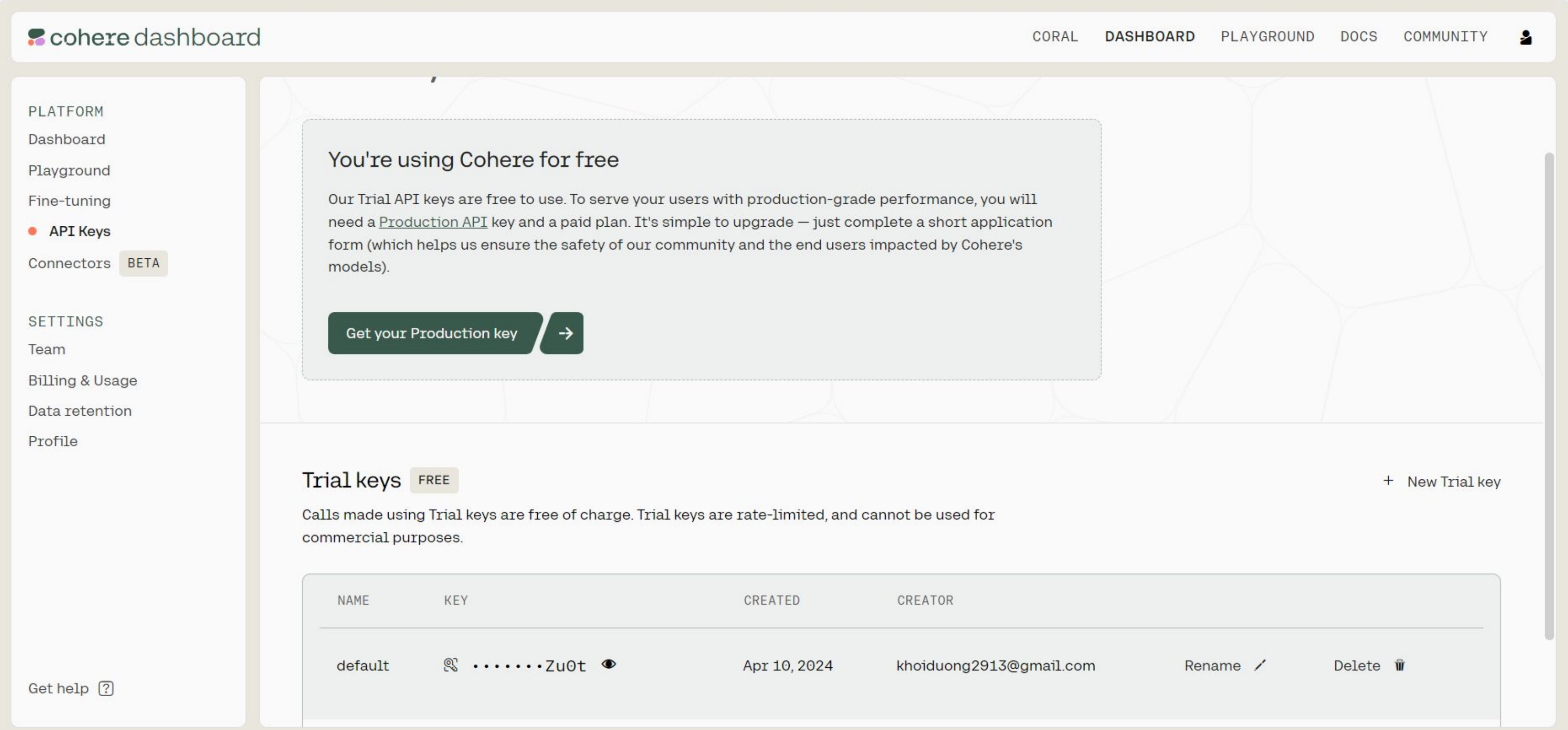
Regulations are compiled in title 38 of the New York City Rules. The NYC Transit Police, and NYC Housing Authority Police Dept. were fully integrated...
82 KB (6,604 words) 12 April 2023

NYC Media

NYC Media is the official public radio, television, and online media network and broadcasting service of New York City, which has been called the media...
10 KB (936 words) 7 November 2022

Rerank provides a powerful semantic boost to the search quality of any keyword or vector search system without requiring any overhaul or replacement.

After creating an account, head to https://dashboard.cohere.com/api-keys to see our private API key.



You're using Cohere for free

Our Trial API keys are free to use. To serve your users with production-grade performance, you will need a Production API key and a paid plan. It's simple to upgrade — just complete a short application form (which helps us ensure the safety of our community and the end users impacted by Cohere's models).

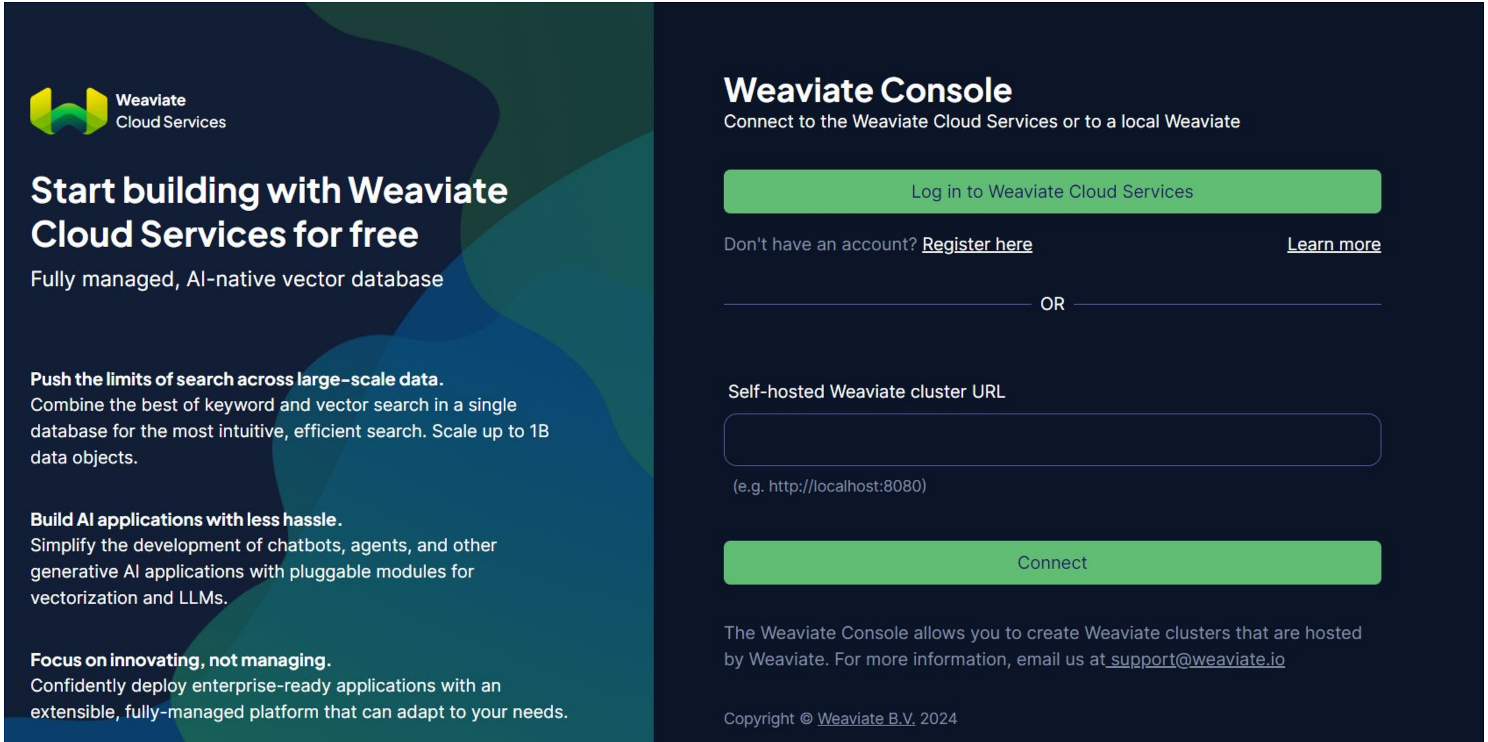
Trial keys FREE

Calls made using Trial keys are free of charge. Trial keys are rate-limited, and cannot be used for commercial purposes.

NAME	KEY	CREATED	CREATOR		
default	•••••••Zu0t	Apr 10, 2024	khoiduong2913@gmail.com	Rename	Delete

Weaviate API key and URL

For Weaviate, go to https://weaviate.io/ and sign up an account.



Weaviate Cloud Services

Start building with Weaviate Cloud Services for free

Fully managed, AI-native vector database

Push the limits of search across large-scale data.
Combine the best of keyword and vector search in a single database for the most intuitive, efficient search. Scale up to 1B data objects.

Build AI applications with less hassle.
Simplify the development of chatbots, agents, and other generative AI applications with pluggable modules for vectorization and LLMs.

Focus on innovating, not managing.
Confidently deploy enterprise-ready applications with an extensible, fully-managed platform that can adapt to your needs.

Weaviate Console

Connect to the Weaviate Cloud Services or to a local Weaviate

Don't have an account? Register here

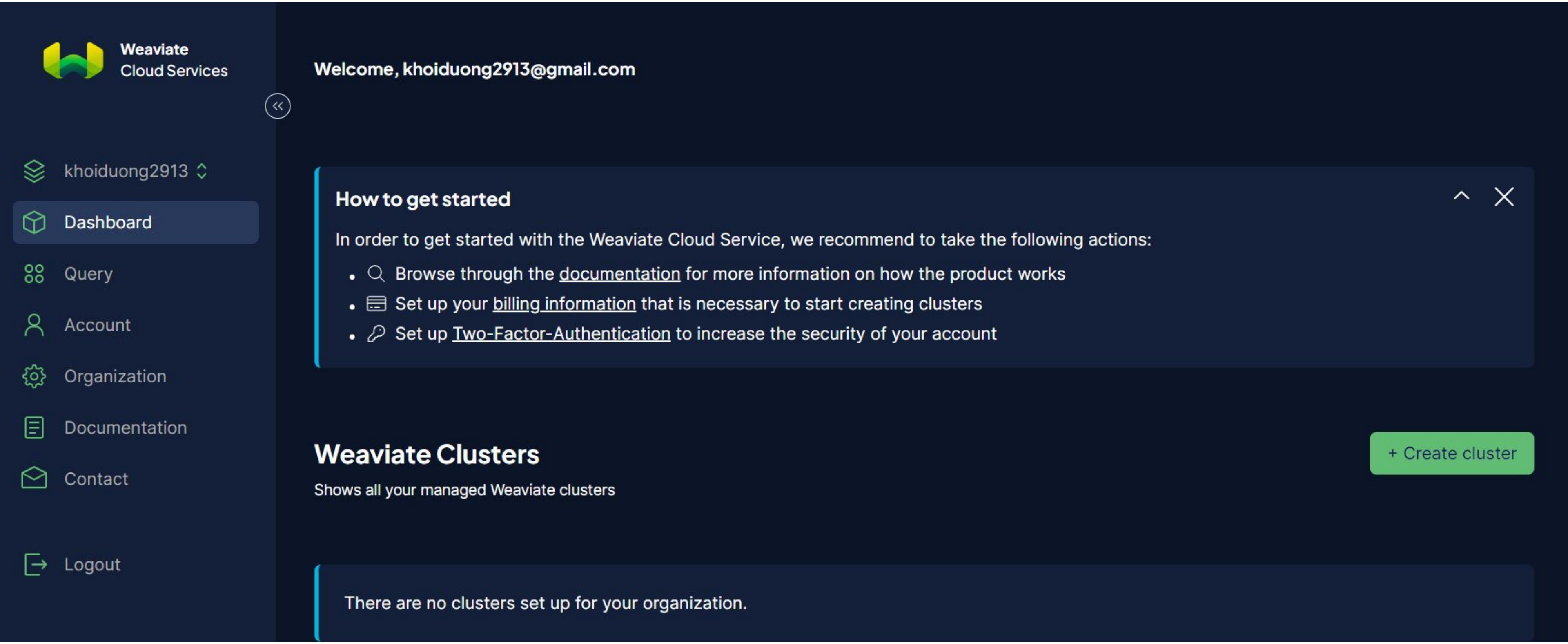
Self-hosted Weaviate cluster URL

(e.g. http://localhost:8080)

The Weaviate Console allows you to create Weaviate clusters that are hosted by Weaviate. For more information, email us at support@weaviate.io

Copyright © Weaviate B.V. 2024

Then, go to https://console.weaviate.cloud/dashboard and create a free cluster



Welcome, khoiduong2913@gmail.com

How to get started

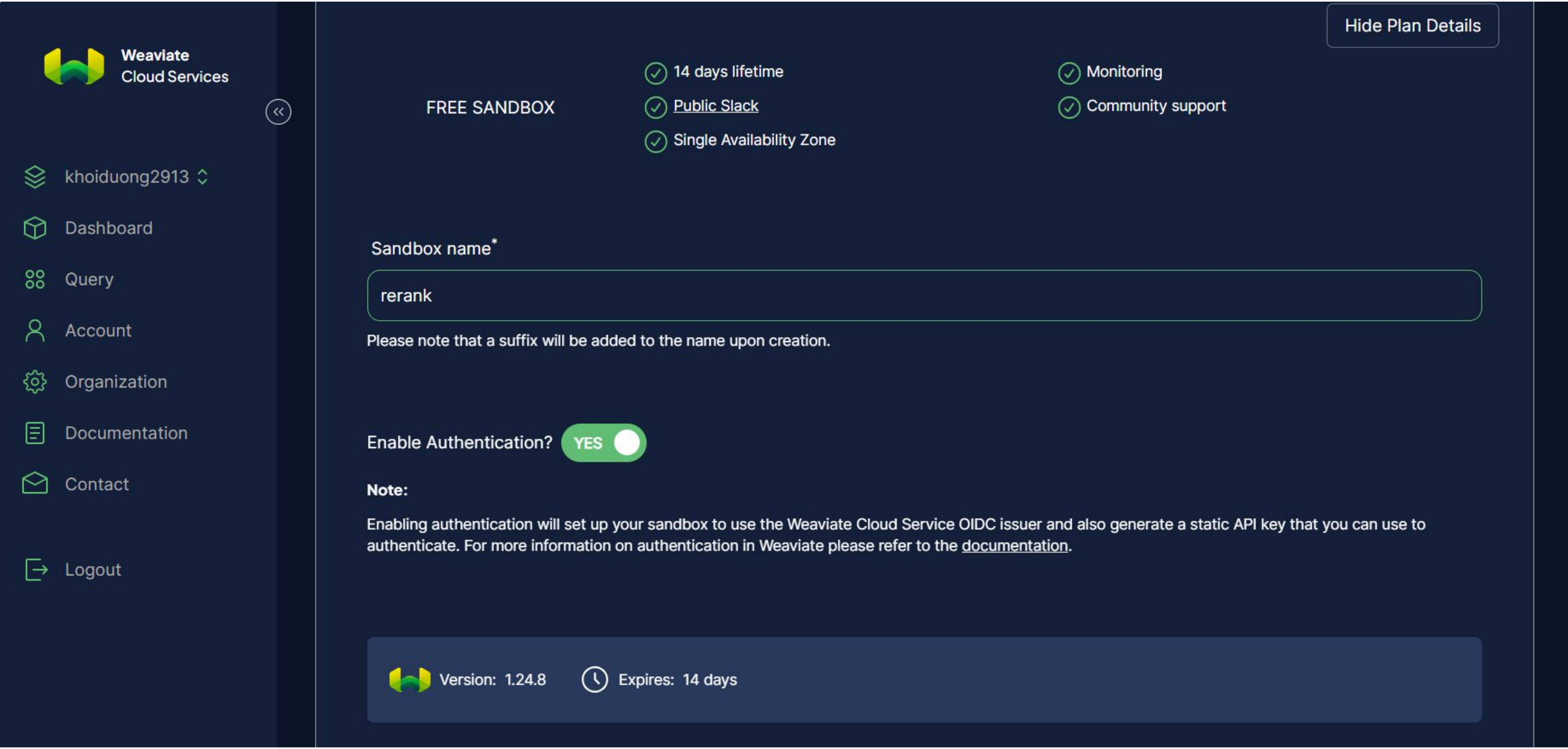
In order to get started with the Weaviate Cloud Service, we recommend to take the following actions:

- Browse through the documentation for more information on how the product works
- Set up your billing information that is necessary to start creating clusters
- Set up Two-Factor-Authentication to increase the security of your account

Weaviate Clusters

Shows all your managed Weaviate clusters

There are no clusters set up for your organization.



FREE SANDBOX

✓ 14 days lifetime

✓ Public Slack

✓ Single Availability Zone

✓ Monitoring

✓ Community support

Sandbox name*

rerank

Please note that a suffix will be added to the name upon creation.

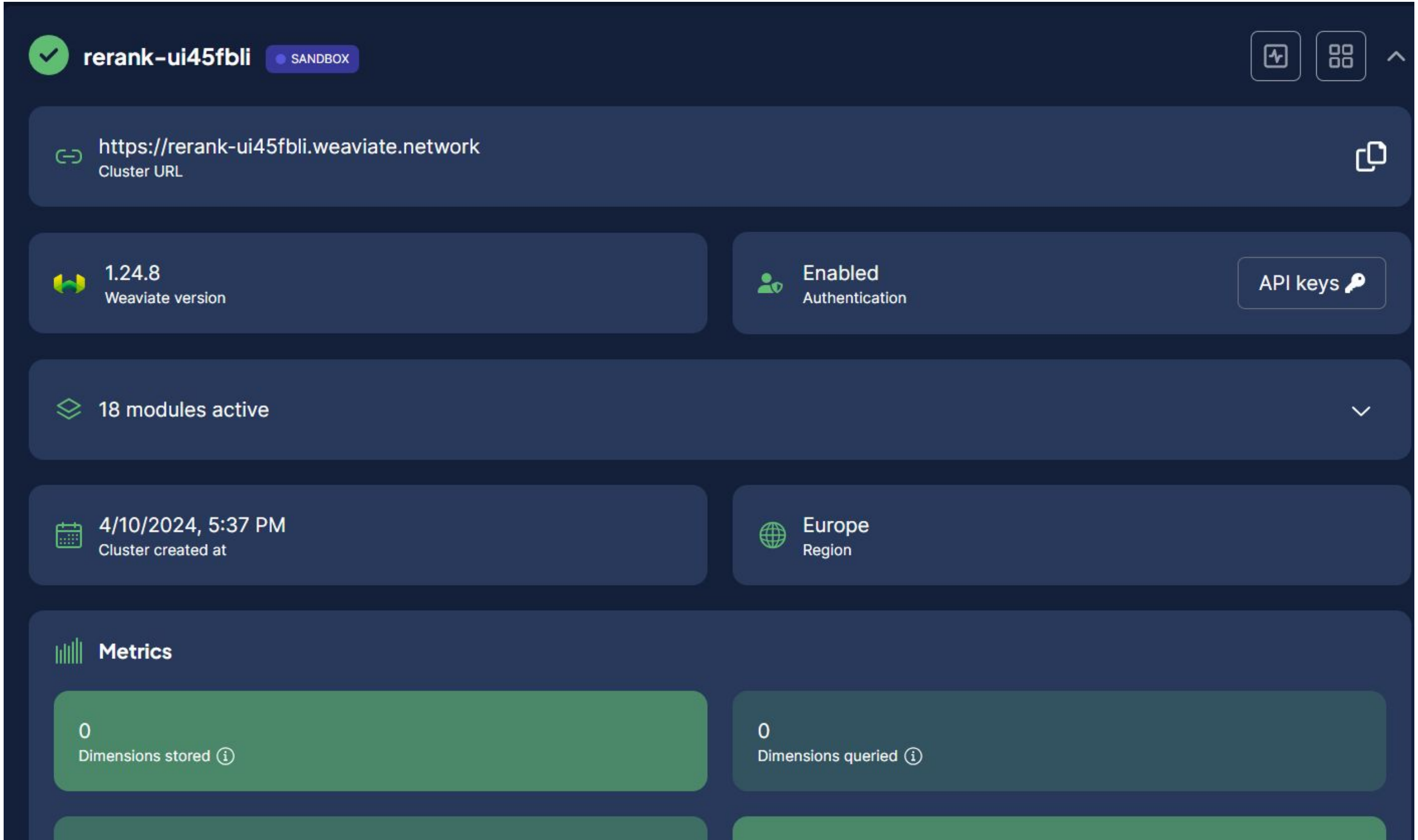
Enable Authentication? YES

Note:

Enabling authentication will set up your sandbox to use the Weaviate Cloud Service OIDC issuer and also generate a static API key that you can use to authenticate. For more information on authentication in Weaviate please refer to the documentation.

Version: 1.24.8 Expires: 14 days

We can check the cluster’s info right after we created it.



rerank-ui45fbli SANDBOX

https://rerank-ui45fbli.weaviate.network
Cluster URL

1.24.8
Weaviate version

Enabled
Authentication

18 modules active

4/10/2024, 5:37 PM
Cluster created at

Europe
Region

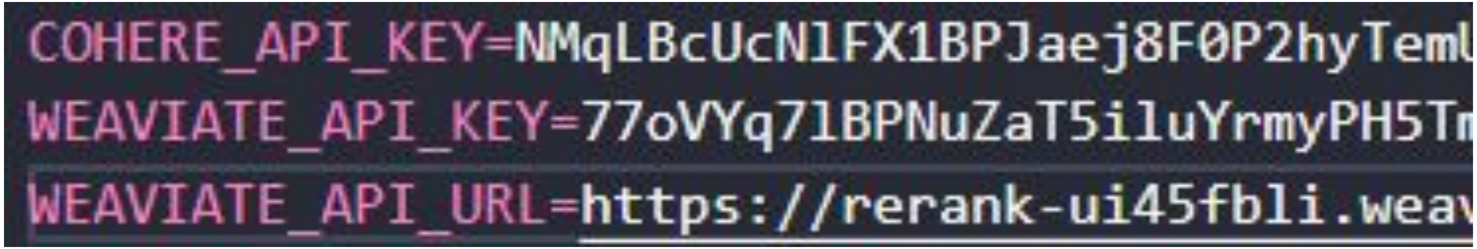
Metrics

0
Dimensions stored

0
Dimensions queried

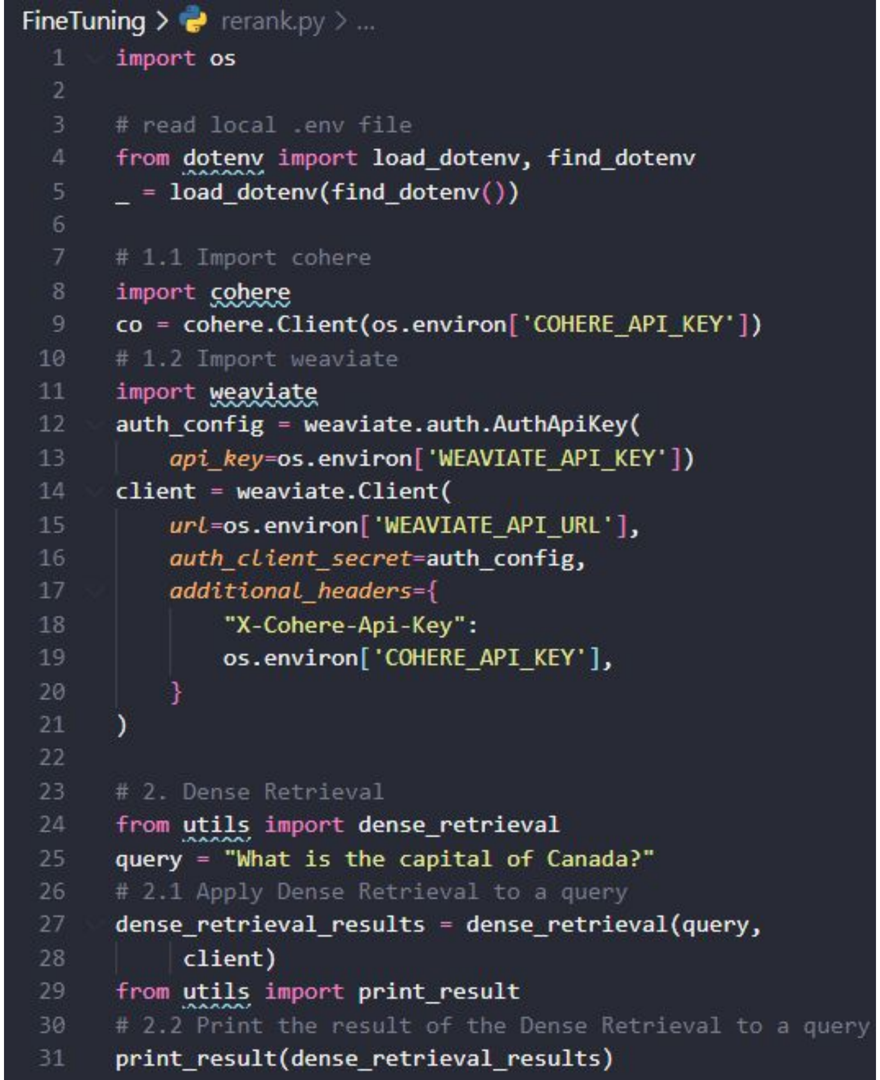
Get API key and URL ready

From the picture above, we can get the WEAVIATE_API_KEY and WEAVIATE_API_URL

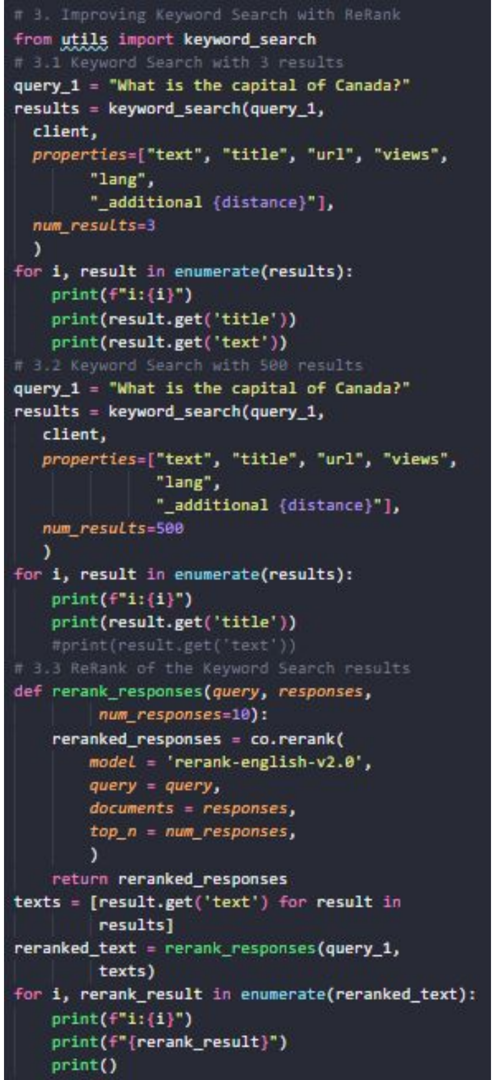


```
COHERE_API_KEY=NMqLBcUcN1FX1BPJaej8F0P2hyTemU
WEAVIATE_API_KEY=77oVYq71BPNuZaT5iluYrmyPH5Tm
WEAVIATE_API_URL=https://rerank-ui45fbli.weav
```

Source code

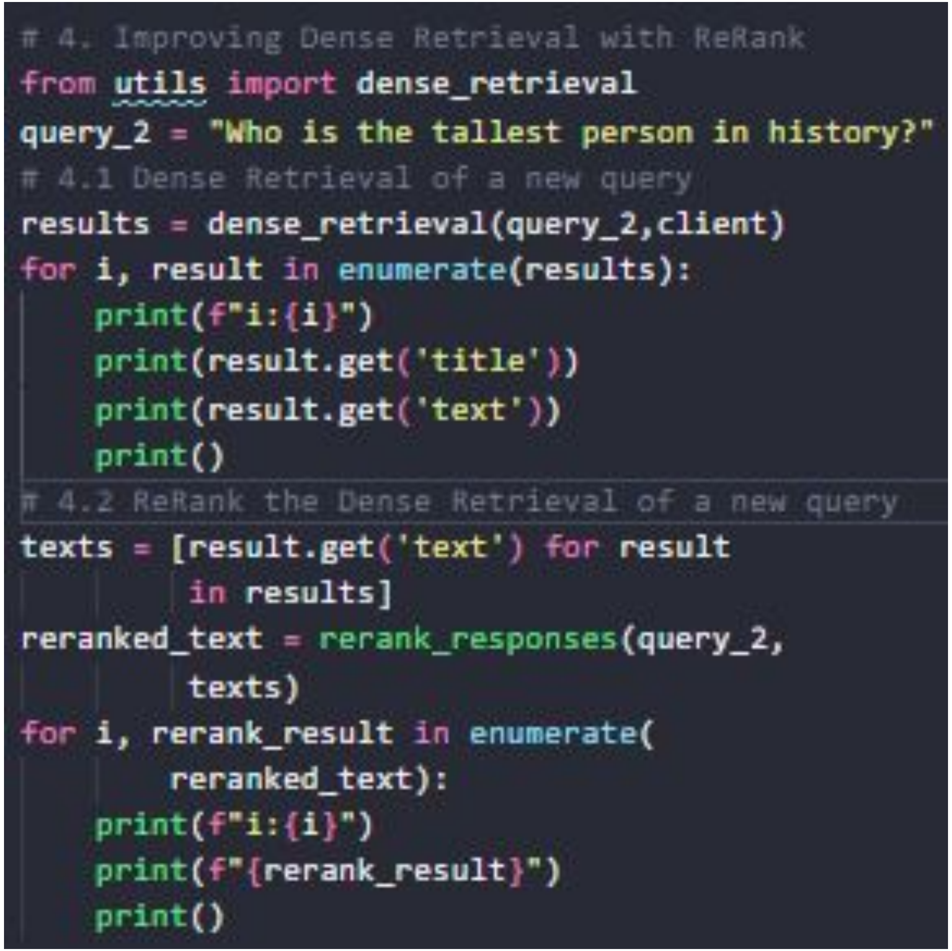


```python
import os

# read local .env file
from dotenv import load_dotenv, find_dotenv
_ = load_dotenv(find_dotenv())

# 1.1 Import cohere
import cohere
co = cohere.Client(os.environ['COHERE_API_KEY'])
# 1.2 Import weaviate
import weaviate
auth_config = weaviate.auth.AuthApiKey(
    api_key=os.environ['WEAVIATE_API_KEY'])
client = weaviate.Client(
    url=os.environ['WEAVIATE_API_URL'],
    auth_client_secret=auth_config,
    additional_headers={
        "X-Cohere-Api-Key":
        os.environ['COHERE_API_KEY'],
    }
)

# 2. Dense Retrieval
from utils import dense_retrieval
query = "What is the capital of Canada?"
# 2.1 Apply Dense Retrieval to a query
dense_retrieval_results = dense_retrieval(query,
    client)
from utils import print_result
# 2.2 Print the result of the Dense Retrieval to a query
print_result(dense_retrieval_results)
```



```python
# 3. Improving Keyword Search with ReRank
from utils import keyword_search
# 3.1 Keyword Search with 3 results
query_1 = "What is the capital of Canada?"
results = keyword_search(query_1,
  client,
  properties=["text", "title", "url", "views",
        "lang",
        "_additional {distance}"],
  num_results=3
  )
for i, result in enumerate(results):
    print(f"i:{i}")
    print(result.get('title'))
    print(result.get('text'))
# 3.2 Keyword Search with 500 results
query_1 = "What is the capital of Canada?"
results = keyword_search(query_1,
   client,
   properties=["text", "title", "url", "views",
               "lang",
               "_additional {distance}"],
   num_results=500
   )
for i, result in enumerate(results):
    print(f"i:{i}")
    print(result.get('title'))
    #print(result.get('text'))
# 3.3 ReRank of the Keyword Search results
def rerank_responses(query, responses,
        num_responses=10):
    reranked_responses = co.rerank(
        model = 'rerank-english-v2.0',
        query = query,
        documents = responses,
        top_n = num_responses,
        )
    return reranked_responses
texts = [result.get('text') for result in
        results]
reranked_text = rerank_responses(query_1,
        texts)
for i, rerank_result in enumerate(reranked_text):
    print(f"i:{i}")
    print(f"{rerank_result}")
    print()
```

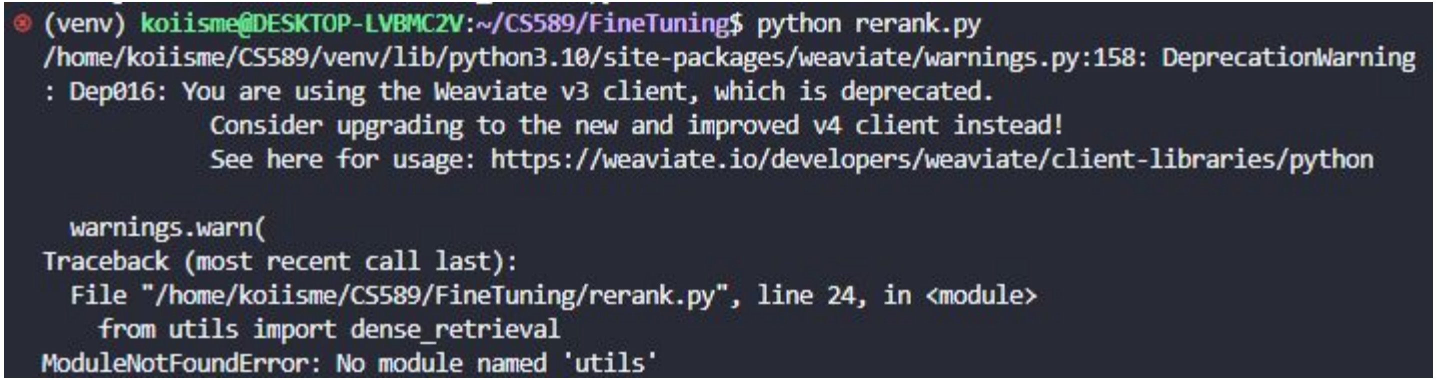
Source code (cont)



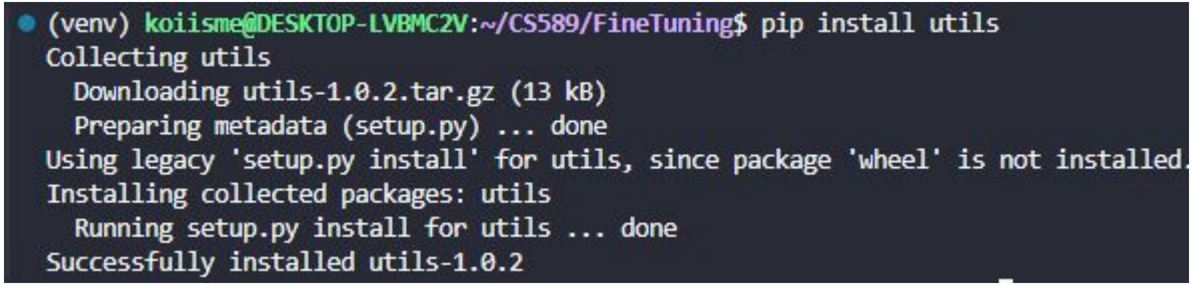
```
# 4. Improving Dense Retrieval with ReRank
from utils import dense_retrieval
query_2 = "Who is the tallest person in history?"
# 4.1 Dense Retrieval of a new query
results = dense_retrieval(query_2,client)
for i, result in enumerate(results):
    print(f"i:{i}")
    print(result.get('title'))
    print(result.get('text'))
    print()
# 4.2 ReRank the Dense Retrieval of a new query
texts = [result.get('text') for result
         in results]
reranked_text = rerank_responses(query_2,
         texts)
for i, rerank_result in enumerate(
        reranked_text):
    print(f"i:{i}")
    print(f"{rerank_result}")
    print()
```

Run the program

There is a missing module named ‘utils’. We need to install it.



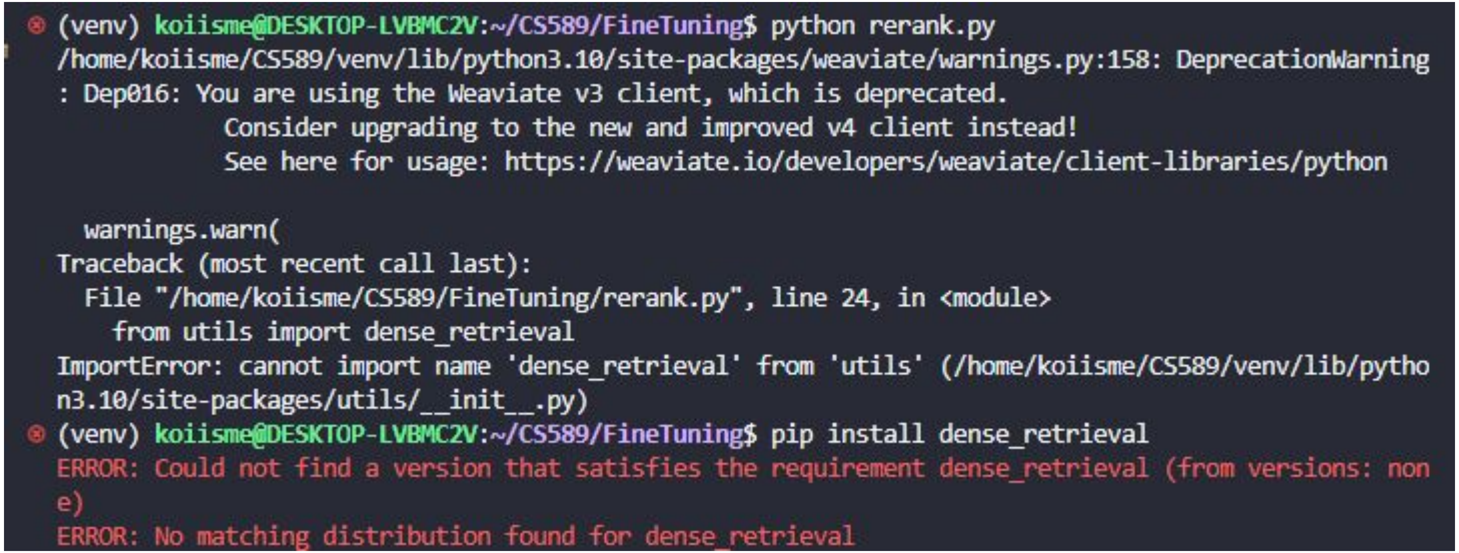
```
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$ python rerank.py
/home/koiisme/CS589/venv/lib/python3.10/site-packages/weaviate/warnings.py:158: DeprecationWarning
: Dep016: You are using the Weaviate v3 client, which is deprecated.
            Consider upgrading to the new and improved v4 client instead!
            See here for usage: https://weaviate.io/developers/weaviate/client-libraries/python

  warnings.warn(
Traceback (most recent call last):
  File "/home/koiisme/CS589/FineTuning/rerank.py", line 24, in <module>
    from utils import dense_retrieval
ModuleNotFoundError: No module named 'utils'
```



```
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$ pip install utils
Collecting utils
  Downloading utils-1.0.2.tar.gz (13 kB)
  Preparing metadata (setup.py) ... done
Using legacy 'setup.py install' for utils, since package 'wheel' is not installed.
Installing collected packages: utils
  Running setup.py install for utils ... done
Successfully installed utils-1.0.2
```

dense_retrieval()

Another problem occurs, it indicates that there is no dense_retrieval in utils module, and there is no module named dense_retrieval

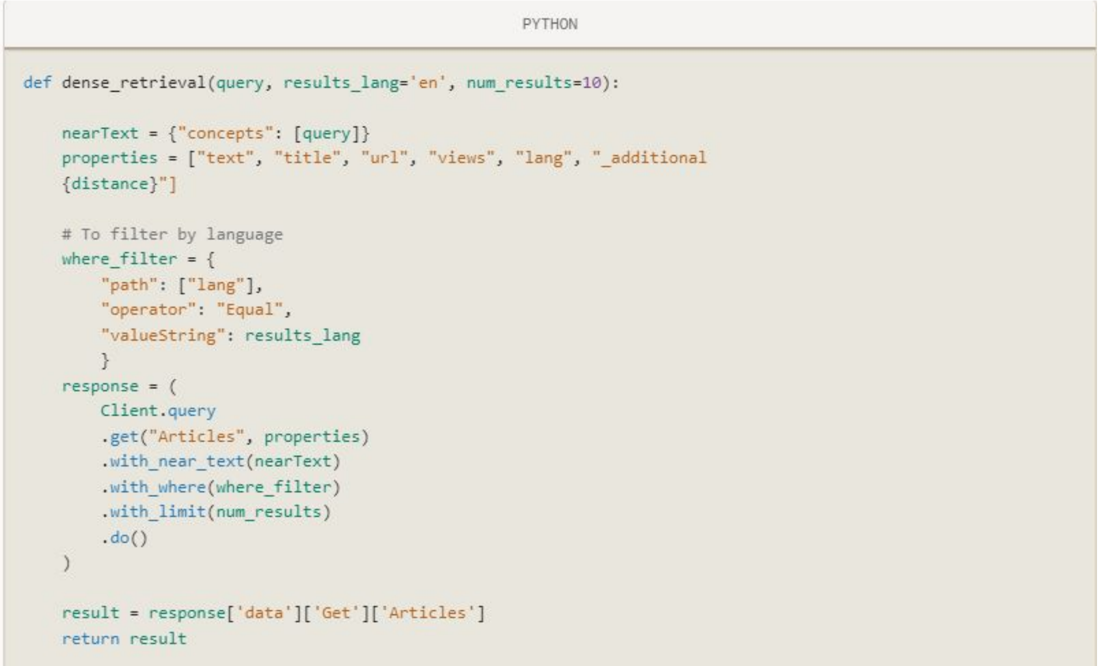


```
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$ python rerank.py
/home/koiisme/CS589/venv/lib/python3.10/site-packages/weaviate/warnings.py:158: DeprecationWarning
: Dep016: You are using the Weaviate v3 client, which is deprecated.
            Consider upgrading to the new and improved v4 client instead!
            See here for usage: https://weaviate.io/developers/weaviate/client-libraries/python

  warnings.warn(
Traceback (most recent call last):
  File "/home/koiisme/CS589/FineTuning/rerank.py", line 24, in <module>
    from utils import dense_retrieval
ImportError: cannot import name 'dense_retrieval' from 'utils' (/home/koiisme/CS589/venv/lib/pytho
n3.10/site-packages/utils/__init__.py)
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$ pip install dense_retrieval
ERROR: Could not find a version that satisfies the requirement dense_retrieval (from versions: non
e)
ERROR: No matching distribution found for dense_retrieval
```

Solution

I did some researches and looked up on the Internet, and from Dense Retrieval document from Cohere, I realized that dense_retrieval is a function that maybe put in the utils.py, which is not mentioned here in the source code.

Therefore, we just need to add the function ***dense_retrieval*** into utils.py to import later.



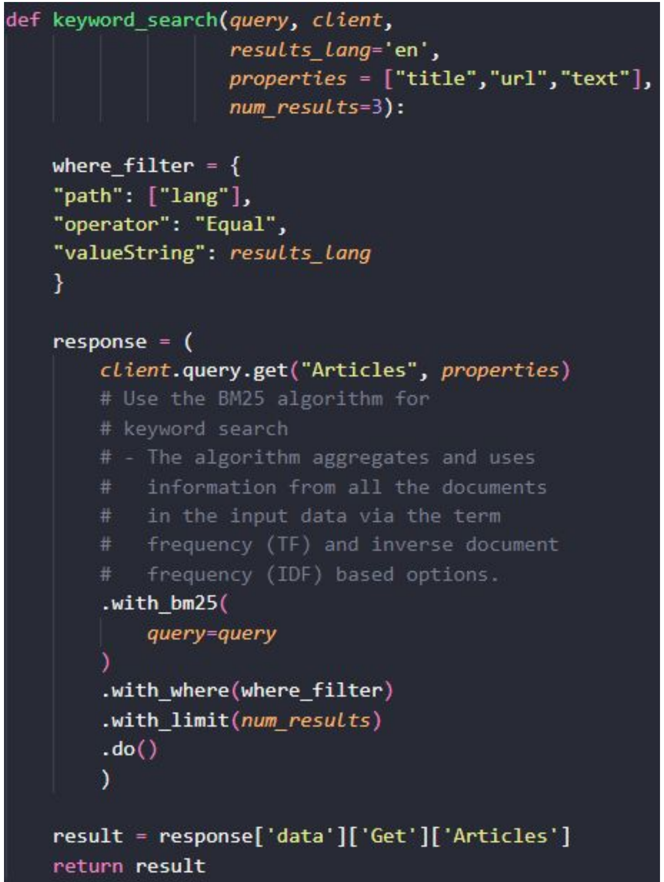
PYTHON

```python
def dense_retrieval(query, results_lang='en', num_results=10):

    nearText = {"concepts": [query]}
    properties = ["text", "title", "url", "views", "lang", "_additional
    {distance}"]

    # To filter by language
    where_filter = {
        "path": ["lang"],
        "operator": "Equal",
        "valueString": results_lang
        }
    response = (
        Client.query
        .get("Articles", properties)
        .with_near_text(nearText)
        .with_where(where_filter)
        .with_limit(num_results)
        .do()
    )

    result = response['data']['Get']['Articles']
    return result
```

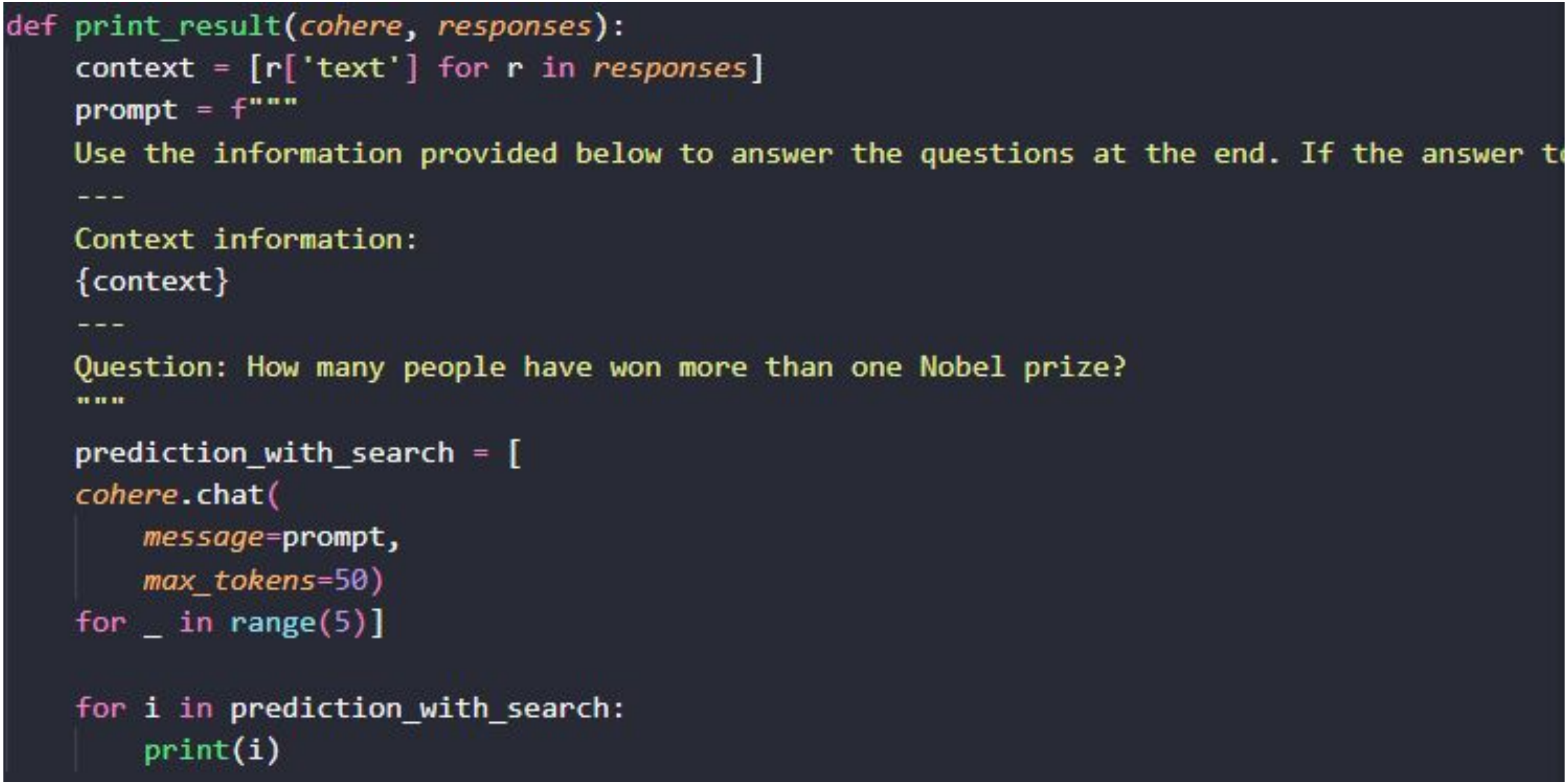
Update utils.py with print_result and keyword_search

Similarly, we need to implement print_result() and keyword_search() to utils.py

- Keyword Search
- Generating Answers
- Semantic Search



```
def keyword_search(query, client,
                   results_lang='en',
                   properties = ["title","url","text"],
                   num_results=3):

    where_filter = {
    "path": ["lang"],
    "operator": "Equal",
    "valueString": results_lang
    }

    response = (
        client.query.get("Articles", properties)
        # Use the BM25 algorithm for
        # keyword search
        # - The algorithm aggregates and uses
        #   information from all the documents
        #   in the input data via the term
        #   frequency (TF) and inverse document
        #   frequency (IDF) based options.
        .with_bm25(
            query=query
        )
        .with_where(where_filter)
        .with_limit(num_results)
        .do()
        )

    result = response['data']['Get']['Articles']
    return result
```

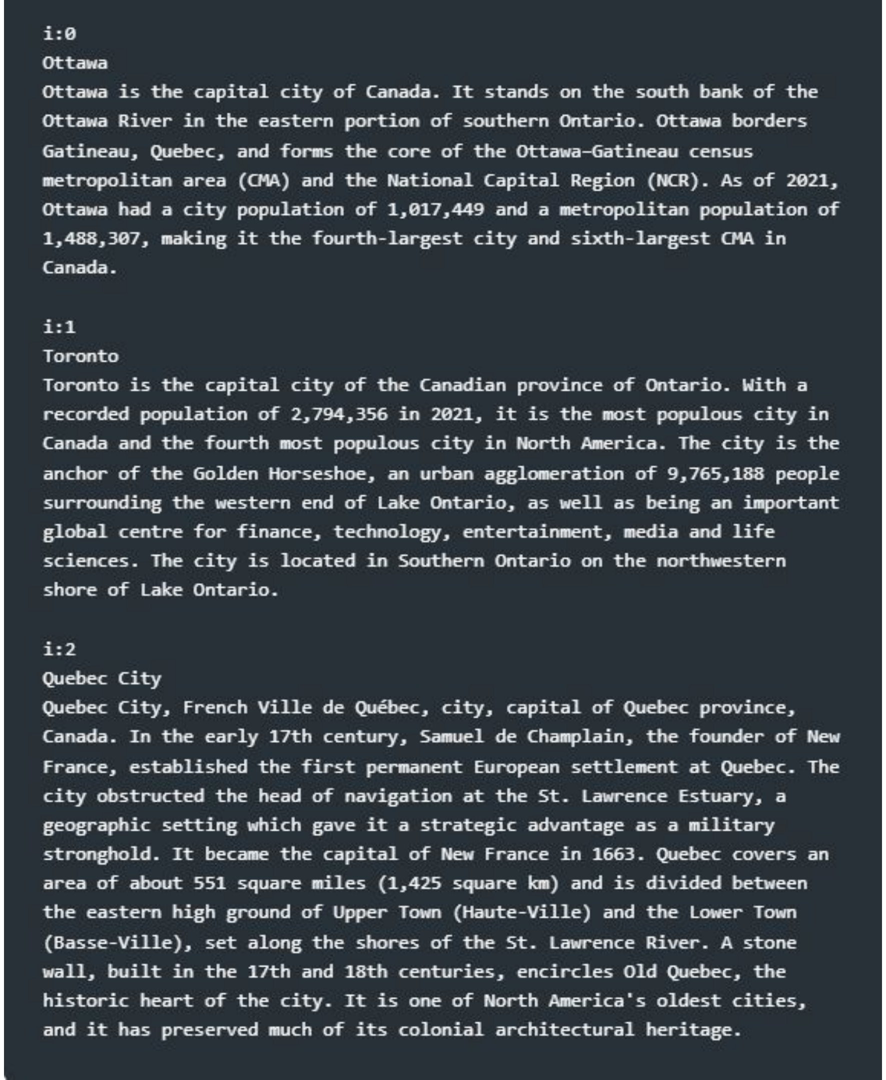


```python
def print_result(cohere, responses):
    context = [r['text'] for r in responses]
    prompt = f"""
    Use the information provided below to answer the questions at the end. If the answer to
    ---
    Context information:
    {context}
    ---
    Question: How many people have won more than one Nobel prize?
    """
    prediction_with_search = [
    cohere.chat(
        message=prompt,
        max_tokens=50)
    for _ in range(5)]

    for i in prediction_with_search:
        print(i)
```

Output result

#2 - Dense_retrieval on query:

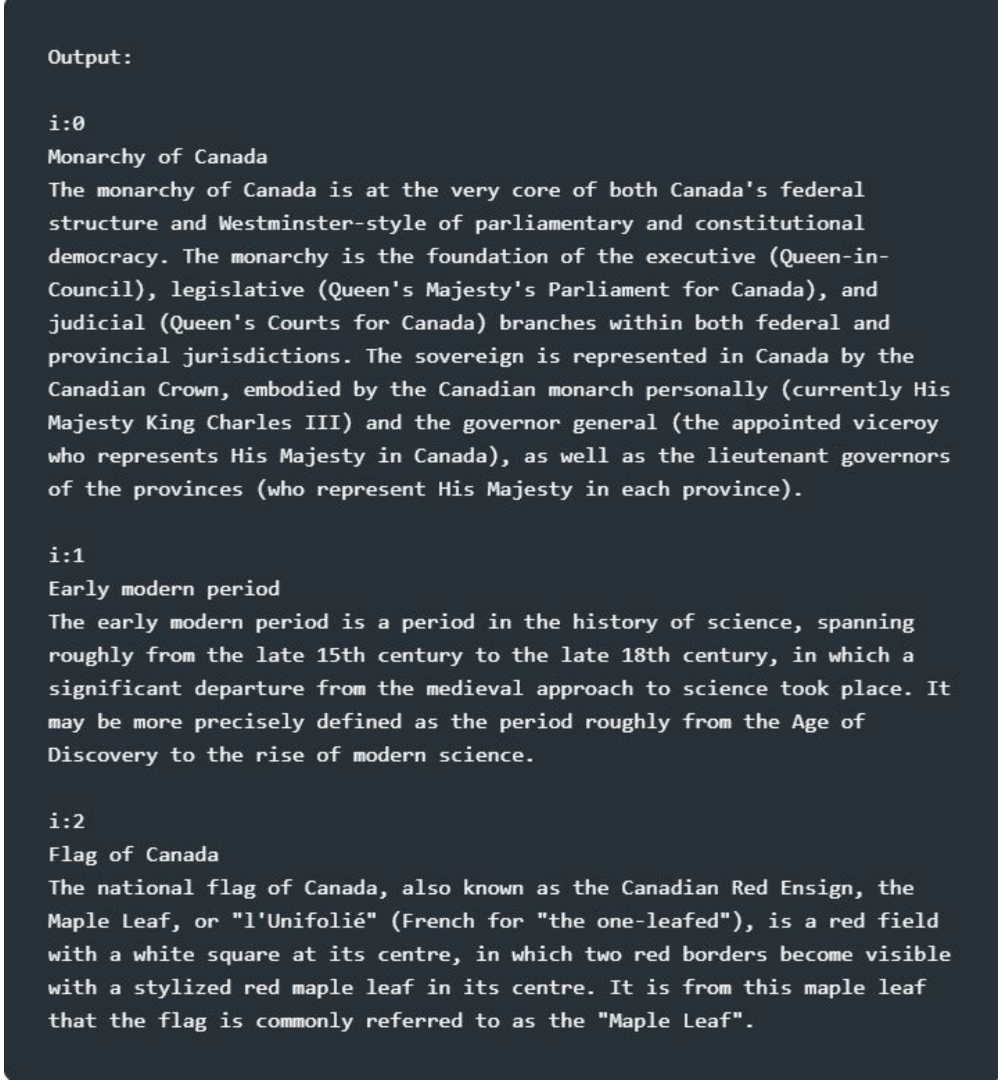
“What is the capital of Canada?”



```
i:0
Ottawa
Ottawa is the capital city of Canada. It stands on the south bank of the
Ottawa River in the eastern portion of southern Ontario. Ottawa borders
Gatineau, Quebec, and forms the core of the Ottawa-Gatineau census
metropolitan area (CMA) and the National Capital Region (NCR). As of 2021,
Ottawa had a city population of 1,017,449 and a metropolitan population of
1,488,307, making it the fourth-largest city and sixth-largest CMA in
Canada.

i:1
Toronto
Toronto is the capital city of the Canadian province of Ontario. With a
recorded population of 2,794,356 in 2021, it is the most populous city in
Canada and the fourth most populous city in North America. The city is the
anchor of the Golden Horseshoe, an urban agglomeration of 9,765,188 people
surrounding the western end of Lake Ontario, as well as being an important
global centre for finance, technology, entertainment, media and life
sciences. The city is located in Southern Ontario on the northwestern
shore of Lake Ontario.

i:2
Quebec City
Quebec City, French Ville de Québec, city, capital of Quebec province,
Canada. In the early 17th century, Samuel de Champlain, the founder of New
France, established the first permanent European settlement at Quebec. The
city obstructed the head of navigation at the St. Lawrence Estuary, a
geographic setting which gave it a strategic advantage as a military
stronghold. It became the capital of New France in 1663. Quebec covers an
area of about 551 square miles (1,425 square km) and is divided between
the eastern high ground of Upper Town (Haute-Ville) and the Lower Town
(Basse-Ville), set along the shores of the St. Lawrence River. A stone
wall, built in the 17th and 18th centuries, encircles Old Quebec, the
historic heart of the city. It is one of North America's oldest cities,
and it has preserved much of its colonial architectural heritage.
```

Output result

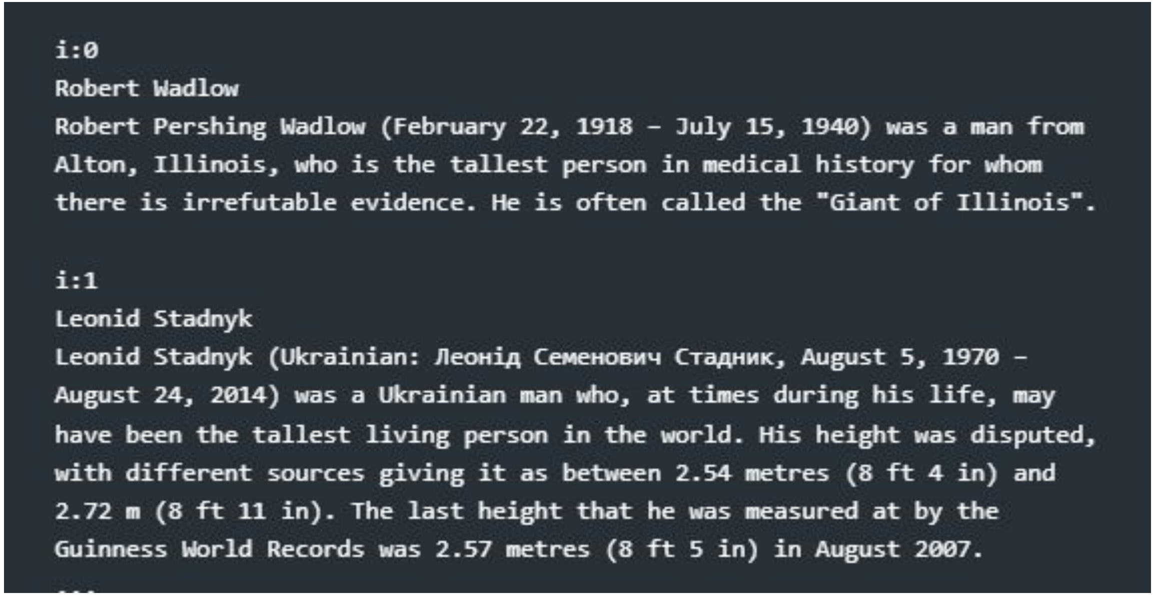
#3 - Keyword Search with 3 results:



```
Output:

i:0
Monarchy of Canada
The monarchy of Canada is at the very core of both Canada's federal
structure and Westminster-style of parliamentary and constitutional
democracy. The monarchy is the foundation of the executive (Queen-in-
Council), legislative (Queen's Majesty's Parliament for Canada), and
judicial (Queen's Courts for Canada) branches within both federal and
provincial jurisdictions. The sovereign is represented in Canada by the
Canadian Crown, embodied by the Canadian monarch personally (currently His
Majesty King Charles III) and the governor general (the appointed viceroy
who represents His Majesty in Canada), as well as the lieutenant governors
of the provinces (who represent His Majesty in each province).

i:1
Early modern period
The early modern period is a period in the history of science, spanning
roughly from the late 15th century to the late 18th century, in which a
significant departure from the medieval approach to science took place. It
may be more precisely defined as the period roughly from the Age of
Discovery to the rise of modern science.

i:2
Flag of Canada
The national flag of Canada, also known as the Canadian Red Ensign, the
Maple Leaf, or "l'Unifolié" (French for "the one-leafed"), is a red field
with a white square at its centre, in which two red borders become visible
with a stylized red maple leaf in its centre. It is from this maple leaf
that the flag is commonly referred to as the "Maple Leaf".
```

Output result

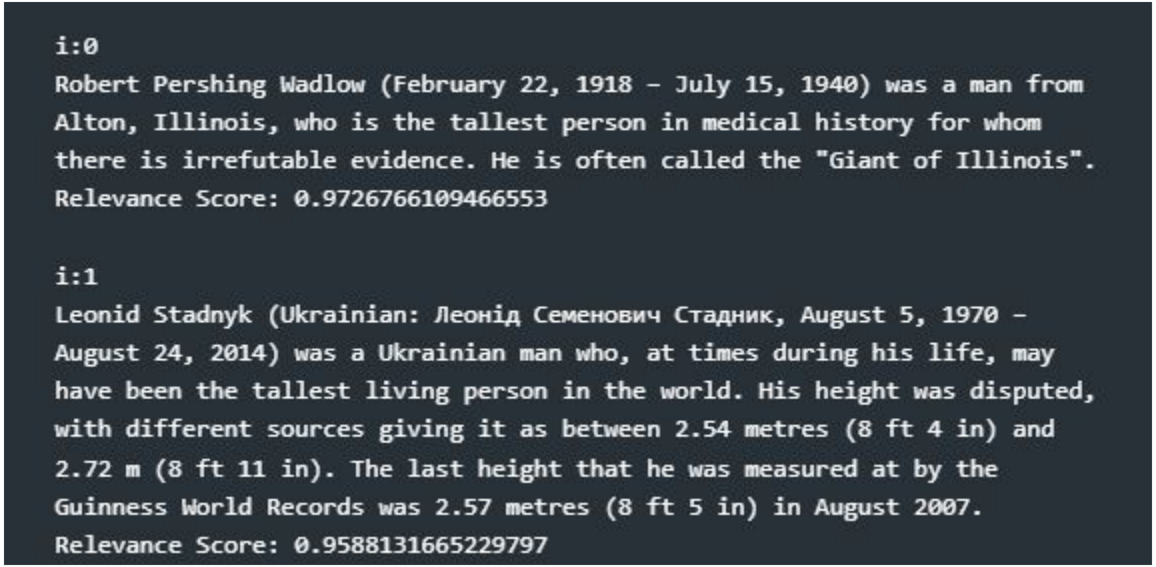
#4.1 Dense Retrieval of a new query



```
i:0
Robert Wadlow
Robert Pershing Wadlow (February 22, 1918 – July 15, 1940) was a man from
Alton, Illinois, who is the tallest person in medical history for whom
there is irrefutable evidence. He is often called the "Giant of Illinois".

i:1
Leonid Stadnyk
Leonid Stadnyk (Ukrainian: Леонід Семенович Стадник, August 5, 1970 –
August 24, 2014) was a Ukrainian man who, at times during his life, may
have been the tallest living person in the world. His height was disputed,
with different sources giving it as between 2.54 metres (8 ft 4 in) and
2.72 m (8 ft 11 in). The last height that he was measured at by the
Guinness World Records was 2.57 metres (8 ft 5 in) in August 2007.
```

Output result

#4.2 - ReRank the Dense Retrieval of a new query



```
i:0
Robert Pershing Wadlow (February 22, 1918 – July 15, 1940) was a man from
Alton, Illinois, who is the tallest person in medical history for whom
there is irrefutable evidence. He is often called the "Giant of Illinois".
Relevance Score: 0.9726766109466553

i:1
Leonid Stadnyk (Ukrainian: Леонід Семенович Стадник, August 5, 1970 –
August 24, 2014) was a Ukrainian man who, at times during his life, may
have been the tallest living person in the world. His height was disputed,
with different sources giving it as between 2.54 metres (8 ft 4 in) and
2.72 m (8 ft 11 in). The last height that he was measured at by the
Guinness World Records was 2.57 metres (8 ft 5 in) in August 2007.
Relevance Score: 0.9588131665229797
```

Reference

- Fine-Tuning based on 2000 drug examples from an Excel file

Original repo: https://github.com/MynameisKoi/CS589/tree/main/FineTuning

Source code:

- rerank.py
- utils.py