Lab 3 - Query Expansion

In [1]:
```
from helper_utils import load_chroma, word_wrap, project_embeddings
from chromadb.utils.embedding_functions import SentenceTransformerEmbeddingF
```

In [2]:
```
embedding_function = SentenceTransformerEmbeddingFunction()

chroma_collection = load_chroma(filename='microsoft_annual_report_2022.pdf',
chroma_collection.count()
```

README.md: 10.7k/10.7k [00:00<00:00, 100% 1.21MB/s]

config.json: 612/612 [00:00<00:00, 100% 75.6kB/s]

config_sentence_transformers.json: 116/116 [00:00<00:00, 100% 14.7kB/s]

data_config.json: 39.3k/39.3k [00:00<00:00, 100% 670kB/s]

pytorch_model.bin: 90.9M/90.9M [00:01<00:00, 100% 70.8MB/s]

In [3]:
```
import os
import openai
from openai import OpenAI

from dotenv import load_dotenv, find_dotenv
_ = load_dotenv(find_dotenv()) # read local .env file
openai.api_key = os.environ['OPENAI_API_KEY']

openai_client = OpenAI()
```

In [4]:
```
import umap

embeddings = chroma_collection.get(include=['embeddings'])['embeddings']
umap_transform = umap.UMAP(random_state=0, transform_seed=0).fit(embeddings)
projected_dataset_embeddings = project_embeddings(embeddings, umap_transform
```

```
/usr/local/lib/python3.9/site-packages/umap/umap_.py:1943: UserWarning: n_job
s value -1 overridden to 1 by setting random_state. Use no seed for paralleli
sm.
  warn(f"n_jobs value {self.n_jobs} overridden to 1 by setting random_state.
Use no seed for parallelism.")
100%|██████████| 349/349 [06:03<00:00,  1.04s/it]
```

Expansion with generated answers

https://arxiv.org/abs/2305.03653 (https://arxiv.org/abs/2305.03653)

In [5]:
```
earch assistant. Provide an example answer to the given question, that might
```

In [6]:
```
original_query = "Was there significant turnover in the executive team?"
hypothetical_answer = augment_query_generated(original_query)

joint_query = f"{original_query} {hypothetical_answer}"
print(word_wrap(joint_query))
```

```
huggingface/tokenizers: The current process just got forked, after parallelis
m has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(tru
e | false)

Was there significant turnover in the executive team? During the fiscal
year, there were no significant changes in the executive team. The core
members of the executive team remained consistent, providing continuity
in leadership and strategic decision-making processes.
```

In [7]:
```
results = chroma_collection.query(query_texts=joint_query, n_results=5, incl
retrieved_documents = results['documents'][0]

for doc in retrieved_documents:
    print(word_wrap(doc))
    print('')
```

unresolved with the irs, evaluating management ’ s estimates relating to their determination of uncertain tax positions required extensive audit effort and a high degree of auditor judgment, including involvement of our tax specialists. how the critical audit matter was addressed in the audit our principal audit procedures to evaluate management ’ s estimates of uncertain tax positions related to unresolved transfer pricing issues included the following : • we evaluated the appropriateness and consistency of management ’ s methods and assumptions used in the identification, recognition, measurement, and disclosure of uncertain tax positions, which included testing the effectiveness of the related internal controls. • we read and evaluated management ’ s documentation, including relevant accounting policies and information obtained by management from outside tax specialists, that detailed the basis of the uncertain tax positions.

to be their authentic selves and do their best work every day. we support multiple highly active employee resource groups for women, families, racial and ethnic minorities, military, people with disabilities, and employees who identify as lgbtqia +, where employees can go for support, networking, and community - building. as described in our 2021 proxy statement, annual performance and compensation reviews of our senior leadership team include an evaluation of their contributions to employee culture and diversity. to ensure accountability over time, we publicly disclose our progress on a multitude of workforce metrics including : • detailed breakdowns of gender, racial, and ethnic minority representation in our employee population, with data by job types, levels, and segments of our business. • our eeo - 1 report (equal employment opportunity). • disability representation. • pay equity (see details below).

86 income taxes — uncertain tax positions — refer to note 12 to the financial statements critical audit matter description the company ’ s long - term income taxes liability includes uncertain tax positions related to transfer pricing issues that remain unresolved with the internal revenue service (“ irs ”). the company remains under irs audit, or subject to irs audit, for tax years subsequent to 2003. while the company has settled a portion of the irs audits, resolution of the remaining matters could have a material impact on the company ’ s financial statements. conclusions on recognizing and measuring uncertain tax positions involve significant estimates and management judgment and include complex considerations of the internal revenue code, related regulations, tax case laws, and prior - year audit settlements. given the complexity and the subjective nature of the transfer pricing issues that remain

statements would be prevented or detected. management conducted an evaluation of the effectiveness of our internal control over financial reporting based on the framework in internal control – integrated framework (2013) issued by the committee of sponsoring organizations of the treadway commission. based on this evaluation, management concluded that the company ’ s internal control over financial reporting was effective as of june 30, 2022. there were no changes in our internal control over financial reporting during the quarter ended june 30, 2022 that have materially affected, or are reasonably likely to materially affect, our internal control over financial reporting. deloitte & touche llp has audited our internal control over financial reporting as of june 30, 2022 ; their report follows.

```
88 report of independent registered public accounting firm to the
stockholders and the board of directors of microsoft corporation
opinion on internal control over financial reporting we have audited
the internal control over financial reporting of microsoft corporation
and subsidiaries ( the “ company ” ) as of june 30, 2022, based on
criteria established in internal control – integrated framework ( 2013
) issued by the committee of sponsoring organizations of the treadway
commission ( coso ). in our opinion, the company maintained, in all
material respects, effective internal control over financial reporting
as of june 30, 2022, based on criteria establis hed in internal control
– integrated framework ( 2013 ) issued by coso. we have also audited,
in accordance with the standards of the public company accounting
oversight board ( united states ) ( pcaob ), the consolidated financial
statements as of and for the year ended june 30, 2022, of the company
and
```

In [8]:
```
retrieved_embeddings = results['embeddings'][0]
original_query_embedding = embedding_function([original_query])
augmented_query_embedding = embedding_function([joint_query])

projected_original_query_embedding = project_embeddings(original_query_embec
projected_augmented_query_embedding = project_embeddings(augmented_query_emb
projected_retrieved_embeddings = project_embeddings(retrieved_embeddings, um
```

```
100%|██████████| 1/1 [00:01<00:00,  1.35s/it]
100%|██████████| 1/1 [00:00<00:00,  1.07it/s]
100%|██████████| 5/5 [00:04<00:00,  1.01it/s]
```

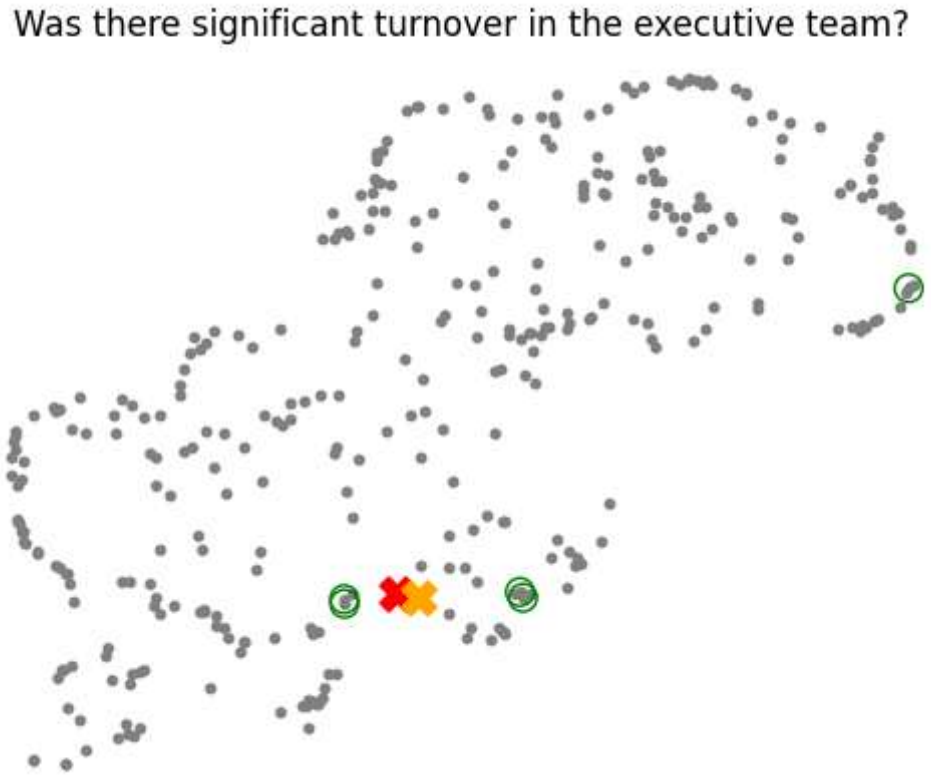
In [9]:
```
import matplotlib.pyplot as plt

# Plot the projected query and retrieved documents in the embedding space
plt.figure()
plt.scatter(projected_dataset_embeddings[:, 0], projected_dataset_embeddings
plt.scatter(projected_retrieved_embeddings[:, 0], projected_retrieved_embedc
plt.scatter(projected_original_query_embedding[:, 0], projected_original_que
plt.scatter(projected_augmented_query_embedding[:, 0], projected_augmented_c

plt.gca().set_aspect('equal', 'datalim')
plt.title(f'{original_query}')
plt.axis('off')
```

```
huggingface/tokenizers: The current process just got forked, after parallelis
m has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(tru
e | false)
huggingface/tokenizers: The current process just got forked, after parallelis
m has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(tru
e | false)

(-1.289832666516304, 8.499054208397865, 1.750054621696472, 9.173876023292541)
```



Expansion with multiple queries

In [10]:
```
def augment_multiple_query(query, model="gpt-3.5-turbo"):
    messages = [
        {
            "role": "system",
            "content": "You are a helpful expert financial research assistan
            "Suggest up to five additional related questions to help them fi
            "Suggest only short questions without compound sentences. Sugges
            "Make sure they are complete questions, and that they are relate
            "Output one question per line. Do not number the questions."
        },
        {"role": "user", "content": query}
    ]

    response = openai_client.chat.completions.create(
        model=model,
        messages=messages,
    )
    content = response.choices[0].message.content
    content = content.split("\n")
    return content
```

In [11]:
```
original_query = "What were the most important factors that contributed to i
augmented_queries = augment_multiple_query(original_query)

for query in augmented_queries:
    print(query)
```

```
- What were the main sources of revenue for the company?
- How did changes in pricing affect revenue growth?
- Were there any new product launches that significantly impacted revenue?
- Did the company enter into any strategic partnerships that boosted revenue?
- Were there any changes in market demand that influenced revenue growth?
```

In [12]:
```
queries = [original_query] + augmented_queries
results = chroma_collection.query(query_texts=queries, n_results=5, include=

retrieved_documents = results['documents']

# Deduplicate the retrieved documents
unique_documents = set()
for documents in retrieved_documents:
    for document in documents:
        unique_documents.add(document)

for i, documents in enumerate(retrieved_documents):
    print(f"Query: {queries[i]}")
    print('')
    print("Results:")
    for doc in documents:
        print(word_wrap(doc))
        print('')
    print('-'*100)
```

```
productivity and business processes revenue increased $ 9. 4 billion or
18 %. • office commercial products and cloud services revenue increased
$ 4. 4 billion or 13 %. office 365 commercial revenue grew 18 % driven
by seat growth of 14 %, with continued momentum in small and medium
business and frontline worker offerings, as well as growth in revenue
per user. office commercial products revenue declined 22 % driven by
continued customer shift to cloud offerings. • office consumer products
and cloud services revenue increased $ 641 million or 11 % driven by
microsoft 365 consumer subscription revenue. microsoft 365 consumer
subscribers grew 15 % to 59. 7 million. • linkedin revenue increased $
3. 5 billion or 34 % driven by a strong job market in our talent
solutions business and advertising demand in our marketing solutions
business.

----------------------------------------------------------------------
-------------------------
Query: - Were there any new product launches that significantly impacted r
evenue?

Results:
```

In [13]:
```
original_query_embedding = embedding_function([original_query])
augmented_query_embeddings = embedding_function(augmented_queries)

project_original_query = project_embeddings(original_query_embedding, umap_t
project_augmented_queries = project_embeddings(augmented_query_embeddings, u
```

```
100%|██████████| 1/1 [00:01<00:00,  1.04s/it]
100%|██████████| 5/5 [00:04<00:00,  1.06it/s]
```

In [14]:
```
result_embeddings = results['embeddings']
result_embeddings = [item for sublist in result_embeddings for item in subli
projected_result_embeddings = project_embeddings(result_embeddings, umap_tra
```

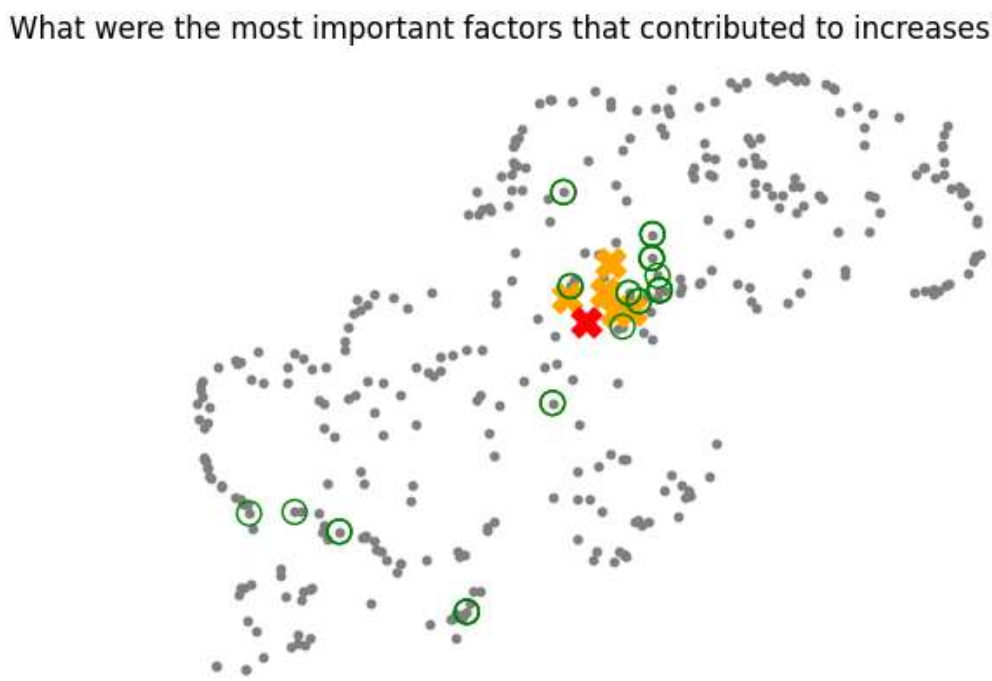
```
100%|██████████| 30/30 [00:32<00:00,  1.07s/it]
```

In [15]:
```
import matplotlib.pyplot as plt

plt.figure()
plt.scatter(projected_dataset_embeddings[:, 0], projected_dataset_embeddings
plt.scatter(project_augmented_queries[:, 0], project_augmented_queries[:, 1]
plt.scatter(projected_result_embeddings[:, 0], projected_result_embeddings[:
plt.scatter(project_original_query[:, 0], project_original_query[:, 1], s=15

plt.gca().set_aspect('equal', 'datalim')
plt.title(f'{original_query}')
plt.axis('off')
```

```
(-1.289832666516304, 8.499054208397865, 1.750054621696472, 9.173876023292541)
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: