In [1]:

```
# Initialize Otter
import otter
grader = otter.Notebook("hw6.ipynb")
```

CPSC 330 - Applied Machine Learning

Homework 6: Clustering

Associated lectures: Lectures 15 and 16

Due date: Check the Calendar

Imports

In [2]:

```
import os
from hashlib import sha1

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

%matplotlib inline
pd.set_option("display.max_colwidth", 0)
```

Submission instructions

rubric={points:6}

Please be aware that this homework assignment requires installation of several packages in your course environment. It's possible that you'll encounter installation challenges, which might be frustrating. However, remember that solving these issues is not wasting time but it is an essential skill for anyone aspiring to work in data science or machine learning.

Follow the homework submission instructions.

You may work in a group on this homework and submit your assignment as a group. Below are some instructions on working as a group.

- The maximum group size is 4.
- Use group work as an opportunity to collaborate and learn new things from each other.
- Be respectful to each other and make sure you understand all the concepts in the assignment well.
- It's your responsibility to make sure that the assignment is submitted by one of the group members before the deadline.
- You can find the instructions on how to do group submission on Gradescope here.

When you are ready to submit your assignment do the following:

1. Run all cells in your notebook to make sure there are no errors by doing `Kernel -> Restart Kernel and Clear All Outputs` and then `Run -> Run All Cells`.
2. Notebooks with cell execution numbers out of order or not starting from "1" will have marks deducted. Notebooks without the output displayed may not be graded at all (because we need to see the output in order to grade your work).
3. Upload the assignment using Gradescope's drag and drop tool. Check out this Gradescope Student Guide if you need help with Gradescope submission.
4. Make sure that the plots and output are rendered properly in your submitted file.
5. If the .ipynb file is too big and doesn't render on Gradescope, also upload a pdf or html in addition to the .ipynb. If the pdf or html also fail to render on Gradescope, please create two files for your homework: hw6a.ipynb with Exercise 1 and hw6b.ipynb with Exercises 2 and 3 and submit these two files in your submission.

Points: 6

Exercise 1: Document clustering warm-up

In this homework, we will explore a popular application of clustering called **document clustering**. A large amount of unlabeled text data is available out there (e.g., news, recipes, online Q&A, tweets), and clustering is a commonly used technique to organize this data in a meaningful way.

As a warm up, in this exercise you will cluster sentences from a toy corpus. Later in the homework you will work with a real corpus.

The code below extracts introductory sentences of Wikipedia articles on a set of queries. To run the code successfully, you will need the `wikipedia` package installed in the course environment.

```
conda activate cpsc330
conda install -c conda-forge wikipedia
```

Your tasks:

Run the code below which

- extracts content of Wikipedia articles on a set of queries
- tokenizes the text (i.e., separates sentences) and
- stores the 2nd sentence in each article as a document representing that article

> Feel free to experiment with Wikipedia queries of your choice. But stick to the provided list for the final submission so that it's easier for the TAs to grade your submission.

> For tokenization we are using the `nltk` package. If you do not have this package in the course environment, you will have to install it.

```
conda activate cpsc330
conda install -c anaconda nltk
```

Even if you have the package installed via the course `conda` environment, you might have to download `nltk` pre-trained models, which can be done with the code below.

In [3]:
```
import nltk

nltk.download("punkt")
nltk.download('punkt_tab')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/zhouzhiying_izzy/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to
[nltk_data]     /Users/zhouzhiying_izzy/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
```

Out[3]:
```
True
```

In [4]:
```
import wikipedia
from nltk.tokenize import sent_tokenize, word_tokenize

queries = [
    "Artificial Intelligence", "Deep learning", "Unsupervised learning", "Qu
    "Environmental protection", "Climate Change", "Renewable Energy", "Biodi
    "French Cuisine", "Bread food", "Dumpling food"
]


wiki_dict = {"wiki query": [], "text": [], "n_words": []}
for i in range(len(queries)):
    text = sent_tokenize(wikipedia.page(queries[i]).content)[1]
    wiki_dict["text"].append(text)
    wiki_dict["n_words"].append(len(word_tokenize(text)))
    wiki_dict["wiki query"].append(queries[i])

wiki_df = pd.DataFrame(wiki_dict)
wiki_df
```

Out[4]:

	wiki query	text	n_words
0	Artificial Intelligence	It is a field of research in computer science that develops and studies methods and software that enable machines to perceive their environment and use learning and intelligence to take actions that maximize their chances of achieving defined goals.	40
1	Deep learning	The field takes inspiration from biological neuroscience and is centered around stacking artificial neurons into layers and "training" them to process data.	25
2	Unsupervised learning	The training data is processed, building a function that maps new data to expected output values.	18
3	Quantum Computing	On small scales, physical matter exhibits properties of both particles and waves, and quantum computing leverages this behavior using specialized hardware.	24
4	Environmental protection	Its objectives are to conserve natural resources and the existing natural environment and, where it is possible, to repair damage and reverse trends.	26
5	Climate Change	Climate change in a broader sense also includes previous long-term changes to Earth's climate.	16
6	Renewable Energy	The most widely used renewable energy types are solar energy, wind power, and hydropower.	17
7	Biodiversity	It can be measured on various levels.	8
8	French Cuisine	In the 14th century, Guillaume Tirel, a court chef known as "Taillevent", wrote Le Viandier, one of the earliest recipe collections of medieval France.	31
9	Bread food	Throughout recorded history and around the world, it has been an important part of many cultures' diet.	20
10	Dumpling food	The dough can be based on bread, wheat or other flours, or potatoes, and it may be filled with meat, fish, tofu, cheese, vegetables, or a combination.	36

Our toy corpus has six toy documents (`text` column in the dataframe) extracted from Wikipedia queries.

1.1 How many clusters?

rubric={points}

Your tasks:

1. If you are asked to cluster the documents from this toy corpus manually, how many clusters would you identify and how would you label each cluster?

Solution_1.1

Points: 1

3 clusters. "Artificial Intelligence", "Deep learning", "Unsupervised learning", "Quantum Computing" are clustered in cs terms category; "Environmental protection", "Climate Change", "Renewable Energy", "Biodiversity" are clustered in environment category; "French Cuisine", "Bread food", "Dumpling food" should be clustered in food category.

1.2 `KMeans` with bag-of-words representation

rubric={points}

In the lecture, we saw that data representation plays a crucial role in clustering. Changing flattened representation of images to feature vectors extracted from pre-trained models greatly improved the quality of clustering.

What kind of representation is suitable for text data? We have used bag-of-words representation to numerically encode text data before, where each document is represented with a vector of word frequencies.

Let's try clustering documents with this simplistic representation.

Your tasks:

1. Create bag-of-words representation using `CountVectorizer` with default arguments for the `text` column in `wiki_df` above.
2. Cluster the encoded documents with `KMeans` clustering. Use `random_state=42` (for reproducibility) and set `n_clusters` to the number you identified in the

previous exercise.

3. Store the clustering labels in `kmeans_bow_labels` variable below.

Solution_1.2

Points: 4

In [5]:
```
from sklearn.feature_extraction.text import CountVectorizer

vec = CountVectorizer()
X_counts = vec.fit_transform(wiki_df["text"])
bow_df = pd.DataFrame(
    X_counts.toarray(), columns=vec.get_feature_names_out(), index=wiki_df["
)
bow_df
```

Out[5]:

	14th	achieving	actions	also	an	and	are	around	artificial	as	...
text											
It is a field of research in computer science that develops and studies methods and software that enable machines to perceive their environment and use learning and intelligence to take actions that maximize their chances of achieving defined goals.	0	1	1	0	0	4	0	0	0	0	...
The field takes inspiration from biological neuroscience and is centered around stacking artificial neurons into layers and "training" them to process data.	0	0	0	0	0	2	0	1	1	0	...
The training data is processed, building a function that maps new data to expected output values.	0	0	0	0	0	0	0	0	0	0	...
On small scales,	0	0	0	0	0	2	0	0	0	0	...

	14th	achieving	actions	also	an	and	are	around	artificial	as	...
text											
physical matter exhibits properties of both particles and waves, and quantum computing leverages this behavior using specialized hardware.											
Its objectives are to conserve natural resources and the existing natural environment and, where it is possible, to repair damage and reverse trends.	0	0	0	0	0	3	1	0	0	0	...
Climate change in a broader sense also includes previous long-term changes to Earth's climate.	0	0	0	1	0	0	0	0	0	0	...
The most widely used renewable energy types are solar energy, wind power, and hydropower.	0	0	0	0	0	1	1	0	0	0	...
It can be measured on various levels.	0	0	0	0	0	0	0	0	0	0	...

	14th	achieving	actions	also	an	and	are	around	artificial	as	...
text											
In the 14th century, Guillaume Tirel, a court chef known as "Taillevent", wrote Le Viandier, one of the earliest recipe collections of medieval France.	1	0	0	0	0	0	0	0	0	1	...
Throughout recorded history and around the world, it has been an important part of many cultures' diet.	0	0	0	0	1	1	0	1	0	0	...
The dough can be based on bread, wheat or other flours, or potatoes, and it may be filled with meat, fish, tofu, cheese, vegetables, or a combination.	0	0	0	0	0	1	0	0	0	0	...

11 rows × 159 columns

In [6]:
```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(X_counts)
kmeans_bow_labels = kmeans.labels_
kmeans_bow_labels
```

Out[6]:
```
array([2, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1], dtype=int32)
```

In [7]:
```
wiki_df["bow_kmeans"] = kmeans_bow_labels
wiki_df
```

Out[7]:

	wiki query	text	n_words	bow_kmeans
0	Artificial Intelligence	It is a field of research in computer science that develops and studies methods and software that enable machines to perceive their environment and use learning and intelligence to take actions that maximize their chances of achieving defined goals.	40	2
1	Deep learning	The field takes inspiration from biological neuroscience and is centered around stacking artificial neurons into layers and "training" them to process data.	25	1
2	Unsupervised learning	The training data is processed, building a function that maps new data to expected output values.	18	1
3	Quantum Computing	On small scales, physical matter exhibits properties of both particles and waves, and quantum computing leverages this behavior using specialized hardware.	24	1
4	Environmental protection	Its objectives are to conserve natural resources and the existing natural environment and, where it is possible, to repair damage and reverse trends.	26	0
5	Climate Change	Climate change in a broader sense also includes previous long-term changes to Earth's climate.	16	1
6	Renewable Energy	The most widely used renewable energy types are solar energy, wind power, and hydropower.	17	1
7	Biodiversity	It can be measured on various levels.	8	1
8	French Cuisine	In the 14th century, Guillaume Tirel, a court chef known as "Taillevent", wrote Le Viandier, one of the earliest recipe collections of medieval France.	31	1
9	Bread food	Throughout recorded history and around the world, it has been an important part of many cultures' diet.	20	1
10	Dumpling food	The dough can be based on bread, wheat or other flours, or potatoes, and it may be filled with meat, fish, tofu, cheese, vegetables, or a combination.	36	1

1.3 Sentence embedding representation

rubric={points}

Bag-of-words representation is limited in that it does not take into account word ordering and context. There are other richer and more expressive representations of text which can be extracted using transfer learning. In this lab, we will use one such representation called sentence embedding representation, which uses deep learning models to generate dense, fixed-length vector representations for sentences. We will extract such representations using sentence transformer package. Sentence embedding takes into account context of words and semantic meaning of sentences and it is likely to work better when we are interested in clustering sentences based on their semantic similarity.

```
conda activate cpsc330
conda install pytorch::pytorch torchvision torchaudio -c
pytorch
conda install -c conda-forge sentence-transformers
```

Your tasks:

1. Run the code below to create sentence embedding representation of documents in our toy corpus.
2. Cluster documents in our toy corpus encoded with this representation (`emb_sents`) and `KMeans` with following arguments:
    - `random_state=42` (for reproducibility)
    - `n_clusters`=the number of clusters you identified in 1.1
3. Store the clustering labels in `kmeans_emb_labels` variable below.

In [8]:

```
#pip install transformers -U
```

In [9]:

```
from sentence_transformers import SentenceTransformer

embedder = SentenceTransformer("paraphrase-distilroberta-base-v1")

# If this cell gives an error, try updating transformers with
# pip install transformers -U
```

```
/opt/anaconda3/envs/cpsc330/lib/python3.12/site-packages/sentence_transforme
rs/cross_encoder/CrossEncoder.py:13: TqdmExperimentalWarning: Using `tqdm.au
tonotebook.tqdm` in notebook mode. Use `tqdm.tqdm` instead to force console
mode (e.g. in jupyter console)
  from tqdm.autonotebook import tqdm, trange
```

In [10]:

```
emb_sents = embedder.encode(wiki_df["text"])
emb_sent_df = pd.DataFrame(emb_sents, index=wiki_df.index)
emb_sent_df
```

Out[10]:

	0	1	2	3	4	5	6
0	0.102874	0.201959	0.044092	0.281749	0.321483	-0.281129	0.042515
1	0.000322	0.428834	0.152298	-0.161278	0.224354	-0.363829	0.110951
2	0.236464	-0.282463	-0.258300	0.300584	0.234606	0.061746	-0.072744
3	0.276844	0.657946	0.106465	0.290567	0.803929	0.023764	0.136675
4	0.200327	0.157551	0.093484	0.120533	-0.439307	0.148568	-0.003543
5	0.189105	0.406864	0.172560	0.273777	0.058933	0.224641	-0.056590
6	-0.066224	0.465511	-0.135840	-0.229255	-0.144745	0.013772	-0.122810
7	-0.139883	0.207129	-0.127447	0.214821	-0.099096	0.063319	-0.347633
8	-0.112771	-0.259073	0.172584	-0.149188	-0.074585	0.222288	-0.213039
9	-0.022418	0.217159	0.022694	0.003616	0.240856	0.358047	-0.053310
10	-0.123724	0.190113	-0.064433	0.206614	0.198812	0.156827	0.040764

11 rows × 768 columns

Solution_1.3

Points: 3

In [11]:
```
kmeans_emb = KMeans(n_clusters=3, random_state=42)
kmeans_emb.fit(emb_sents)
```

Out[11]:

▾ KMeans

```
KMeans(n_clusters=3, random_state=42)
```

In [12]:
```
kmeans_emb_labels = kmeans_emb.labels_
kmeans_emb_labels
```

Out[12]:
```
array([2, 2, 2, 2, 0, 0, 0, 1, 1, 1, 1], dtype=int32)
```

In [13]:
```
wiki_df["emb_kmeans"] = kmeans_emb_labels
wiki_df
```

Out[13]:

	wiki query	text	n_words	bow_kmeans	emb_kmeans
0	Artificial Intelligence	It is a field of research in computer science that develops and studies methods and software that enable machines to perceive their environment and use learning and intelligence to take actions that maximize their chances of achieving defined goals.	40	2	2
1	Deep learning	The field takes inspiration from biological neuroscience and is centered around stacking artificial neurons into layers and "training" them to process data.	25	1	2
2	Unsupervised learning	The training data is processed, building a function that maps new data to expected output values.	18	1	2
3	Quantum Computing	On small scales, physical matter exhibits properties of both particles and waves, and quantum computing leverages this behavior using specialized hardware.	24	1	2
4	Environmental protection	Its objectives are to conserve natural resources and the existing natural environment and, where it is possible, to repair damage and reverse trends.	26	0	0
5	Climate Change	Climate change in a broader sense also includes previous long-term changes to Earth's climate.	16	1	0
6	Renewable Energy	The most widely used renewable energy types are solar energy, wind power, and hydropower.	17	1	0
7	Biodiversity	It can be measured on various levels.	8	1	1
8	French Cuisine	In the 14th century, Guillaume Tirel, a court chef known as "Taillevent",	31	1	1

	wiki query	text	n_words	bow_kmeans	emb_kmeans
		wrote Le Viandier, one of the earliest recipe collections of medieval France.			
9	Bread food	Throughout recorded history and around the world, it has been an important part of many cultures' diet.	20	1	1
10	Dumpling food	The dough can be based on bread, wheat or other flours, or potatoes, and it may be filled with meat, fish, tofu, cheese, vegetables, or a combination.	36	1	1

1.4 DBSCAN with cosine distance

rubric={points}

Now try `DBSCAN` on our toy dataset. K-Means is kind of bound to the Euclidean distance because it is based on the notion of means. With `DBSCAN` we can try different distance metrics. In the context of text data, cosine similarities or cosine distances tend to work well. Given vectors u and v, the **cosine distance** between the vectors is defined as:

$$distance_{cosine}(u, v) = 1 - (\frac{u \cdot v}{\|u\|_2 \|v\|_2})$$

Your tasks

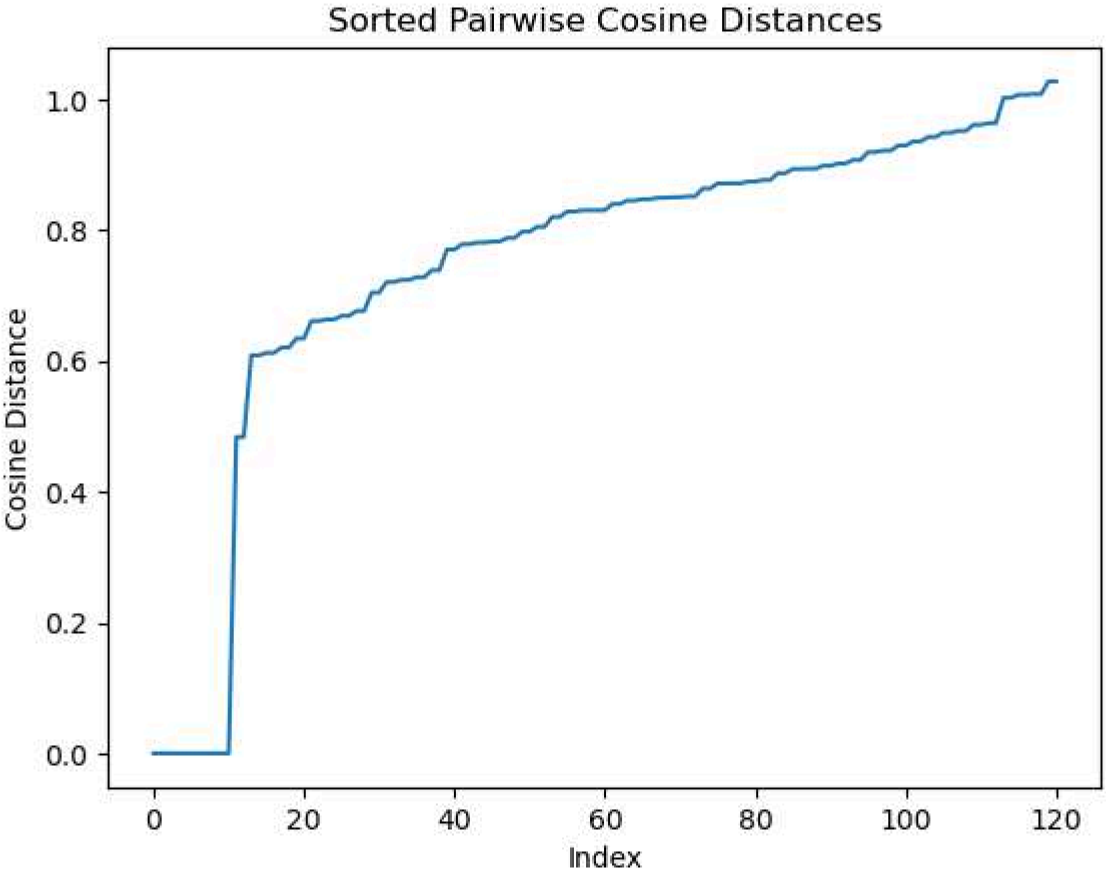
1. Cluster documents in our toy corpus encoded with sentence embedding representation (`emb_sents`) and DBSCAN with `metric='cosine'` . You will have to set appropriate values for the hyperparamters `eps` and `min_samples` to get meaningful clusters, as default values of these hyperparameters are unlikely to work well on this toy dataset.
2. Store the clustering labels in the `dbscan_emb_labels` variable below.

Solution_1.4

Points: 4

In [14]:

```
from sklearn.metrics.pairwise import cosine_distances

cosine_dists = cosine_distances(emb_sents)

# Flatten the distance matrix and sort distances to find a reasonable range
flattened_distances = cosine_dists.flatten()
sorted_distances = np.sort(flattened_distances)

import matplotlib.pyplot as plt

plt.plot(sorted_distances)
plt.xlabel("Index")
plt.ylabel("Cosine Distance")
plt.title("Sorted Pairwise Cosine Distances")
plt.show()
```



Based on the plot, the best eps is around 0.68.

We want min_sample to be 3 since we think there're at least 3 points in each of the 3 clusters we identified earlier.

In [15]:

```
from sklearn.cluster import DBSCAN
dbscan = DBSCAN(eps=0.68, min_samples=3, metric='cosine')
dbscan.fit(emb_sents)
```

Out[15]:

```
▾ DBSCAN
DBSCAN(eps=0.68, metric='cosine', min_samples=3)
```

In [16]:

```
dbscan_emb_labels = dbscan.labels_
dbscan_emb_labels
```

Out[16]:

```
array([ 0,  0,  0,  0,  0,  0,  0,  1, -1,  1,  1])
```

In [17]:

```
wiki_df["emb_dbscan"] = dbscan_emb_labels
wiki_df
```

Out[17]:

	wiki query	text	n_words	bow_kmeans	emb_kmeans	emb_dbscan
0	Artificial Intelligence	It is a field of research in computer science that develops and studies methods and software that enable machines to perceive their environment and use learning and intelligence to take actions that maximize their chances of achieving defined goals.	40	2	2	0
1	Deep learning	The field takes inspiration from biological neuroscience and is centered around stacking artificial neurons into layers and "training" them to process data.	25	1	2	0
2	Unsupervised learning	The training data is processed, building a function that maps new data to expected output values.	18	1	2	0
3	Quantum Computing	On small scales, physical matter	24	1	2	0

	wiki query	text	n_words	bow_kmeans	emb_kmeans	emb_dbscan
		exhibits properties of both particles and waves, and quantum computing leverages this behavior using specialized hardware.				
4	Environmental protection	Its objectives are to conserve natural resources and the existing natural environment and, where it is possible, to repair damage and reverse trends.	26	0	0	0
5	Climate Change	Climate change in a broader sense also includes previous long-term changes to Earth's climate.	16	1	0	0
6	Renewable Energy	The most widely used renewable energy types are solar energy, wind power, and hydropower.	17	1	0	0
7	Biodiversity	It can be measured on various levels.	8	1	1	1
8	French Cuisine	In the 14th century, Guillaume Tirel, a court chef known	31	1	1	-1

	wiki query	text	n_words	bow_kmeans	emb_kmeans	emb_dbscan
		as "Taillevent", wrote Le Viandier, one of the earliest recipe collections of medieval France.				
9	Bread food	Throughout recorded history and around the world, it has been an important part of many cultures' diet.	20	1	1	1
10	Dumpling food	The dough can be based on bread, wheat or other flours, or potatoes, and it may be filled with meat, fish, tofu, cheese, vegetables, or a combination.	36	1	1	1

1.5 Hierarchical clustering with sentence embedding representation

rubric={points}

Your tasks:

Try hierarchical clustering on `emb_sents`. In particular

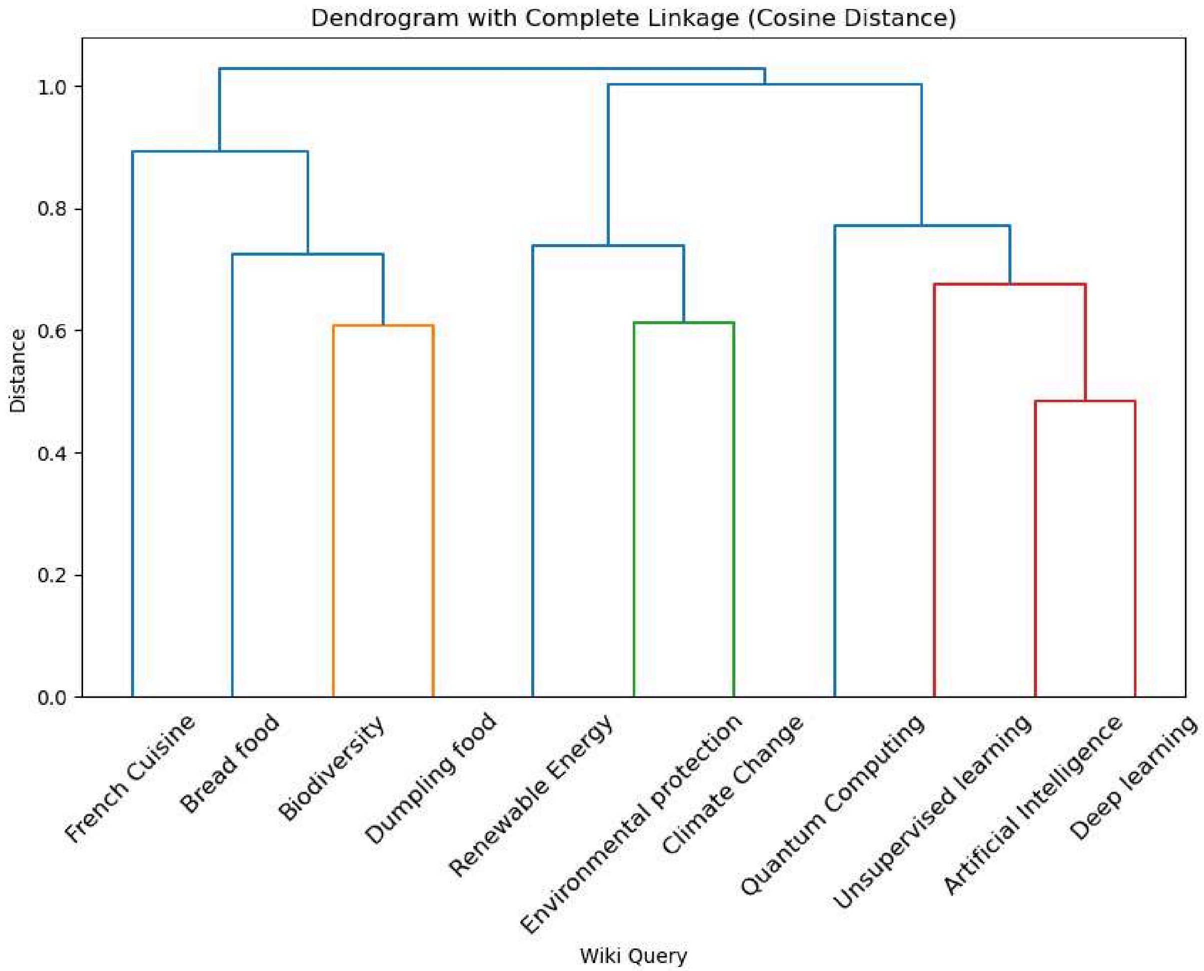
1. Create and show a dendrogram with `complete` linkage and `metric='cosine'` on this toy dataset.
2. Create flat clusters using `fcluster` with appropriate hyperparameters and store cluster labels to `hier_emb_labels` variable below.

Solution_1.5

Points: 3

In [18]:

```
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster
linkage_matrix = linkage(emb_sents, method='complete', metric='cosine')
plt.figure(figsize=(10, 6))
dendrogram(linkage_matrix, labels=wiki_df['wiki query'].values, leaf_rotatio
plt.title('Dendrogram with Complete Linkage (Cosine Distance)')
plt.xlabel('Wiki Query')
plt.ylabel('Distance')
plt.show()
```



In [19]:

```
hier_emb_labels = fcluster(linkage_matrix, 3, criterion='maxclust')
hier_emb_labels
```

Out[19]:

```
array([3, 3, 3, 3, 2, 2, 2, 1, 1, 1, 1], dtype=int32)
```

In [20]:

```
# hier_emb_labels = fcluster(Z, 3, criterion="maxclust") # alternative solut
```

In [21]:

```
wiki_df["emb_hierarchical"] = hier_emb_labels
wiki_df
```

Out[21]:

	wiki query	text	n_words	bow_kmeans	emb_kmeans	emb_dbscan	e
0	Artificial Intelligence	It is a field of research in computer science that develops and studies methods and software that enable machines to perceive their environment and use learning and intelligence to take actions that maximize their chances of achieving defined goals.	40	2	2	0	
1	Deep learning	The field takes inspiration from biological neuroscience and is centered around stacking artificial neurons into layers and "training" them to process data.	25	1	2	0	
2	Unsupervised learning	The training data is processed, building a function that maps new data to expected output values.	18	1	2	0	
3	Quantum Computing	On small scales, physical matter	24	1	2	0	

	wiki query	text	n_words	bow_kmeans	emb_kmeans	emb_dbscan	e
		exhibits properties of both particles and waves, and quantum computing leverages this behavior using specialized hardware.					
4	Environmental protection	Its objectives are to conserve natural resources and the existing natural environment and, where it is possible, to repair damage and reverse trends.	26	0	0	0	
5	Climate Change	Climate change in a broader sense also includes previous long-term changes to Earth's climate.	16	1	0	0	
6	Renewable Energy	The most widely used renewable energy types are solar energy, wind power, and hydropower.	17	1	0	0	
7	Biodiversity	It can be measured on various levels.	8	1	1	1	
8	French Cuisine	In the 14th century, Guillaume Tirel, a court chef known	31	1	1	-1	

	wiki query	text	n_words	bow_kmeans	emb_kmeans	emb_dbscan	e
		as "Taillevent", wrote Le Viandier, one of the earliest recipe collections of medieval France.					
9	Bread food	Throughout recorded history and around the world, it has been an important part of many cultures' diet.	20	1	1	1	
10	Dumpling food	The dough can be based on bread, wheat or other flours, or potatoes, and it may be filled with meat, fish, tofu, cheese, vegetables, or a combination.	36	1	1	1	

1.6 Discussion

rubric={points}

Your tasks:

1. Reflect on and discuss the clustering results of the methods you explored in the previous exercises, focusing on the following points:
    - effect of input representation on clustering results
    - whether the clustering results match with your intuitions and the challenges associated with getting the desired clustering results with each method

Solution_1.6

Points: 4

Input Representation Effect:

- Bag-of-Words (BoW) representation performed poorly.
- Embedding-based methods clearly outperformed BoW representation. This is evident in how embeddings consistently grouped related topics together - AI and computing topics were clustered together in emb_kmeans (cluster 2) and hierarchical clustering (label 3), while environmental topics (rows 4-6) were clustered in another distinct group.
- The embedding-based methods' ability to capture semantic relationships resulted in more intuitive clustering compared to the simpler BoW approach.

Clustering Results and Challenges: The clustering results generally aligned with expected topic groupings, but each method showed distinct strengths and weaknesses.

- BoW performs poorly on capturing semantic relationships.
- Hierarchical clustering had the most interpretable results with clear separation between technology (label 3), environmental (label 2), and food-related topics (label 1).
- DBSCAN struggled significantly, marking most documents as noise (0) or outliers (-1), highlighting the difficulty of parameter tuning.
- K-means performed reasonably well but was limited by the need to pre-specify cluster numbers. The results show how semantically clear documents can be challenging to cluster automatically, especially when they span multiple topics like "French Cuisine" which combines food and cultural elements.

1.7 Visualizing clusters

rubric={points:4}

One approach to working with unlabeled data is visualization. That said, our data is high-dimensional, making it challenging to visualize. Take sentence embedding representation as an example: each instance is depicted in 768 dimensions. To visualize such high-dimensional data, we can employ dimensionality reduction techniques to extract the most significant 2 or 3 components, and then visualize this low-dimensional data.

Given data as a `numpy` array and corresponding cluster assignments, the `plot_umap_clusters` function below transforms the data by applying dimensionality reduction technique called UMAP to it and plots the transformed data with different colours for different clusters.

> *Note: At this point we are using this function only for visualization and you are not expected to understand the UMAP part.*

You'll have to install the `umap-learn` package in the course conda environment either with `conda` or `pip`, as described in the documentation.

```
> conda activate cpsc330
> conda install -c conda-forge umap-learn
```

or

```
> conda activate cpsc330
> pip install umap-learn
```

If you get an error with the import below try

```
pip install --upgrade numba umap-learn
```

Your tasks:

1. Visualize the clusters created by the methods above using `plot_umap_clusters` function below. In other words, visualize clusters identified by each of the methods below.
    - K-Means with bag-of-words representation
    - K-Means with sentence embedding representation
    - DBSCAN with sentence embedding representation
    - Flat cluster of hierarchical clustering with sentence embedding representation

In [22]:

```python
import umap
```

In [23]:

```python
def plot_umap_clusters(
    data,
    cluster_labels,
    raw_sents=wiki_df["text"],
    show_labels=False,
    size=50,
    n_neighbors=15,
    title="UMAP visualization",
    ignore_noise=False,
):
    """
    Carry out dimensionality reduction using UMAP and plot 2-dimensional clu
    Parameters
    -----------
    data : numpy array
        data as a numpy array
    cluster_labels : list
        cluster labels for each row in the dataset
    raw_sents : list
        the original raw sentences for labeling datapoints
    show_labels : boolean
        whether you want to show labels for points or not (default: False)
    size : int
        size of points in the scatterplot
    n_neighbors : int
        n_neighbors hyperparameter of UMAP. See the documentation.
    title : str
        title for the visualization plot

    Returns
    -----------
    None. Shows the clusters.
    """

    reducer = umap.UMAP(n_neighbors=n_neighbors, random_state=42)
    Z = reducer.fit_transform(data)  # reduce dimensionality
    umap_df = pd.DataFrame(data=Z, columns=["dim1", "dim2"])
    umap_df["cluster"] = cluster_labels

    if ignore_noise:
        umap_df = umap_df[umap_df["cluster"] != -1]

    labels = np.unique(umap_df["cluster"])

    fig, ax = plt.subplots(figsize=(6, 5))
    ax.set_title(title)

    scatter = ax.scatter(
        umap_df["dim1"],
        umap_df["dim2"],
        c=umap_df["cluster"],
        cmap="tab20b",
        s=size,
        #edgecolors="k",
        #linewidths=0.1,
    )

    legend = ax.legend(*scatter.legend_elements(), loc="best", title="Cluste
    ax.add_artist(legend)

    if show_labels:
        x = umap_df["dim1"].tolist()
        y = umap_df["dim2"].tolist()
        for i, txt in enumerate(raw_sents):
            ax.annotate(" ".join(txt.split()[:10]), (x[i], y[i]))
    plt.show()
```

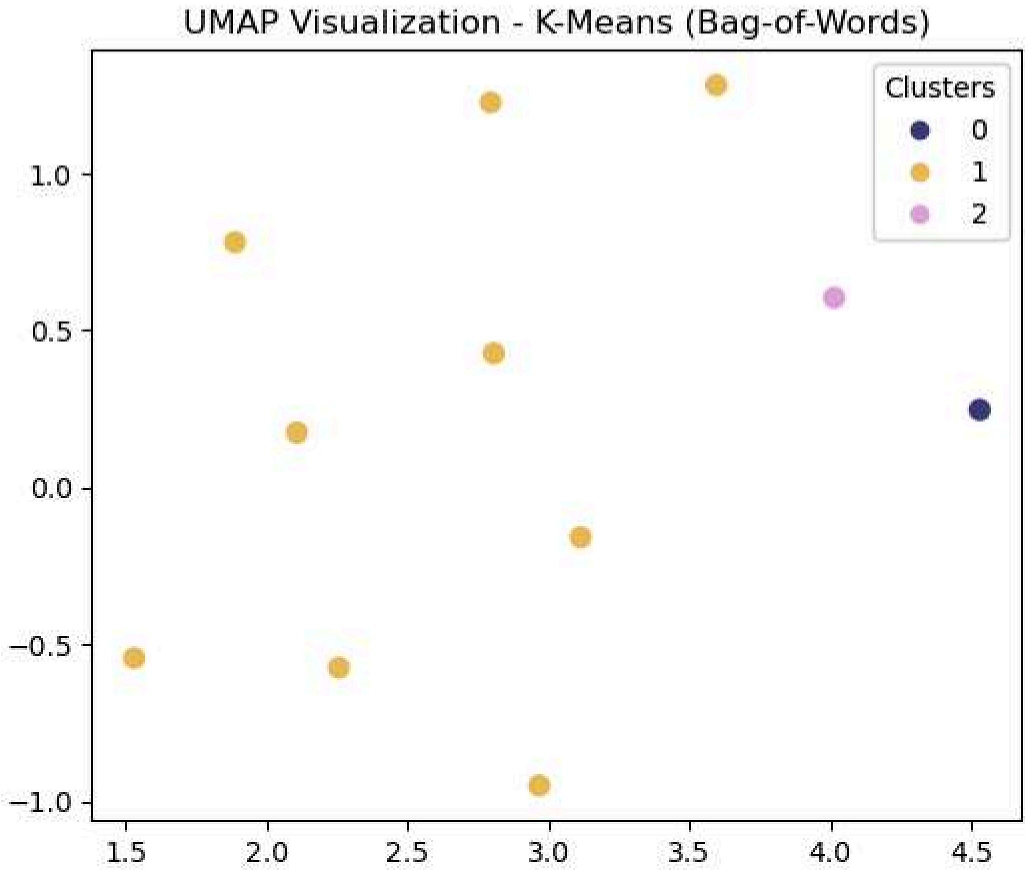
Solution_1.7

Points: 4

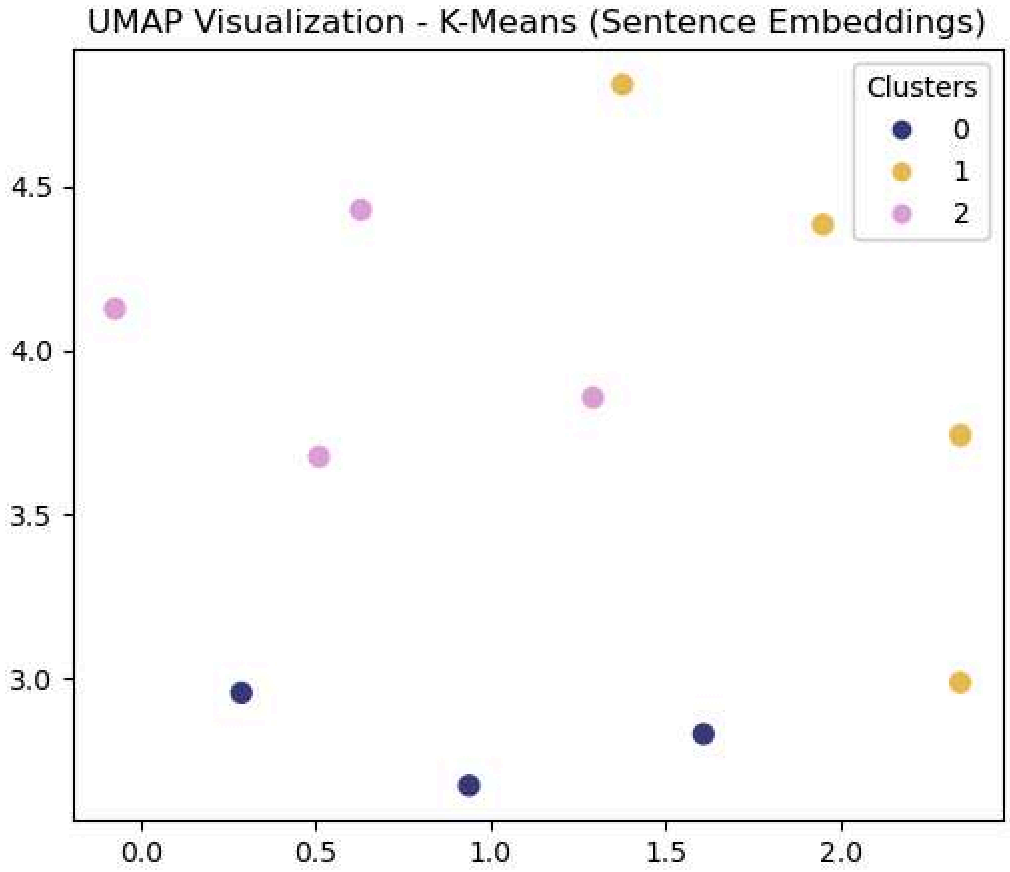
In [24]:

```
# K-Means with Bag-of-Words Representation
plot_umap_clusters(
    data=X_counts.toarray(),
    cluster_labels=kmeans_bow_labels,
    raw_sents=wiki_df["text"],
    title="UMAP Visualization - K-Means (Bag-of-Words)"
)

# K-Means with Sentence Embedding Representation
plot_umap_clusters(
    data=emb_sents,
    cluster_labels=kmeans_emb_labels,
    raw_sents=wiki_df["text"],
    title="UMAP Visualization - K-Means (Sentence Embeddings)"
)

# DBSCAN with Sentence Embedding Representation
plot_umap_clusters(
    data=emb_sents,
    cluster_labels=dbscan_emb_labels,
    raw_sents=wiki_df["text"],
    title="UMAP Visualization - DBSCAN (Sentence Embeddings)",
    ignore_noise=True  # Ignore noise points for better visualization
)

# Hierarchical Clustering with Sentence Embedding Representation
plot_umap_clusters(
    data=emb_sents,
    cluster_labels=hier_emb_labels,
    raw_sents=wiki_df["text"],
    title="UMAP Visualization - Hierarchical Clustering (Sentence Embeddings
)
```

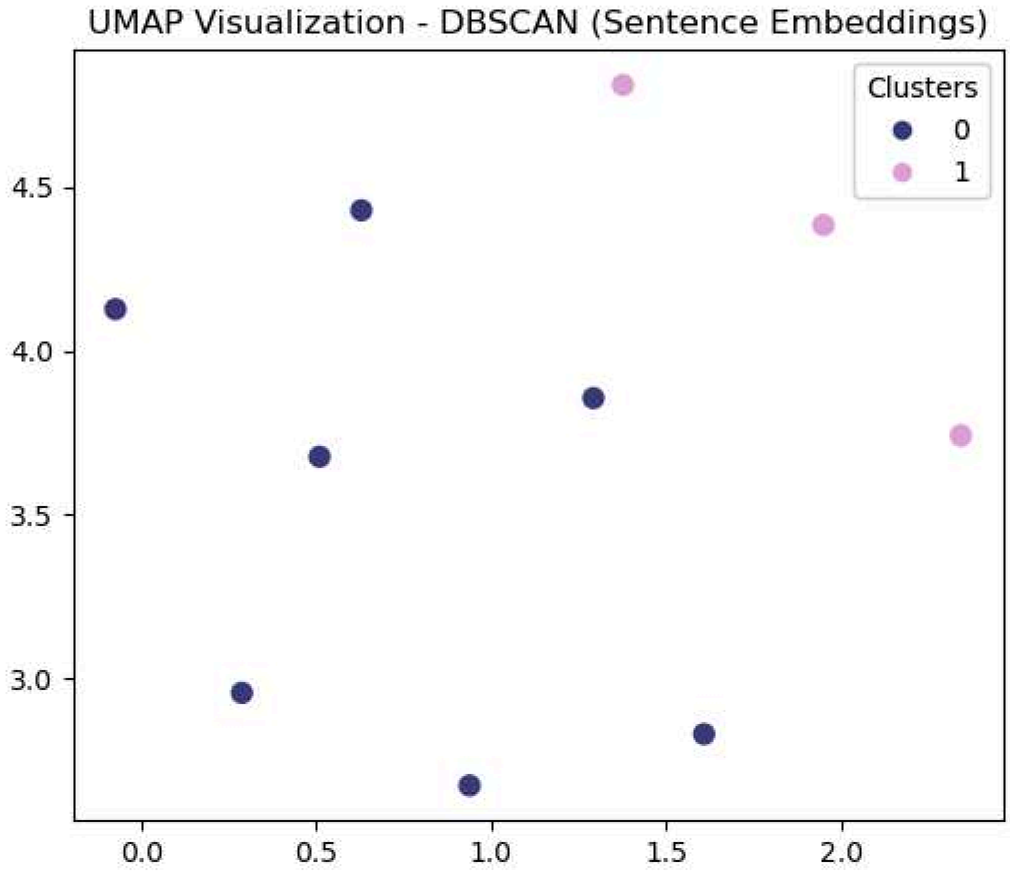
```
/opt/anaconda3/envs/cpsc330/lib/python3.12/site-packages/umap/umap_.py:1952:
UserWarning: n_jobs value 1 overridden to 1 by setting random_state. Use no
seed for parallelism.
  warn(
/opt/anaconda3/envs/cpsc330/lib/python3.12/site-packages/umap/umap_.py:2462:
UserWarning: n_neighbors is larger than the dataset size; truncating to X.sh
ape[0] - 1
  warn(
```



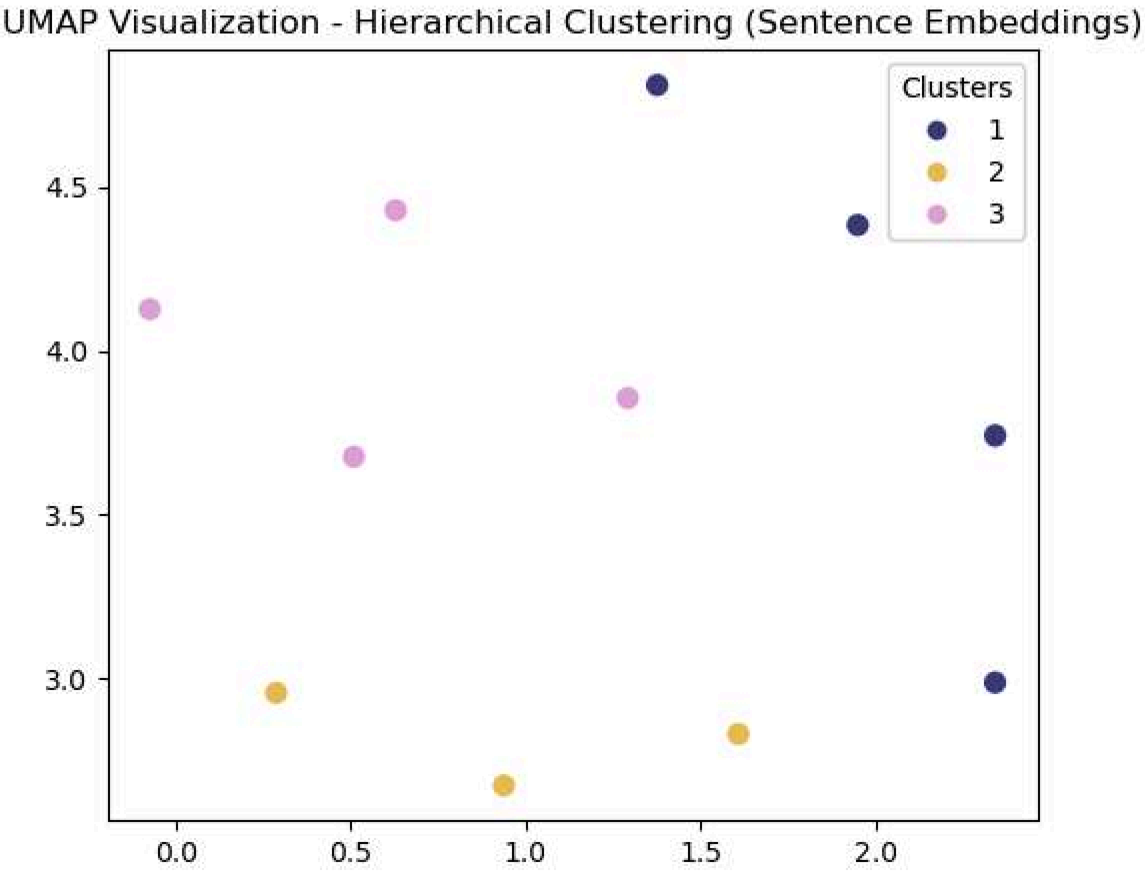
```
/opt/anaconda3/envs/cpsc330/lib/python3.12/site-packages/umap/umap_.py:1952:
UserWarning: n_jobs value 1 overridden to 1 by setting random_state. Use no
seed for parallelism.
  warn(
/opt/anaconda3/envs/cpsc330/lib/python3.12/site-packages/umap/umap_.py:2462:
UserWarning: n_neighbors is larger than the dataset size; truncating to X.sh
ape[0] - 1
  warn(
```



```
/opt/anaconda3/envs/cpsc330/lib/python3.12/site-packages/umap/umap_.py:1952:
UserWarning: n_jobs value 1 overridden to 1 by setting random_state. Use no
seed for parallelism.
  warn(
/opt/anaconda3/envs/cpsc330/lib/python3.12/site-packages/umap/umap_.py:2462:
UserWarning: n_neighbors is larger than the dataset size; truncating to X.sh
ape[0] - 1
  warn(
```



```
/opt/anaconda3/envs/cpsc330/lib/python3.12/site-packages/umap/umap_.py:1952:
UserWarning: n_jobs value 1 overridden to 1 by setting random_state. Use no
seed for parallelism.
  warn(
/opt/anaconda3/envs/cpsc330/lib/python3.12/site-packages/umap/umap_.py:2462:
UserWarning: n_neighbors is larger than the dataset size; truncating to X.sh
ape[0] - 1
  warn(
```



Exercise 2: Food.com recipes

Now that we have applied document clustering on a toy corpus, let's move to a more realistic corpus.

In the lecture, we worked on an activity of manually clustering food items and discussed challenges associated with it. We also applied different clustering algorithms to cluster food images. We'll continue this theme of clustering food items in this lab. But instead of images we will cluster textual description of food items, i.e., recipe names.

In this lab, we will work with a sample of Kaggle's Food.com recipes corpus. This corpus contains 180K+ recipes and 700K+ recipe reviews. In this lab, we'll only focus on recipes and **not** on reviews. The recipes are present in `RAW_recipes.csv`. Our goal is to find categories or groupings of recipes from this corpus based on their names.

Your tasks: