Install dependencies

In [279…
```
%pip install "bertopic[visualization]" sentence-transformers umap-learn hdbscan scikit-learn
```

```
huggingface/tokenizers: The current process just got forked, after parallelism 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=(true | false)
Requirement already satisfied: sentence-transformers in /opt/anaconda3/lib/python3.12/site-packages (5.1.0)
Requirement already satisfied: umap-learn in /opt/anaconda3/lib/python3.12/site-packages (0.5.9.post2)
Requirement already satisfied: hdbscan in /opt/anaconda3/lib/python3.12/site-packages (0.8.40)
Requirement already satisfied: scikit-learn in /opt/anaconda3/lib/python3.12/site-packages (1.7.1)
Requirement already satisfied: bertopic[visualization] in /opt/anaconda3/lib/python3.12/site-packages (0.17.3)
WARNING: bertopic 0.17.3 does not provide the extra 'visualization'
Requirement already satisfied: numpy>=1.20.0 in /opt/anaconda3/lib/python3.12/site-packages (from bertopic[visualization]) (1.26.4)
Requirement already satisfied: pandas>=1.1.5 in /opt/anaconda3/lib/python3.12/site-packages (from bertopic[visualization]) (2.2.2)
Requirement already satisfied: plotly>=4.7.0 in /opt/anaconda3/lib/python3.12/site-packages (from bertopic[visualization]) (5.24.1)
Requirement already satisfied: tqdm>=4.41.1 in /opt/anaconda3/lib/python3.12/site-packages (from bertopic[visualization]) (4.66.5)
Requirement already satisfied: llvmlite>0.36.0 in /opt/anaconda3/lib/python3.12/site-packages (from bertopic[visualization]) (0.43.0)
Requirement already satisfied: transformers<5.0.0,>=4.41.0 in /opt/anaconda3/lib/python3.12/site-packages (from sentence-transformers) (4.56.0)
Requirement already satisfied: torch>=1.11.0 in /opt/anaconda3/lib/python3.12/site-packages (from sentence-transformers) (2.8.0)
Requirement already satisfied: scipy in /opt/anaconda3/lib/python3.12/site-packages (from sentence-transformers) (1.13.1)
Requirement already satisfied: huggingface-hub>=0.20.0 in /opt/anaconda3/lib/python3.12/site-packages (from sentence-transformers) (0.34.4)
Requirement already satisfied: Pillow in /opt/anaconda3/lib/python3.12/site-packages (from sentence-transformers) (10.4.0)
Requirement already satisfied: typing_extensions>=4.5.0 in /opt/anaconda3/lib/python3.12/site-packages (from sentence-transformers) (4.11.0)
Requirement already satisfied: filelock in /opt/anaconda3/lib/python3.12/site-packages (from transformers<5.0.0,>=4.41.0->sentence-transformers) (3.13.1)
Requirement already satisfied: packaging>=20.0 in /opt/anaconda3/lib/python3.12/site-packages (from transformers<5.0.0,>=4.41.0->sentence-transformers)
(24.1)
Requirement already satisfied: pyyaml>=5.1 in /opt/anaconda3/lib/python3.12/site-packages (from transformers<5.0.0,>=4.41.0->sentence-transformers) (6.0.
1)
Requirement already satisfied: regex!=2019.12.17 in /opt/anaconda3/lib/python3.12/site-packages (from transformers<5.0.0,>=4.41.0->sentence-transformers)
(2024.9.11)
Requirement already satisfied: requests in /opt/anaconda3/lib/python3.12/site-packages (from transformers<5.0.0,>=4.41.0->sentence-transformers) (2.32.3)
Requirement already satisfied: tokenizers<=0.23.0,>=0.22.0 in /opt/anaconda3/lib/python3.12/site-packages (from transformers<5.0.0,>=4.41.0->sentence-tra
nsformers) (0.22.0)
Requirement already satisfied: safetensors>=0.4.3 in /opt/anaconda3/lib/python3.12/site-packages (from transformers<5.0.0,>=4.41.0->sentence-transformer
s) (0.6.2)
Requirement already satisfied: fsspec>=2023.5.0 in /opt/anaconda3/lib/python3.12/site-packages (from huggingface-hub>=0.20.0->sentence-transformers) (202
4.6.1)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /opt/anaconda3/lib/python3.12/site-packages (from huggingface-hub>=0.20.0->sentence-transformers)
(1.1.9)
Requirement already satisfied: numba>=0.51.2 in /opt/anaconda3/lib/python3.12/site-packages (from umap-learn) (0.60.0)
Requirement already satisfied: pynndescent>=0.5 in /opt/anaconda3/lib/python3.12/site-packages (from umap-learn) (0.5.13)
Requirement already satisfied: joblib>=1.0 in /opt/anaconda3/lib/python3.12/site-packages (from hdbscan) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /opt/anaconda3/lib/python3.12/site-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/anaconda3/lib/python3.12/site-packages (from pandas>=1.1.5->bertopic[visualization]) (2.9.
0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/anaconda3/lib/python3.12/site-packages (from pandas>=1.1.5->bertopic[visualization]) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /opt/anaconda3/lib/python3.12/site-packages (from pandas>=1.1.5->bertopic[visualization]) (2023.3)
Requirement already satisfied: tenacity>=6.2.0 in /opt/anaconda3/lib/python3.12/site-packages (from plotly>=4.7.0->bertopic[visualization]) (8.2.3)
Requirement already satisfied: six>=1.5 in /opt/anaconda3/lib/python3.12/site-packages (from python-dateutil>=2.8.2->pandas>=1.1.5->bertopic[visualizatio
n]) (1.16.0)
Requirement already satisfied: setuptools in /opt/anaconda3/lib/python3.12/site-packages (from torch>=1.11.0->sentence-transformers) (75.1.0)
Requirement already satisfied: sympy>=1.13.3 in /opt/anaconda3/lib/python3.12/site-packages (from torch>=1.11.0->sentence-transformers) (1.14.0)
Requirement already satisfied: networkx in /opt/anaconda3/lib/python3.12/site-packages (from torch>=1.11.0->sentence-transformers) (3.3)
Requirement already satisfied: jinja2 in /opt/anaconda3/lib/python3.12/site-packages (from torch>=1.11.0->sentence-transformers) (3.1.4)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /opt/anaconda3/lib/python3.12/site-packages (from sympy>=1.13.3->torch>=1.11.0->sentence-transformer
s) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/anaconda3/lib/python3.12/site-packages (from jinja2->torch>=1.11.0->sentence-transformers) (2.1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /opt/anaconda3/lib/python3.12/site-packages (from requests->transformers<5.0.0,>=4.41.0->sente
nce-transformers) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /opt/anaconda3/lib/python3.12/site-packages (from requests->transformers<5.0.0,>=4.41.0->sentence-transfor
mers) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/anaconda3/lib/python3.12/site-packages (from requests->transformers<5.0.0,>=4.41.0->sentence-tr
ansformers) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in /opt/anaconda3/lib/python3.12/site-packages (from requests->transformers<5.0.0,>=4.41.0->sentence-tr
ansformers) (2025.1.31)
Note: you may need to restart the kernel to use updated packages.
```

Import libraries and set paths

In [282…
```
# import the necessary libraries
import os
import re
import pandas as pd
from pathlib import Path
```

In [284…
```
# Paths
DATA_FILE = "349Articles.csv" # filename
OUT_DIR   = Path("bertopic_outputs") # outputs folder
OUT_DIR.mkdir(exist_ok=True)
```

Load dataset

In [287…
```
df = pd.read_csv(DATA_FILE, dtype=str, encoding="utf-8", low_memory=False) # Load CSV file
# Show first columns
print("Columns available:", list(df.columns)[:15])
print("Total rows:", len(df))
```

```
Columns available: ['Authors', 'Author full names', 'Author(s) ID', 'Title', 'Year', 'Source title', 'Volume', 'Issue', 'Art. No.', 'Page start', 'Page e
nd', 'Page count', 'Cited by', 'DOI', 'Link']
Total rows: 349
```

Extract Abstracts, Titles, and Keywords

In [290…
```
# Candidate column names
abstract_candidates = ["Abstract", "Abstracts", "abstract", "AB", "Description"]
title_candidates    = ["Title", "Document Title", "title", "TI"]
kw_candidates       = ["Author Keywords", "authkeywords", "author keywords", "AU Keywords", "Index Keywords", "index keywords"]
```

In [292…
```
def first_col(cols):
    for c in cols:
        if c in df.columns:
```

```
            return c
    return None
```

In [294…
```
AB_COL = first_col(abstract_candidates) # abstract column
TI_COL = first_col(title_candidates) # title column
KW_COL = first_col(kw_candidates) # keywords column
```

In [296…
```
if AB_COL is None and TI_COL is None:
    raise ValueError(f"Could not locate Abstract or Title columns. Columns found: {list(df.columns)[:20]}")
```

In [298…
```
# Text cleaning
def clean(s):
    if pd.isna(s):
        return ""
    s = str(s)
    s = re.sub(r"\s+", " ", s) # collapse whitespace
    return s.strip()
```

In [300…
```
abstracts = df[AB_COL].map(clean) if AB_COL else pd.Series([""]*len(df))
titles    = df[TI_COL].map(clean) if TI_COL else pd.Series([""]*len(df))
kws       = df[KW_COL].fillna("").astype(str) if KW_COL else pd.Series([""]*len(df))
```

In [302…
```
if KW_COL and KW_COL in df.columns:
    kws = df[KW_COL].fillna("").astype(str)
else:
    kws = pd.Series([""]*len(df))
```

In [304…
```
texts = []
for a, t, k in zip(abstracts, titles, kws):
    text = a
    if len(text) < 200:
        add = " ".join([t, k])
        text = (text + " " + add).strip()
    texts.append(text)
```

In [306…
```
docs = [x for x in texts if len(x.split()) >= 20]
orig_idx = [i for i, x in enumerate(texts) if len(x.split()) >= 20]
```

In [308…
```
print(f"Loaded {len(df)} rows; using {len(docs)} documents with >= 20 words.")
```

```
Loaded 349 rows; using 349 documents with >= 20 words.
```

Import BERTopic and models

In [311…
```
# Topic modeling setup
from bertopic import BERTopic
from sentence_transformers import SentenceTransformer
import umap
import hdbscan
```

In [313…
```
# Sentence encoder
encoder = SentenceTransformer("all-MiniLM-L6-v2")
```

In [314…
```
# Dimensionality reduction
umap_model = umap.UMAP(
    n_neighbors=15,  # local neighborhood size
    n_components=5, # low-dimensional space
    min_dist=0.0, # tight clusters
    metric="cosine",
    random_state=42,
)

# clustering
hdbscan_model = hdbscan.HDBSCAN(
    min_cluster_size=15, # minimum docs per topic
    min_samples=5, # noise sensitivity
    metric="euclidean",
    cluster_selection_method="eom",
    prediction_data=True
)

# Topic model
topic_model = BERTopic(
    embedding_model=encoder,
    umap_model=umap_model,
    hdbscan_model=hdbscan_model,
    language="english",
    calculate_probabilities=True,
    verbose=True,
    nr_topics=None, # let HDBSCAN decide
    top_n_words=10 # top terms per topic
)
```

Fit model

In [318…
```
topics, probs = topic_model.fit_transform(docs)
print("Unique topics (incl. -1 = outliers):", len(set(topics)))
```

```
2025-08-31 00:39:18,124 - BERTopic - Embedding - Transforming documents to embeddings.
Batches:   0%|          | 0/11 [00:00<?, ?it/s]
2025-08-31 00:39:19,483 - BERTopic - Embedding - Completed ✓
2025-08-31 00:39:19,484 - BERTopic - Dimensionality - Fitting the dimensionality reduction algorithm
2025-08-31 00:39:19,948 - BERTopic - Dimensionality - Completed ✓
2025-08-31 00:39:19,949 - BERTopic - Cluster - Start clustering the reduced embeddings
2025-08-31 00:39:19,970 - BERTopic - Cluster - Completed ✓
2025-08-31 00:39:19,972 - BERTopic - Representation - Fine-tuning topics using representation models.
2025-08-31 00:39:20,022 - BERTopic - Representation - Completed ✓
Unique topics (incl. -1 = outliers): 7
```

Export topic info and top terms

In [320…
```
# Topic summary table (topic id, size, label) → CSV
topic_info = topic_model.get_topic_info()
```

```
topic_info.to_csv(OUT_DIR/"topic_info.csv", index=False)
```

In [323…
```
# Long table of top terms per topic (skip -1 which is noise/outliers)
rows = []
for t_id in topic_info["Topic"].tolist():
    if t_id == -1:  # outliers
        continue
    words = topic_model.get_topic(t_id)
    rows.append({
        "topic_id": t_id,
        "size": int(topic_info.loc[topic_info["Topic"]==t_id, "Count"].values[0]),
        "terms": ", ".join([w for w, _ in words]),
    })
pd.DataFrame(rows).to_csv(OUT_DIR/"topics_top_terms.csv", index=False)
```

Export document–topic assignments

In [326…
```
# Per-document topics and max probability
assignments = pd.DataFrame({
    "orig_row": orig_idx,
    "topic": topics,
    "prob_max": probs.max(axis=1) if probs is not None else None,
    "text": docs
})
assignments.to_csv(OUT_DIR/"doc_topics.csv", index=False)
```

Export interactive HTML visuals

In [329…
```
#  Intertopic distance map (UMAP in 2D)
fig_itm = topic_model.visualize_topics(width=1200, height=800)
fig_itm.write_html(str(OUT_DIR/"intertopic_map.html"))


#  Topic bar chart (by frequency)
fig_bar = topic_model.visualize_barchart(top_n_topics=20, width=1200, height=800)
fig_bar.write_html(str(OUT_DIR/"topics_barchart.html"))


#  Hierarchical clusters (dendrogram)
fig_hier = topic_model.visualize_hierarchy(width=1200, height=800)
fig_hier.write_html(str(OUT_DIR/"topics_hierarchy.html"))

#  Heatmap of topic similarity
fig_heat = topic_model.visualize_heatmap(width=1200, height=900)
fig_heat.write_html(str(OUT_DIR/"topics_heatmap.html"))
```

In [331…
```
print("\nSaved outputs to:", OUT_DIR.resolve())
```

```
Saved outputs to: /Users/ohannz/bertopic_outputs
```