

Start coding or [generate](#) with AI.

```
# %%capture
!pip install bertopic datasets openai datamaplot
```

Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.8->data)

Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.8->data)

Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.10/dist-packages (from numba>=0.56)

Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.5->bertopic)

Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.5->bertopic)

Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly>=4.7.0->bertopic)

Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from pydantic<3,>=1.9)

Requirement already satisfied: pydantic-core==2.23.4 in /usr/local/lib/python3.10/dist-packages (from pydantic<3,>=1.9)

Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.3)

Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->data)

Requirement already satisfied: networkx>=2.8 in /usr/local/lib/python3.10/dist-packages (from scikit-image>=0.22->data)

Requirement already satisfied: imageio>=2.33 in /usr/local/lib/python3.10/dist-packages (from scikit-image>=0.22->data)

Requirement already satisfied: tifffile>=2022.8.12 in /usr/local/lib/python3.10/dist-packages (from scikit-image>=0.22->data)

Requirement already satisfied: lazy-loader>=0.4 in /usr/local/lib/python3.10/dist-packages (from scikit-image>=0.22->data)

Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.2)

Requirement already satisfied: transformers<5.0.0,>=4.41.0 in /usr/local/lib/python3.10/dist-packages (from sentence-transformers>=0.2)

Requirement already satisfied: torch>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from sentence-transformers>=0.2)

Collecting pynndescent>=0.5 (from umap-learn>=0.5.0->bertopic)

Downloading pynndescent-0.5.13-py3-none-any.whl.metadata (6.8 kB)

Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->datamaplot) (2.1.5)

Collecting Pyqtreet>=2.0.0 (from pylabadjust->datamaplot)

Downloading Pyqtreet-1.0.0.tar.gz (5.2 kB)

Preparing metadata (setup.py) ... done

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib)

Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->sentence-transformers)

Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch)

Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers<5.0.0,>=4.41.0)

Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from transformers<5.0.0,>=4.41.0)

Requirement already satisfied: tokenizers<0.21,>=0.20 in /usr/local/lib/python3.10/dist-packages (from transformers<5.0.0,>=4.41.0)

Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.10/dist-packages (from yarl<2.0,>=1.12.0->aioscrape)

Requirement already satisfied: click>=8.1 in /usr/local/lib/python3.10/dist-packages (from dask>=2024.9.0->datashader)

Requirement already satisfied: cloudpickle>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from dask>=2024.9.0->datashader)

Requirement already satisfied: partd>=1.4.0 in /usr/local/lib/python3.10/dist-packages (from dask>=2024.9.0->datashader)

Requirement already satisfied: importlib-metadata>=4.13.0 in /usr/local/lib/python3.10/dist-packages (from dask>=2024.9.0->datashader)

Requirement already satisfied: zipp>=3.20 in /usr/local/lib/python3.10/dist-packages (from importlib-metadata>=4.13.0->dask)

Requirement already satisfied: locket in /usr/local/lib/python3.10/dist-packages (from partd>=1.4.0->dask->datashader)

Downloading bertopic-0.16.4-py3-none-any.whl (143 kB)

143.7/143.7 kB 9.9 MB/s eta 0:00:00

Downloading datasets-3.1.0-py3-none-any.whl (480 kB)

480.6/480.6 kB 25.0 MB/s eta 0:00:00

Downloading datamaplot-0.4.2-py3-none-any.whl (72 kB)

72.4/72.4 kB 7.4 MB/s eta 0:00:00

Downloading colorspacious-1.1.2-py2.py3-none-any.whl (37 kB)

Downloading datashader-0.16.3-py2.py3-none-any.whl (18.3 MB)

18.3/18.3 MB 51.7 MB/s eta 0:00:00

Downloading dill-0.3.8-py3-none-any.whl (116 kB)

116.3/116.3 kB 2.7 MB/s eta 0:00:00

Downloading fsspec-2024.9.0-py3-none-any.whl (179 kB)

179.3/179.3 kB 12.2 MB/s eta 0:00:00

Downloading hdbscan-0.8.39-cp310-cp310-manylinux_2_17_x86_64_manylinux2014_x86_64.whl (4.2 MB)

4.2/4.2 MB 55.9 MB/s eta 0:00:00

Downloading multiprocessing-0.70.16-py310-none-any.whl (134 kB)

134.8/134.8 kB 10.0 MB/s eta 0:00:00

Downloading rcssmin-1.1.3-cp310-cp310-manylinux1_x86_64.whl (49 kB)

49.8/49.8 kB 3.3 MB/s eta 0:00:00

Downloading rjsmin-1.2.3-cp310-cp310-manylinux1_x86_64.whl (34 kB)

34.0/34.0 kB 10.0 MB/s eta 0:00:00

Downloading umap-learn-0.5.7-py3-none-any.whl (88 kB)

88.0/88.0 kB 7.2 MB/s eta 0:00:00

ArXiv Articles:Computation and Language

```
# load data from hugging face
from datasets import load_dataset
dataset = load_dataset("maartengr/arxiv_nlp")["train"]
# print(dataset)
```

```
#extract metadata
abstracts = dataset['Abstracts']
print(abstracts)
titles = dataset['Titles']
print(titles)
```

```
[' In this paper Arabic was investigated from the speech recognition problem\npoint of view. We propose a novel approach\nfor Arabic Speech Recognition Using CMUSphinx System', 'Arabic Speech Recognition System using CMU-Sphinx4']
```

✓ A Common Pipeline for Text Clustering

✓ 1.Embedding Document

```
from sentence_transformers import SentenceTransformer
embedding_model = SentenceTransformer('thenlper/gte-small')
embeddings = embedding_model.encode(abstracts, show_progress_bar = True)
print(embeddings)
```

↗ The cache for model files in Transformers v4.22.0 has been updated. Migrating your old cache. This is a one-time only op

```
0it [00:00, ?it/s]
modules.json: 0%|          | 0.00/385 [00:00<?, ?B/s]
README.md: 0%|          | 0.00/68.1k [00:00<?, ?B/s]
sentence_bert_config.json: 0%|          | 0.00/57.0 [00:00<?, ?B/s]
config.json: 0%|          | 0.00/583 [00:00<?, ?B/s]
model.safetensors: 0%|          | 0.00/66.7M [00:00<?, ?B/s]
tokenizer_config.json: 0%|          | 0.00/394 [00:00<?, ?B/s]
vocab.txt: 0%|          | 0.00/232k [00:00<?, ?B/s]
tokenizer.json: 0%|          | 0.00/712k [00:00<?, ?B/s]
special_tokens_map.json: 0%|          | 0.00/125 [00:00<?, ?B/s]
1_Pooling/config.json: 0%|          | 0.00/190 [00:00<?, ?B/s]
Batches: 0%|          | 0/1405 [00:00<?, ?it/s]
[[-8.3818287e-02  4.1654281e-02  1.3067336e-02 ...  1.2165051e-02
 -4.8086634e-03 -4.0820194e-03]
 [-8.7145895e-02  4.0569924e-02  1.8635092e-02 ...  1.4508341e-02
 -4.5975532e-05  3.1807183e-03]
 [-8.3583340e-02 -1.3517680e-02  5.2751530e-02 ...  6.2981874e-02
  5.3653855e-02 -1.1938489e-02]
 ...
 [-6.5208301e-02 -4.2661652e-03  2.7181083e-02 ...  3.7392065e-02
 -3.7366904e-03 -3.1422281e-03]
 [-5.7497695e-02 -3.3351649e-02 -1.8073303e-03 ... -1.4886928e-02
  1.6042769e-02  1.2014104e-02]
 [-6.9950350e-02 -7.1072794e-04  2.5283867e-02 ...  4.3381646e-02
 -1.3664219e-03  6.2695227e-04]]
```

```
# Check the dimensions of the resulting embeddings
```

```
print(embeddings.shape)
```

↗ (44949, 384)

✓ 2. Reduce the Dimentionality of Embeddings

```
from umap import UMAP
```

```
# We reduce the input embeddings from 384 dimenions to 5 dimenions
umap_model = UMAP(
    n_components=5, min_dist=0.0, metric='cosine', random_state=42
)
reduced_embeddings = umap_model.fit_transform(embeddings)
```

↗ /usr/local/lib/python3.10/dist-packages/umap/umap_.py:1952: UserWarning: n_jobs value 1 overridden to 1 by setting rando
warn(

✓ 3.Cluster the Reduced Embedding

```
from hdbscan import HDBSCAN
```

```
# We fit the model and extract the clusters
hdbscan_model = HDBSCAN(
    min_cluster_size=50, metric='euclidean', cluster_selection_method='eom'
).fit(reduced_embeddings)
clusters = hdbscan_model.labels_
```

```
# How many clusters did we generate?
len(set(clusters))
```

↗ 153

✓ Inspecting the Clusters

```
import numpy as np
```

```
# Print first three documents in cluster 0
cluster = 0
for index in np.where(clusters==cluster)[0][:3]:
    print(abstracts[index][:300] + "... \n")
```

↗ This works aims to design a statistical machine translation from English text to American Sign Language (ASL). The system is based on Moses tool with some modifications and the results are synthesized through a 3D avatar for interpretation. First, we translate the input text to gloss, a written fo...

Researches on signed languages still strongly dissociate linguistic issues related on phonological and phonetic aspects, and gesture studies for recognition and synthesis purposes. This paper focuses on the imbrication of motion and meaning for the analysis, synthesis and evaluation of sign lang...

Modern computational linguistic software cannot produce important aspects of sign language translation. Using some researches we deduce that the majority of automatic sign language translation systems ignore many aspects when they generate animation; therefore the interpretation lost the truth inf...

\Next, we reduce our embeddings to 2-dimensions so that we can plot them and get a rough understanding of the generated clusters.

```
import pandas as pd
```

```
# Reduce 384-dimensional embeddings to 2 dimensions for easier visualization
reduced_embeddings = UMAP(
    n_components=2, min_dist=0.0, metric='cosine', random_state=42
).fit_transform(embeddings)
```

```
# Create dataframe
df = pd.DataFrame(reduced_embeddings, columns=["x", "y"])
df["title"] = titles
df["cluster"] = [str(c) for c in clusters]
```


```
# Select outliers and non-outliers (clusters)
clusters_df = df.loc[df.cluster != "-1", :]
outliers_df = df.loc[df.cluster == "-1", :]
```

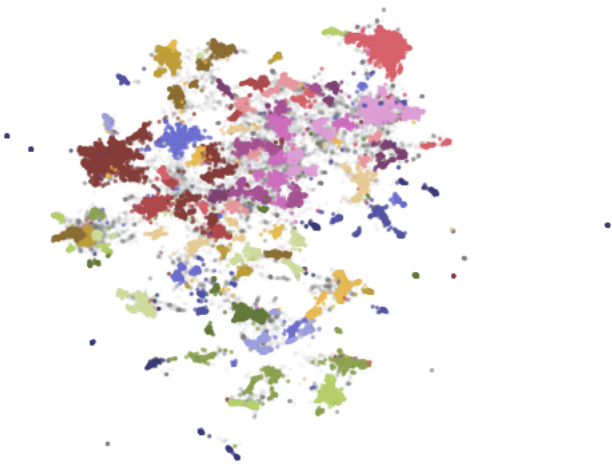
↗ /usr/local/lib/python3.10/dist-packages/umap/umap_.py:1952: UserWarning: n_jobs value 1 overridden to 1 by setting random_state

✓ Static Plot

```
import matplotlib.pyplot as plt
```

```
# Plot outliers and non-outliers separately
plt.scatter(outliers_df.x, outliers_df.y, alpha=0.05, s=2, c="grey")
plt.scatter(
    clusters_df.x, clusters_df.y, c=clusters_df.cluster.astype(int),
    alpha=0.6, s=2, cmap='tab20b'
)
plt.axis('off')
# plt.savefig("matplotlib.png", dpi=300) # Uncomment to save the graph as a .png
```

 (-7.778345727920533,
10.878833436965943,
-1.711702972650528,
16.388065367937088)




- ✖ From Text Clustering to Topic Modeling
- ✖ BERTopic: A Modular Topic Modeling Framework

```
# !pip install bertopic
from bertopic import BERTopic

# Train our model with our previously defined models
topic_model = BERTopic(
    embedding_model=embedding_model,
    umap_model=umap_model,
    hdbscan_model=hdbscan_model,
    verbose=True
).fit(abstracts, embeddings)


2024-11-16 08:04:50,187 - BERTopic - Dimensionality - Fitting the dimensionality reduction algorithm
2024-11-16 08:05:55,676 - BERTopic - Dimensionality - Completed ✓
2024-11-16 08:05:55,680 - BERTopic - Cluster - Start clustering the reduced embeddings
2024-11-16 08:05:57,776 - BERTopic - Cluster - Completed ✓
2024-11-16 08:05:57,794 - BERTopic - Representation - Extracting topics from clusters using representation models.
2024-11-16 08:06:02,801 - BERTopic - Representation - Completed ✓
```

topic_model.get_topic_info()



	Topic	Count	Name	Representation	Representative_Docs
0	-1	14462	-1_the_of_and_to	[the, of, and, to, in, we, for, that, language...	[Cross-lingual text classification aims at t...
1	0	2241	0_question_questions_qa_answer	[question, questions, qa, answer, answering, a...	[Question generation (QG) attempts to solve ...
2	1	2098	1_speech_asr_recognition_end	[speech, asr, recognition, end, acoustic, audi...	[End-to-end models have achieved impressive ...
3	2	903	2_image_visual_multimodal_images	[image, visual, multimodal, images, vision, mo...	[In this paper we propose a model to learn m...
4	3	887	3_summarization_summaries_summary_abstractive	[summarization, summaries, summary, abstractiv...	[We present a novel divide-and-conquer metho...
...
148	147	54	147_counseling_mental_therapy_health	[counseling, mental, therapy, health, psychoth...	[Mental health care poses an increasingly se...
149	148	53	148_chatgpt_its_openai_has	[chatgpt, its, openai, has, it, tasks, capabil...	[Over the last few years, large language mod...

topic_model.get_topic(0)

 [('question', 0.021262463291547223),
('questions', 0.015866039067984204),
('qa', 0.015830640927795868),

```
(('answer', 0.015787698152510205),
 ('answering', 0.014859992848422435),
 ('answers', 0.00992918704536005),
 ('retrieval', 0.009497931820914705),
 ('comprehension', 0.007719047154229789),
 ('reading', 0.007175282051339653),
 ('knowledge', 0.0063049421989358])
```

```
topic_model.find_topics("topic modeling")
```

```
↗ ([22, -1, 50, 38, 84],
 [0.95448655, 0.91218555, 0.9067658, 0.9051957, 0.9026561])
```

```
topic_model.get_topic(22)
```

```
↗ [('topic', 0.06782148231481569),
 ('topics', 0.03509097163093816),
 ('lda', 0.0162350543969945),
 ('latent', 0.013482620892138605),
 ('document', 0.01258276852968132),
 ('documents', 0.012463820004375148),
 ('modeling', 0.011571581804609226),
 ('dirichlet', 0.009901318233964887),
 ('word', 0.00852094200971816),
 ('allocation', 0.007792539607690728)]
```

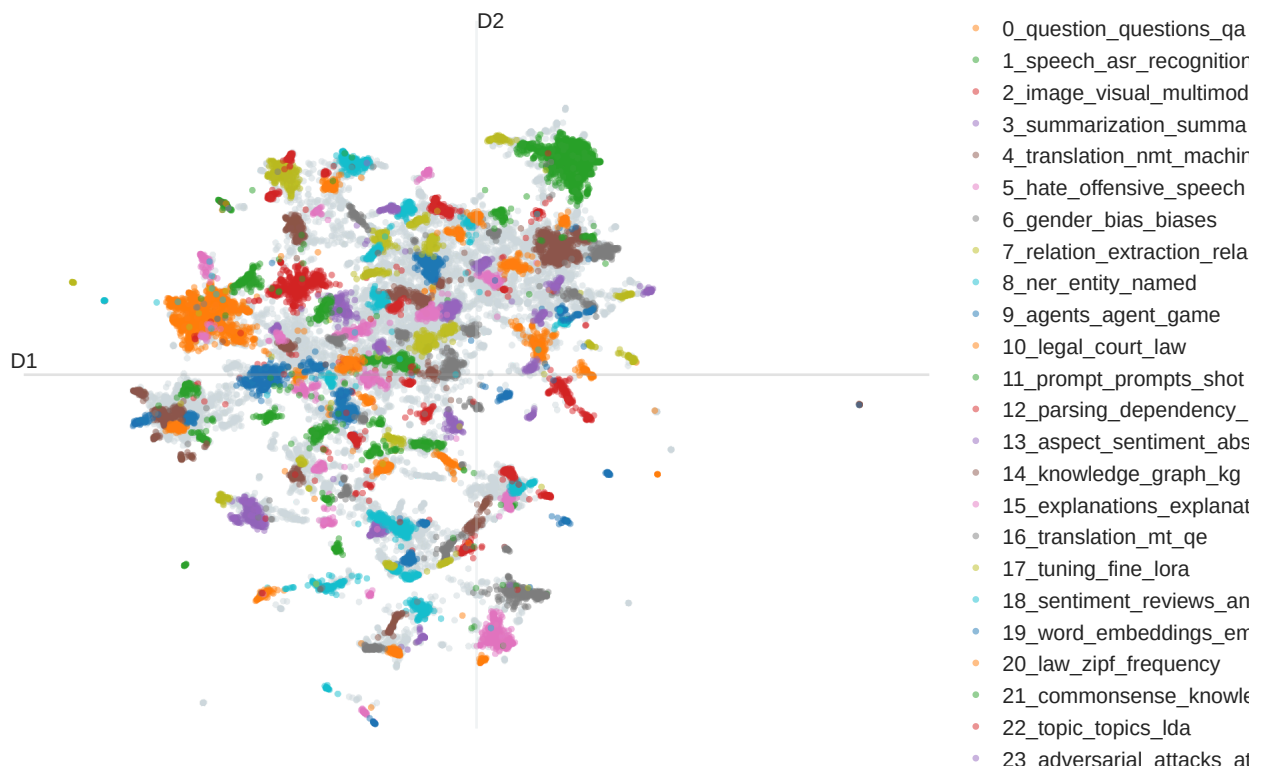
Visualizations

```
# Visualize topics and documents
fig = topic_model.visualize_documents(
    titles,
    reduced_embeddings=reduced_embeddings,
    width=1200,
    hide_annotations=True
)

# Update fonts of legend for easier visualization
fig.update_layout(font=dict(size=16))
```

```
↗
```

Documents and Topics



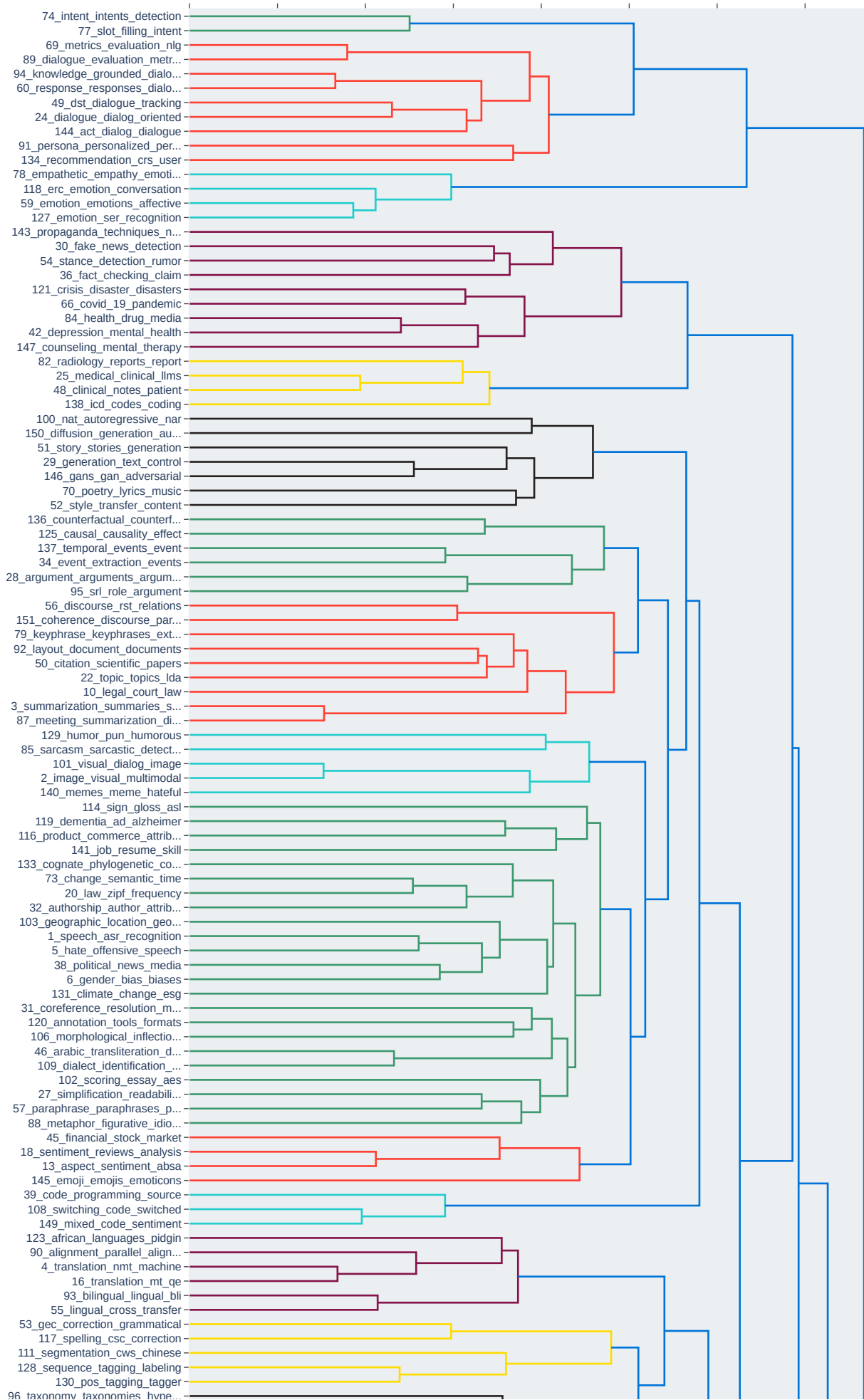
```
#visualize barchart with ranked keywords
# Visualize barchart with ranked keywords
topic_model.visualize_barchart()

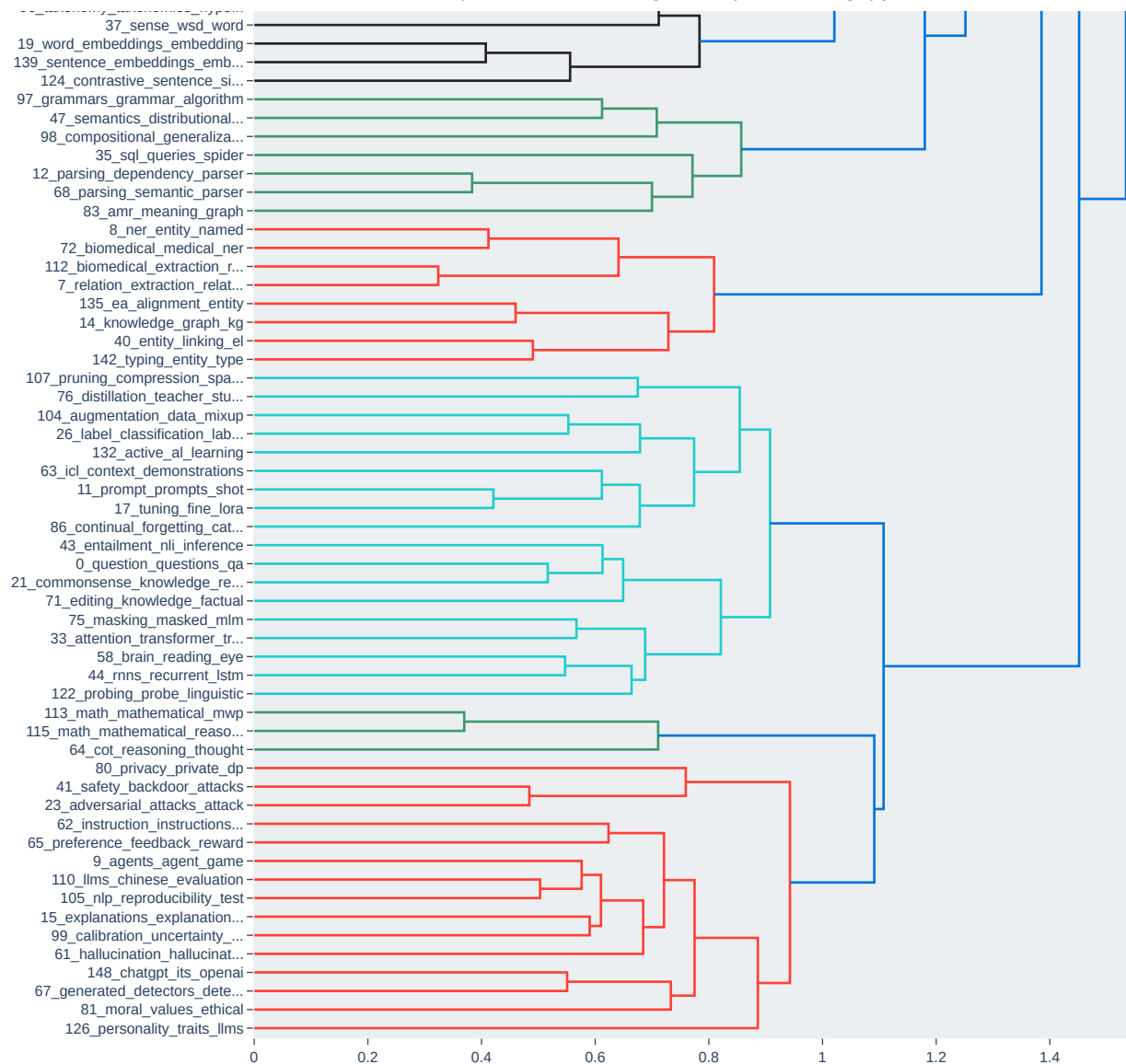
# Visualize relationships between topics
topic_model.visualize_heatmap(n_clusters=30)

# Visualize the potential hierarchical structure of topics
topic_model.visualize_hierarchy()
```



Hierarchical Clustering





✓ Representation Models

Double-click (or enter) to edit

```
from bertopic.representation import KeyBERTInspired
from bertopic import BERTopic

# Create your representation model
representation_model = KeyBERTInspired()

# Use the representation model in BERTopic on top of the default pipeline
topic_model = BERTopic(representation_model=representation_model)

# Save original representations
from copy import deepcopy
original_topics = deepcopy(topic_model.topic_representations_)

def topic_differences(model, original_topics, nr_topics=5):
    """Show the differences in topic representations between two models """
    df = pd.DataFrame(columns=["Topic", "Original", "Updated"])
    for topic in range(nr_topics):

        # Extract top 5 words per topic per model
        og_words = " | ".join(list(zip(*original_topics[topic]))[0][:5])
        new_words = " | ".join(list(zip(*model.get_topic(topic)))[0][:5])
        df.loc[len(df)] = [topic, og_words, new_words]

    return df
```

✓ KeyBert Inspired

```
from bertopic.representation import KeyBERTInspired

# Update our topic representations using KeyBERTInspired
print(type(abstracts))
representation_model = KeyBERTInspired()
topic_model.update_topics(abstracts, representation_model=representation_model)
```