```
from bertopic import BERTopic
from sentence_transformers import SentenceTransformer
from sklearn.datasets import fetch_20newsgroups
```

```
data = fetch_20newsgroups(subset='train',
                          remove=("headers", "footers", "quotes"))
documents = data.data[:2000]   # limit to avoid RAM issues

print("Loaded documents:", len(documents))
```

```
Loaded documents: 2000
```

```
# Embedding model
embedding_model = SentenceTransformer("all-MiniLM-L6-v2")
```

```
modules.json:  100%                                        349/349  [00:00<00:00,  32.2kB/s]

config_sentence_transformers.json:  100%                   116/116  [00:00<00:00,  4.57kB/s]

README.md:        10.5k/?  [00:00<00:00,  660kB/s]

sentence_bert_config.json:  100%                           53.0/53.0  [00:00<00:00,  3.42kB/s]

config.json:  100%                                         612/612  [00:00<00:00,  61.4kB/s]

model.safetensors:  100%                                   90.9M/90.9M  [00:01<00:00,  113MB/s]

tokenizer_config.json:  100%                               350/350  [00:00<00:00,  9.49kB/s]

vocab.txt:        232k/?  [00:00<00:00,  2.53MB/s]

tokenizer.json:        466k/?  [00:00<00:00,  8.00MB/s]

special_tokens_map.json:  100%                             112/112  [00:00<00:00,  2.87kB/s]

config.json:  100%                                         190/190  [00:00<00:00,  2.54kB/s]
```

```
# BERTopic with simple config (lite mode)
topic_model = BERTopic(
    embedding_model=embedding_model,
    low_memory=True,
    verbose=True
)
```

```
# Fit-transform
topics, probs = topic_model.fit_transform(documents)
```

```
2025-11-23 08:02:39,826 - BERTopic - Embedding - Transforming documents to embeddings.
Batches:  100%                                       63/63  [02:57<00:00,   1.73it/s]
2025-11-23 08:05:37,735 - BERTopic - Embedding - Completed ✓
2025-11-23 08:05:37,737 - BERTopic - Dimensionality - Fitting the dimensionality reduction algorithm
2025-11-23 08:05:55,876 - BERTopic - Dimensionality - Completed ✓
2025-11-23 08:05:55,878 - BERTopic - Cluster - Start clustering the reduced embeddings
2025-11-23 08:05:55,955 - BERTopic - Cluster - Completed ✓
2025-11-23 08:05:55,971 - BERTopic - Representation - Fine-tuning topics using representation models.
2025-11-23 08:05:56,460 - BERTopic - Representation - Completed ✓
```

```
# Print top topic info
topic_info = topic_model.get_topic_info()
print(topic_info.head())
```

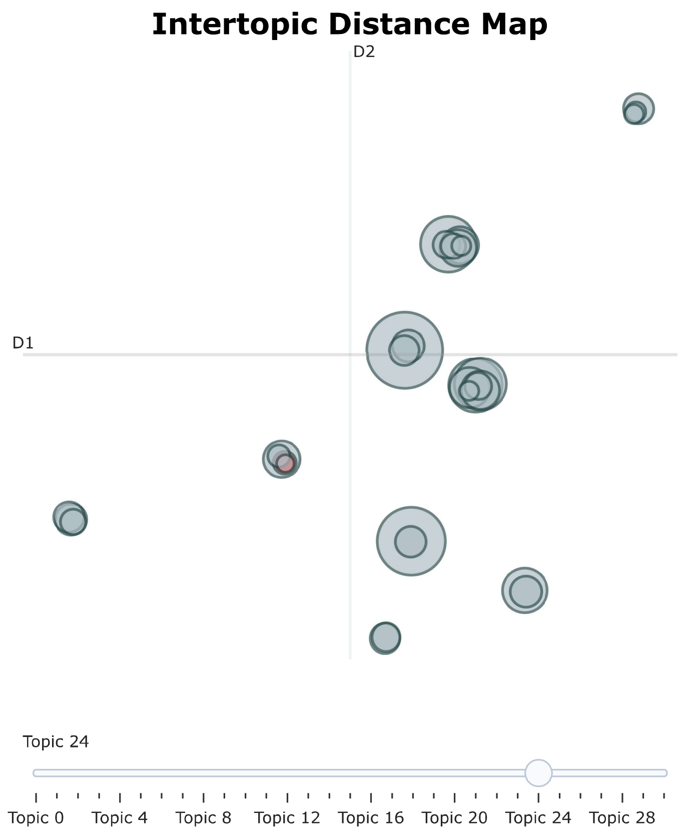
```
   Topic  Count               Name  \
0     -1    613   -1_ax_the_max_to
1      0    193     0_he_the_in_to
2      1    154   1_is_of_the_that
3      2    102  2_00_dos_for_good
4      3     94     3_the_to_of_it

                                     Representation  \
0       [ax, the, max, to, and, of, is, for, it, in]
1  [he, the, in, to, and, team, game, was, of, that]
2   [is, of, the, that, to, not, and, it, jesus, in]
3  [00, dos, for, good, 50, excellent, the, offer...
4  [the, to, of, it, be, that, is, clipper, in, and]

                                Representative_Docs
```

```
0  [THE WHITE HOUSE\n\n                   Office o...
1  [1992-93 Los Angeles Kings notes.\n------------...
2  [My last article included this quote:\n\n "If ...
3  [Discounts!  Please take\t$2.00 off each item ...
4  [After reading the debate over the Clipper, I ...
```

```
# Print example topic words
print("\nTopic 0 Keywords:")
print(topic_model.get_topic(0))
```

```
Topic 0 Keywords:
[('he', np.float64(0.026522111785216008)), ('the', np.float64(0.025063933391144429)), ('in', np.float64(0.020598082585780234)), (
```

```
# Basic visualization (works without heavy install)
fig = topic_model.visualize_topics()
fig.show()
```



```
Start coding or generate with AI.
```