Tweets_Canada_v0

September 8, 2023

[94]:
```
#!python -m pip install bertopic
#!python -m pip install git+https://github.com/jgmjgm/BERTopic.git@v0.14
```

1 Canada tweets topic model

Load the Canadian tweets and then determine the clusters. Each nation has their own customized notebook.

[1]:
```
# Read data
import pandas as pd
import numpy as np
np.random.seed(123)
import umap

from sklearn.feature_extraction.text import CountVectorizer
import tqdm as notebook_tqdm
import tqdm

COUNTRY2="CA"
##df = pd.read_csv("TopicSearch_2023-02-18/out_CA.csv")
df = pd.read_csv("../location/out_CA.csv")
df.head()
```

[1]:

	Unnamed: 0	created_at	entities	author_id
0	1	2021-12-30 09:50:03+00:00	{'hashtags': [{'start': 0, 'end': 4, 'tag': 'H…	164372140
1	11	2021-12-30 08:22:50+00:00	{'urls': [{'start': 238, 'end': 261, 'url': 'h…	3226565029
2	17	2021-12-30 08:04:15+00:00	{'annotations': [{'start': 27, 'end': 28, 'pro…	927436681879330816
3	23	2021-12-30 07:05:52+00:00	{'annotations': [{'start': 68, 'end': 73, 'pro…	312077424
4	26	2021-12-30 06:49:02+00:00	{'annotations': [{'start': 33, 'end': 46, 'pro…	1460683717920243712

```
                                  referenced_tweets      conversation_id  \
0                                               NaN  1476490761092972544
1                                               NaN  1476468809938657280
2                                               NaN  1476464131720896512
3  [{'type': 'quoted', 'id': '1476290688069386246'}]  1476449441888419840
4  [{'type': 'replied_to', 'id': '147643764116792…  1476437641167925248

             tweet_id  in_reply_to_user_id  text geo_id  …  \
0  1476490761092972544                  NaN   NaN    NaN  …
1  1476468809938657280                  NaN   NaN    NaN  …
2  1476464131720896512                  NaN   NaN    NaN  …
3  1476449441888419840                  NaN   NaN    NaN  …
4  1476445204420784128          468340229.0   NaN    NaN  …

         username            profile_created                 profile_loc  \
0      standard3d  2010-07-08 18:26:50+00:00             Toronto, Ontario
1         UrbanVN  2015-05-25 19:57:06+00:00  Vancouver, British Columbia
2  Brendon_Delorme  2017-11-06 07:25:10+00:00             Calgary, Alberta
3         robdoel  2011-06-06 15:04:06+00:00             Airdrie, Alberta
4         TeKBC19  2021-11-16 18:58:47+00:00       Hope, British Columbia

                     url followers following total_tweets  \
0  https://t.co/TLkvrJtkJU    2122.0    5003.0     138398.0
1   http://t.co/2gGZZfH2zg     217.0      48.0      77564.0
2                      NaN      37.0      91.0        426.0
3                      NaN    3974.0    4941.0      75033.0
4                      NaN       2.0      23.0         40.0

                                    cleaned_text country_cat  \
0  #HTC Vive now supported #samsung #moto #huawei…          CA
1  IoT in smart cities Market Huge Growth Opportu…          CA
2  A Canadian was arrested in HK. Will the CCP us…          CA
3  A leader, would immediately boycott the Olympi…          CA
4  @Huawei_Canada Why can't I buy a Matebook X Pr…          CA

                 date
0  2021-12-30 09:50:03
1  2021-12-30 08:22:50
2  2021-12-30 08:04:15
3  2021-12-30 07:05:52
4  2021-12-30 06:49:02

[5 rows x 33 columns]
```

2 Load data

Create a column called week that we'll use to order the tweets

[2]:
```
from datetime import datetime
from datetime import date
from dateutil.relativedelta import relativedelta

df['date'] = df['created_at'].apply( lambda x: datetime.strptime(x.replace('+00:
 ↪00',''), '%Y-%m-%d %H:%M:%S') )
df["week"] = df['date'].dt.to_period('W').dt.start_time
df = df.sort_values(['week', 'followers'], ascending=[True, False])

# Make lists: weekly timestamp and text
timestamps = df["week"].values #.tolist()
ts = df["created_at"].values #.tolist()
timestamps = [str(t)[0:10] for t in timestamps]
text = df["cleaned_text"].values.tolist()
```

3 TOPIC MODEL

BERTopic model from https://maartengr.github.io/BERTopic/index.html

[3]:
```
# First run of the model
import hdbscan
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sentence_transformers import SentenceTransformer
import umap
from matplotlib.pyplot import figure

from bertopic import BERTopic
from bertopic.representation import KeyBERTInspired
from bertopic.representation import MaximalMarginalRelevance

#representation_model = KeyBERTInspired()
representation_model = MaximalMarginalRelevance(diversity=0.3) # WIP by Maarten.
 ↪ I don't see any real effect
sentence_model = SentenceTransformer('all-MiniLM-L6-v2')
#clusterer1 = hdbscan.HDBSCAN( min_cluster_size=50, metric='euclidean',␣
 ↪cluster_selection_method="leaf", memory="memory1", min_samples=30,␣
 ↪gen_min_span_tree=True, prediction_data=True ) #, min_samples=30 )
clusterer1 = hdbscan.HDBSCAN( min_cluster_size=80, metric='euclidean',␣
 ↪cluster_selection_method="leaf", memory="memory1", min_samples=30,␣
 ↪gen_min_span_tree=True, prediction_data=True ) #, min_samples=30 )

umodel1 = umap.UMAP(n_neighbors=40, n_components=5, min_dist=0.
 ↪0,metric='cosine', low_memory=True, random_state=42) #nn was 15
```

```
topic_model1 = BERTopic(
    hdbscan_model = clusterer1,
    umap_model= umodel1,
    language='english',
    calculate_probabilities=True,
    verbose=True,
    #n_gram_range=(1, 3),
    embedding_model=sentence_model,
    representation_model=representation_model,
    #nr_topics = 4
)
```

4 Run the topic model

[4]:
```
%%time
topics1, probs1 = topic_model1.fit_transform(text)
df["topic"] = topics1 # Save results back to the dataframe
```

```
Batches:   0%|          | 0/296 [00:00<?, ?it/s]

2023-09-08 20:57:27,735 - BERTopic - Transformed documents to Embeddings
2023-09-08 20:58:35,172 - BERTopic - Reduced dimensionality
2023-09-08 20:58:35,993 - BERTopic - Clustered reduced embeddings

CPU times: total: 20min 25s
Wall time: 4min 17s
```

[5]:
```
topic_model1.save(f"BERTopic_model_{COUNTRY2}.model")
```

5 Update labels

Update the topic labels to be easier to read. This is accessed as a custom label.

[6]:
```
import nltk
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
stop_words.add("https")
stop_words.add("000")

topic_labels = topic_model1.generate_topic_labels(nr_words=15,
                                                  topic_prefix=True,
                                                  word_length=15,
                                                  separator=", ")
#print( topic_labels)
# Clean up labels - remove stopwords
all_labels = []
for lab in topic_labels:
```

```
    line = []
    for word in lab.split(", "):
        if word not in stop_words and 'https' not in word:
            line.append(word)
    all_labels.append(", ".join(line))
all_labels

topic_model1.set_topic_labels(all_labels)
```

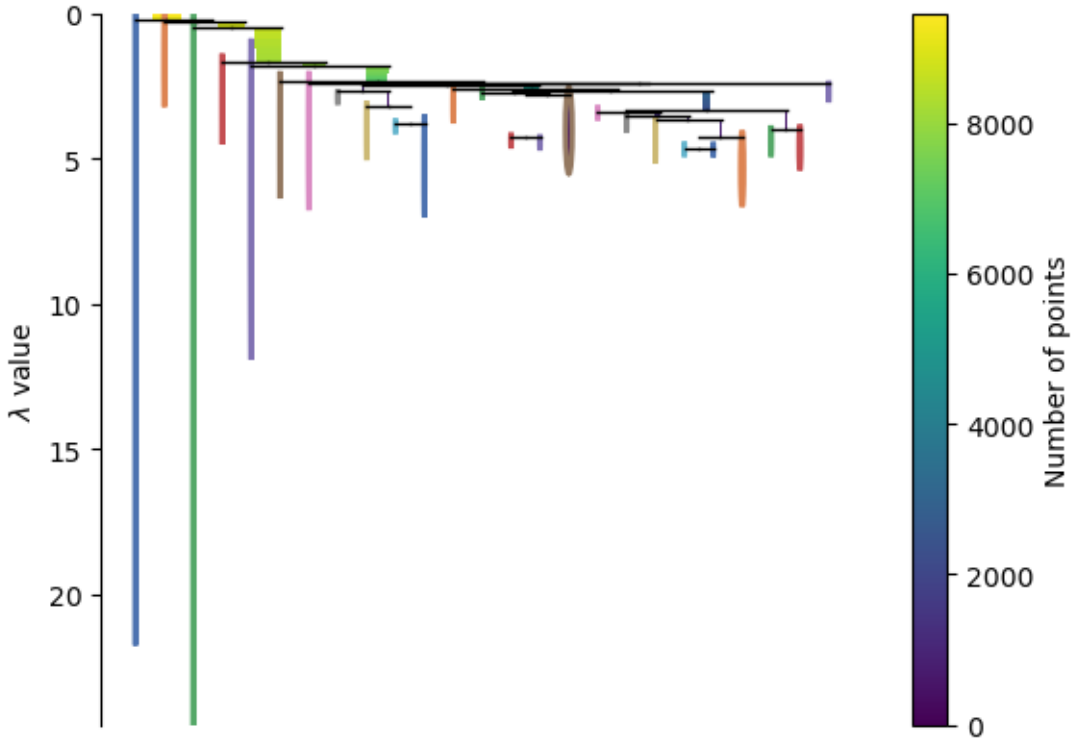
6 Examine clusters

Basic look at the grouping of the clusters. Really need to examine over time.

Also show the hierarchy of clusters. Get a sense of whether we should combine topics.

[7]:
```
from matplotlib.pyplot import figure
###figure(figsize=(8, 6), dpi=320)
clusterer1 = topic_model1.hdbscan_model
clusterer1.condensed_tree_.plot(select_clusters=True, selection_palette=sns.
 ↪color_palette('deep', 50))
```

[7]:
```
<AxesSubplot:ylabel='$\\lambda$ value'>
```



[8]:
```
topic_model1.visualize_hierarchy(custom_labels=True, color_threshold=1.5 ) #,␣
↪topics=keep)
```

7 Create a shortlist of relevant topics

Not all topics are relevant. Some are ads for Huawei phones, sales, etc.

[9]:
```
ignore_topics = [-1] #[9,8,18,21,19,22,-1] # leave out -1 for now! leaf
keep = set(topic_model1.topics_) - set(ignore_topics)

print(keep)
```

```
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24}
```

8 Write out CSV of topics

[10]:
```
# Create a dict with the topic as key and label as val
named = {}
for line in all_labels:
    words = line.split(", ")
    topic = words[0]
    named[ int(topic) ] = line.replace(', ',' ').strip()

print( named)
```

```
{-1: '-1 huawei co china canada', 0: '0 canada trudeau huawei china
justintrudeau decision canadian ban', 1: '1 huawei jeancharest_ phone thanks
charest', 2: '2 quebec liberal cpc jean huawei leader premier meng wanzhou', 3:
'3 market cities iot cisco growth ibm industrial intel forecast oracle', 4: '4
supported mate moto samsung android arch2o_magazine huawei gearvr autodesk 3d',
5: '5 liberal jeancharest_ huawei conservative pierrepoilievre pierrepoilievre
cpc_hq liberals vote', 6: '6 much jeancharest_ huawei jean pay pierrepoilievre
number month', 7: '7 ban huawei banning still uk', 8: '8 enforcement trudeau
childbase crimes covid kovrig warns putin 5g belt', 9: '9 bell telus equipment
compensation telecoms ban liberals huawei taxpayers', 10: '10 wd_black logitechg
asususa neweggcan huawei_canada projector logitech headset desk prizes', 11: '11
huawei security china chinese privacy sanctions track', 12: '12 5g canada
network huawei justintrudeau ban security trudeau', 13: '13 5g huawei co
ericsson network city', 14: '14 ties pals fed pushed media quebec liberals
conservative liberal', 15: '15 surveillance hack accused china spies huawei
uyghur telecom report', 16: '16 2021 patent patents huawei revenue apple
patentnews autonomous china alibaba', 17: '17 pierrepoilievre ctv_powerplay
evanlsolomon pierre chicomm huawei qdko2gbgvx article hh8wfasnsl', 18: '18
brands samsung usb cable charger lg sony lens fastcharger huawei', 19: '19
matepad tablet mateview matebook monitor laptop huawei gb notebook macbook', 20:
'20 smartwatch fitness tracker giveaway battery huawei smartwatches fitbit strap
```

```
garmin', 21: '21 huawei latest via youtube wifi connecting yotpo android
hyperloop', 22: '22 months huawei announce editorial cdnpoli alanfryermedia
overdue promised would', 23: '23 camera p30 pro phone pixel huawei p20 google
new cameras', 24: '24 ambassador warns 5g ploy signal china says banning
huawei'}
```

[11]:
```
# Count tweets in each topic
topic_numbers = set(df.topic.values.tolist())
sm1 = df.groupby(["topic"]).count().reset_index()
sm1 = sm1.iloc[:,[0,1]]
sm1.columns = ["topic","count"]
countd = dict( zip(sm1.topic.values, sm1["count"].values) )
```

9 Write out topics to be analyzed in more depth

[127]:
```
smaller = df[ df['topic'].isin( keep ) ]
smaller["count"] = smaller["topic"].apply( lambda x: countd.get(x,"?") )
#smaller = df.copy()
smaller["name"]  = smaller["topic"].apply( lambda x: named.get(x,"UNKNOWN") )
smaller.to_csv("topics_CA.csv")
smaller.head()
```

[127]:

	Unnamed: 0	created_at	entities	author_id	referenced_tweets	conversation_id	tweet_id	in_reply_to_user_id	text	geo_id	…
943	67976	2021-11-21 16:06:14+00:00	{'urls': [{'start': 57, 'end': 80, 'url': 'htt…	24700876	NaN	1462452302296592384	1462452302296592384	NaN	NaN	NaN	…
973	70303	2021-11-21 04:09:08+00:00	{'urls': [{'start': 48, 'end': 71, 'url': 'htt…	24700876	NaN	1462271836931211264	1462271836931211264	NaN	NaN	NaN	…
1011	71439	2021-11-20 19:48:10+00:00	{'annotations': [{'start': 8, 'end': 13, 'prob…	16753407	[{'type': 'replied_to', 'id': '146214550165514…	1461681969855700992	1462145763870486528	16753407.0	NaN	5d058f2e9fe1516c	…
936	67957	2021-11-21 17:50:11+00:00	{'mentions': [{'start': 25, 'end': 39, 'userna…	23865085	NaN	1462478461222719488					
969	70296	2021-11-21 04:38:02+00:00	{'urls': [{'start': 48, 'end': 71, 'url': 'htt…	27646245	NaN	1462279111120986112					

```
936   1462478461222719488                  NaN   NaN               NaN  …
969   1462279111120986112                  NaN   NaN               NaN  …

                          url  followers  following  total_tweets  \
943   https://t.co/syvPNjzI8W   385006.0      594.0      271514.0
973   https://t.co/syvPNjzI8W   385006.0      594.0      271514.0
1011  https://t.co/ptKr36fetU   137908.0      991.0      115187.0
936   https://t.co/pwAbpkhlmq   129943.0    10215.0       41105.0
969    http://t.co/3mOQNo2wu8   104840.0      878.0      200520.0

                                        cleaned_text country_cat  \
943   Sunday's editorial: PM's decision on Huawei lo…          CA
973   EDITORIAL: PM's decision on Huawei long overdu…          CA
1011  "I urge Canada to resolve this Huawei issue," …          CA
936   For the last four years, @JustinTrudeau has be…          CA
969   EDITORIAL: PM's decision on Huawei long overdu…          CA

                    date       week topic count
943  2021-11-21 16:06:14 2021-11-15    22    82
973  2021-11-21 04:09:08 2021-11-15    22    82
1011 2021-11-20 19:48:10 2021-11-15     0  1126
936  2021-11-21 17:50:11 2021-11-15    12   106
969  2021-11-21 04:38:02 2021-11-15    22    82

[5 rows x 36 columns]
```

10 Examine the topics and tweets

[12]:
```
from IPython.display import display, HTML
#df = df.sort_values(['topics',"week","followers"], ascending=[True,True,False])
s = df[df.topic==0][["followers","week","cleaned_text"]].copy()
display(HTML(s.to_html()))
```

```
<IPython.core.display.HTML object>
```

11 Run dynamic model

[14]:
```
%%time
topics_over_time = topic_model1.topics_over_time(text, timestamps, nr_bins=24,␣
 ↪datetime_format="%Y-%m-%d")
```

```
24it [01:07,  2.83s/it]

CPU times: total: 10min 25s
Wall time: 1min 8s
```

[15]:
```
topics = set( topic_model1.topics_ )
topics.remove(-1)
#topics.remove(0)
print( topics)
topic_model1.visualize_topics_over_time(topics_over_time, topics=keep ) #,␣
 ↪topics=lst, top_n_topics=20)
```

```
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24}
```

[16]:
```
fig = topic_model1.visualize_barchart(n_words=10)
fig
```

12 Visualizations

[17]:
```
# Create lst that contains the outliers (-1) so it isn't displayed

topic_model1.visualize_topics(topics=keep)
```

12.1 Print out list of topics and count the number in each cluster

[18]:
```
df.rename(columns={"topics":"topic"}, inplace=True)
#print( df.columns)

topic_numbers = set(df.topic.values.tolist())
sm1 = df.groupby(["topic"]).count().reset_index()
sm1 = sm1.iloc[:,[0,1]]
sm1.columns = ["topic","count"]
print( sm1.topic.values )
d = dict( zip(sm1.topic.values, sm1["count"].values) )
sm1 = sm1[sm1["count"]>0]

sm1
```

```
[-1  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
 23 24]
```

[18]:

	topic	count
0	-1	4835
1	0	1126
2	1	523
3	2	317
4	3	203
5	4	184
6	5	184
7	6	175
8	7	158

```
9        8    141
10       9    137
11      10    135
12      11    119
13      12    106
14      13    106
15      14    105
16      15    104
17      16    104
18      17    104
19      18     98
20      19     90
21      20     89
22      21     85
23      22     82
24      23     82
25      24     80
```

[19]:
```
# Shows a topic document cloud. Takes a long time to run! Not very informative.
lst = keep
print(f"Examine topics:{lst}")
ht = topic_model1.hierarchical_topics(text)
print("1")
fig = topic_model1.visualize_hierarchical_documents(text,ht,custom_labels=True,␣
 ↪topics=lst, hide_document_hover=False)
print("2")
fig.write_html("hier_docs.html")
print("3")

fig
```

```
Examine topics:{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 20, 21, 22, 23, 24}
```

```
100%|
    | 24/24 [00:01<00:00, 12.92it/s]
```

```
1
2
3
```

13 Reduce outliers

Source: https://maartengr.github.io/BERTopic/getting_started/outlier_reduction/outlier_reduction.html#visua
topics-over-time

This is an experiment. Connect the outlying points to the nearest clusters. When we drop the irrelevant topics, many of our outliers will be included.

[20]:
```
new_topics1 = topic_model1.reduce_outliers( text, topics1, probabilities=probs1␣
↪)
df["new_topics"] = new_topics1
```

```
100%|
     | 5/5 [00:01<00:00,  3.69it/s]
```

[21]:
```
# Look at the results by counting across topic groups
topic_numbers = set(df.topics.values.tolist())
sm1 = df.groupby(["new_topics"]).count().reset_index()
sm1 = sm1.iloc[:,[0,1]]
sm1.columns = ["new_topics","count"]
sm1
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[21], line 2
      1 # Look at the results by counting across topic groups
----> 2 topic_numbers = set(df.topics.values.tolist())
      3 sm1 = df.groupby(["new_topics"]).count().reset_index()
      4 sm1 = sm1.iloc[:,[0,1]]

File ~\anaconda3\envs\BERTopic\lib\site-packages\pandas\core\generic.py:5902, i␣
↪NDFrame.__getattr__(self, name)
   5895 if (
   5896     name not in self._internal_names_set
   5897     and name not in self._metadata
   5898     and name not in self._accessors
   5899     and self._info_axis._can_hold_identifiers_and_holds_name(name)
   5900 ):
   5901     return self[name]
-> 5902 return object.__getattribute__(self, name)

AttributeError: 'DataFrame' object has no attribute 'topics'
```

[ ]:
```
# Update the topic labels. these are stored as a custom label
#from sklearn.feature_extraction.text import CountVectorizer
#vectorizer_model = CountVectorizer(ngram_range=(1, 3), stop_words="english")
#topic_model1.update_topics(text2, vectorizer_model=vectorizer_model)
```

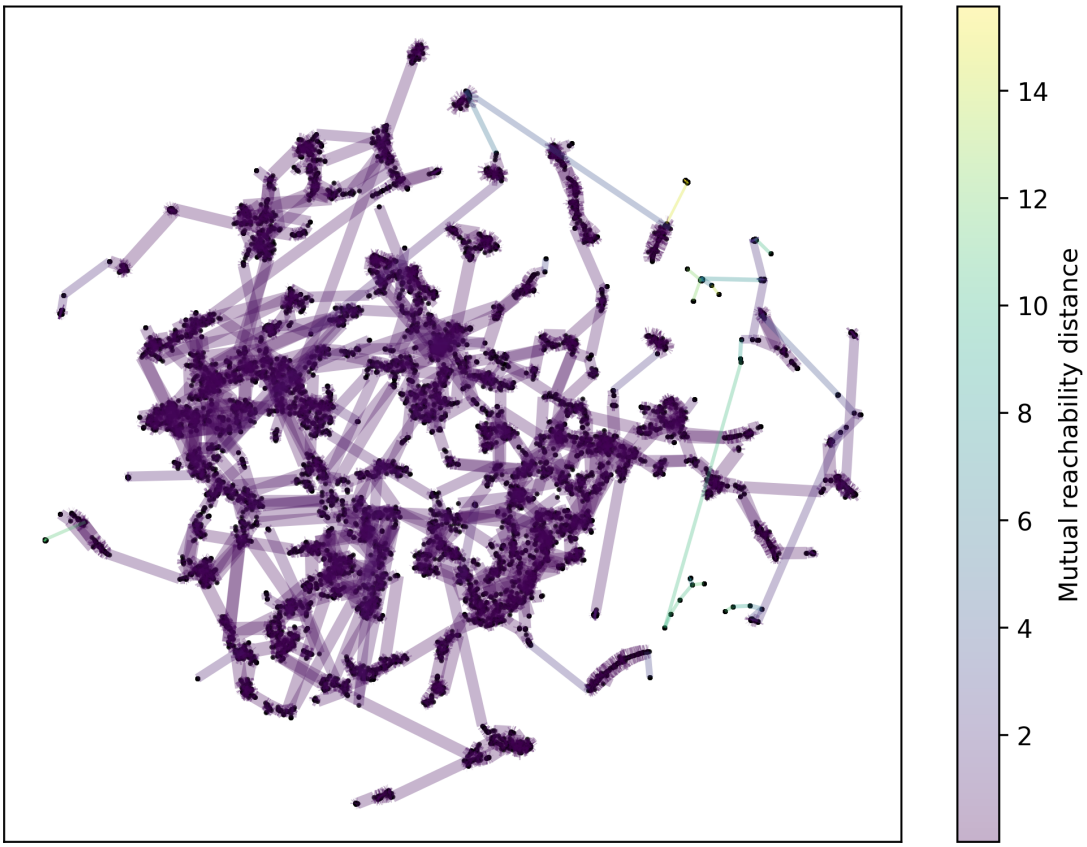
[22]:
```
topic_model1.visualize_hierarchy( custom_labels=True, color_threshold=1.5 ) #,␣
↪topics=lst)
```

[23]:
```
from matplotlib.pyplot import figure

figure(figsize=(8, 6), dpi=320)
```

```
clusterer1.minimum_spanning_tree_.plot(edge_cmap='viridis',
                                       edge_alpha=0.3,
                                       node_size=1,
                                       edge_linewidth=1)
```

[23]: <AxesSubplot:>



[24]:
```
fig = topic_model1.visualize_barchart(n_words=15)
fig.write_html("bar_chart.html")
fig
```

[25]:
```
fig = topic_model1.visualize_documents(text)
fig.write_html("topics.html")
fig
```

[27]:
```
topic_model1.visualize_topics()
```

[28]:
```
# Visualization tips
# https://maartengr.github.io/BERTopic/getting_started/visualization/
  ↪visualization.html#quick-start
```

```
hierarchical_topics = topic_model1.hierarchical_topics(text)
fig = topic_model1.visualize_hierarchy(hierarchical_topics=hierarchical_topics,␣
 ↪custom_labels=True)
fig.write_html("hier_cluster.html")
fig
```

```
100%|
    | 24/24 [00:01<00:00, 13.70it/s]
```

[29]:
```
topic_model1.visualize_heatmap( n_clusters=10)
```

[30]:
```
topic_model1.visualize_term_rank(topics=keep)
```

[ ]:
```

```

[ ]:
```
all_labels

"""
# Canada - topics. After analysis
Group1 = [0,2,5,6,7,8,9,11,12,13,14,15,17,22,24] # Huawei as security threat
Group2 = [1,3,4,10,16,18,19,20,21,23,] # Huawei as a legitimate company. i.e.␣
 ↪normal PR
words1 = []
words2 = []
d = { int(l[0]):l[1:] for l in [ line.split(', ') for line in all_labels ] }
#for k,lst in d.items():
for k in Group1:
    words1.extend( d[k] )
for k in Group2:
    words2.extend( d[k] )
list( set(words2) )
"""
```