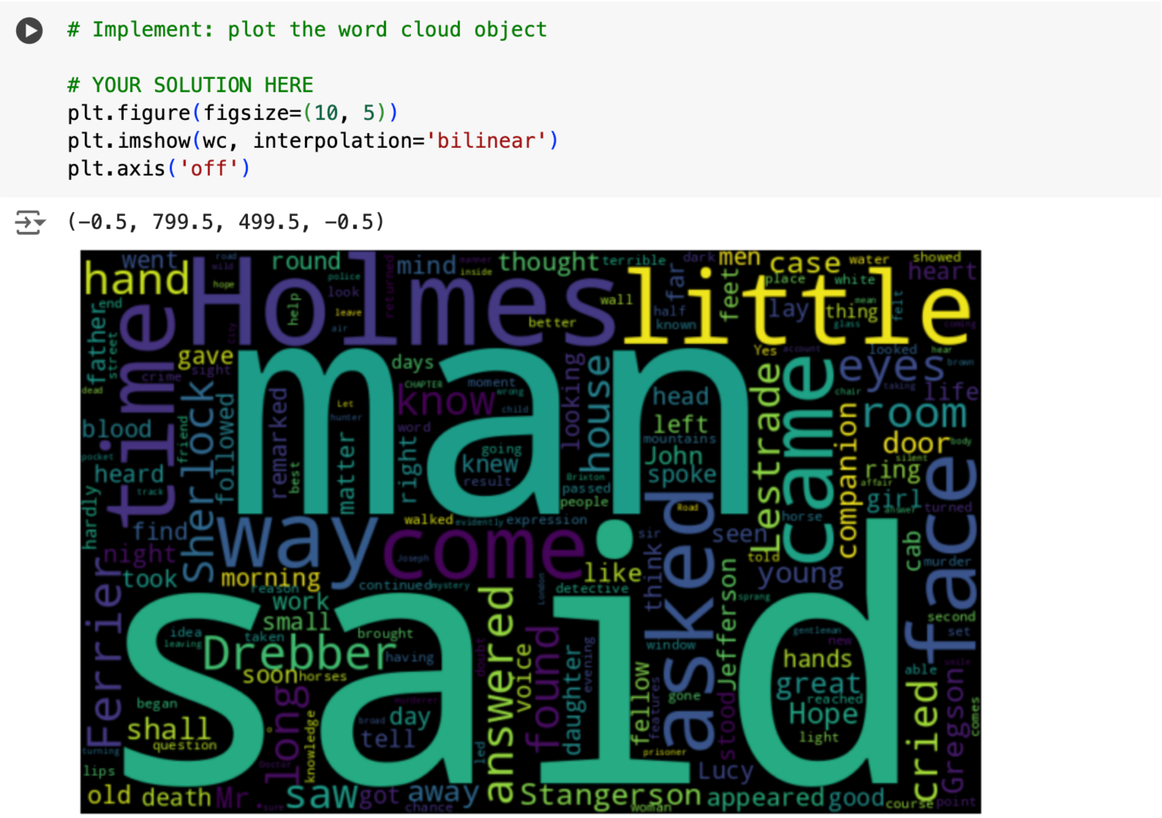
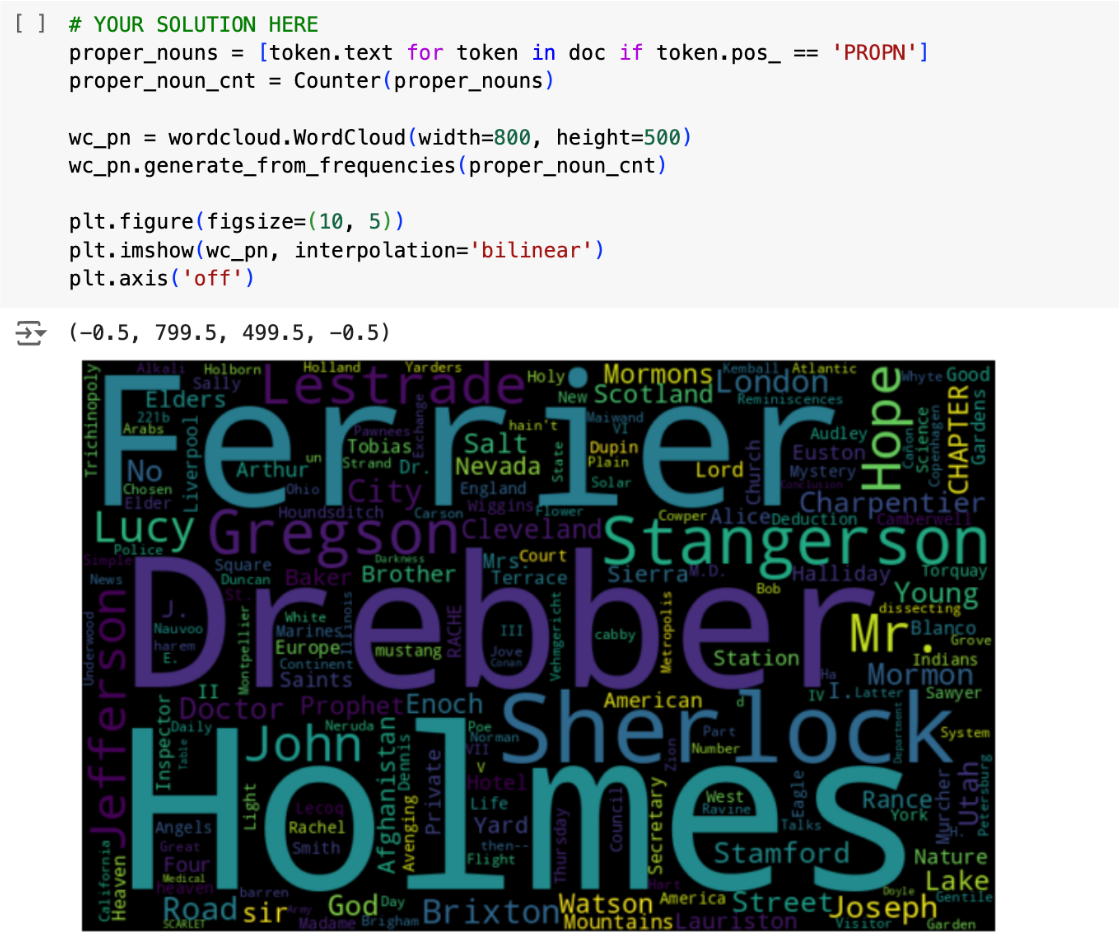
Plots for Lab13_Chede_Neha



```
# Implement: plot the word cloud object

# YOUR SOLUTION HERE
plt.figure(figsize=(10, 5))
plt.imshow(wc, interpolation='bilinear')
plt.axis('off')
```

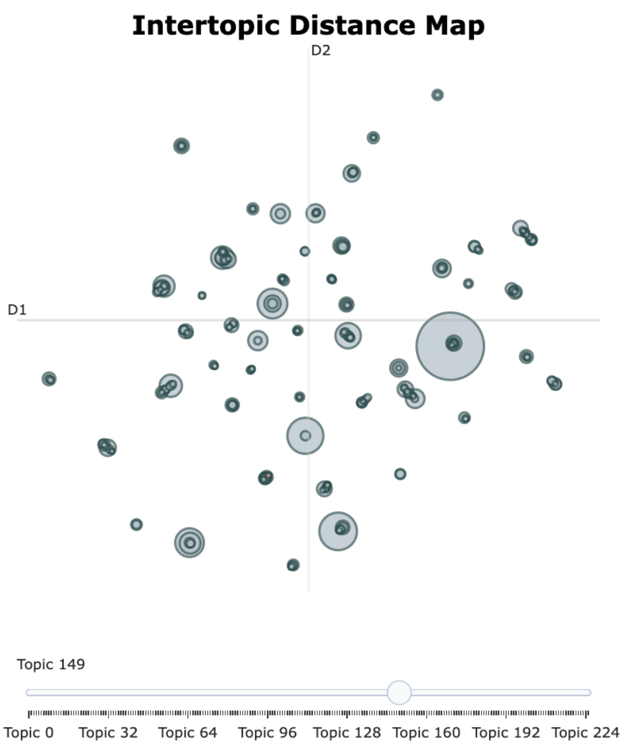
```
(-0.5, 799.5, 499.5, -0.5)
```



```
# YOUR SOLUTION HERE
proper_nouns = [token.text for token in doc if token.pos_ == 'PROPN']
proper_noun_cnt = Counter(proper_nouns)

wc_pn = wordcloud.WordCloud(width=800, height=500)
wc_pn.generate_from_frequencies(proper_noun_cnt)

plt.figure(figsize=(10, 5))
plt.imshow(wc_pn, interpolation='bilinear')
plt.axis('off')
```

```
(-0.5, 799.5, 499.5, -0.5)
```

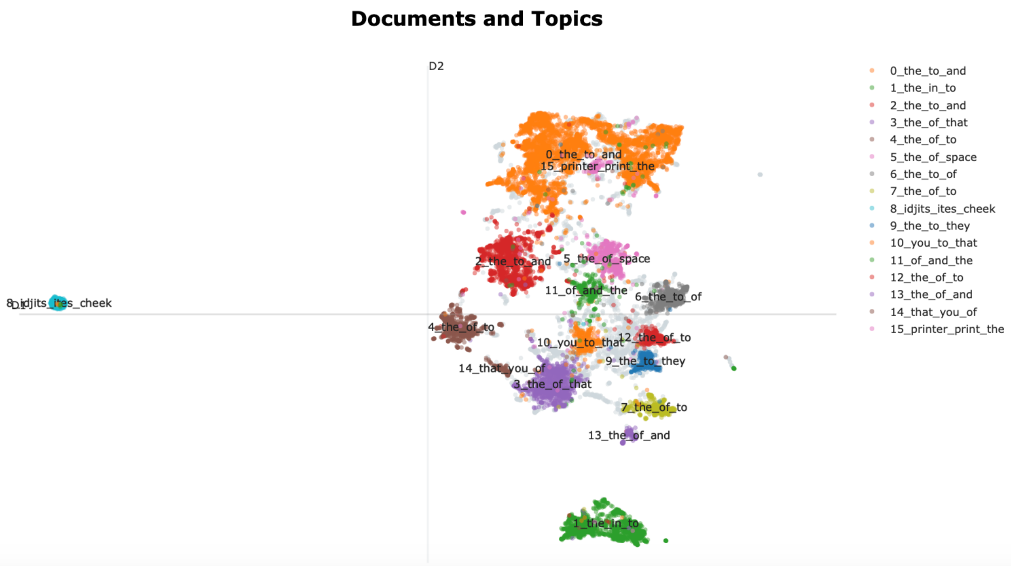
```
[ ] topic_model.visualize_topics()
```



Another visualization we can do (see https://maartengr.github.io/BERTopic/getting_started/visualization/visualization.html#visualize-documents) is to use the sentence embedding model to embed the documents into a vector space and visualize all the documents using the UMAP algorithm, along with the topics.

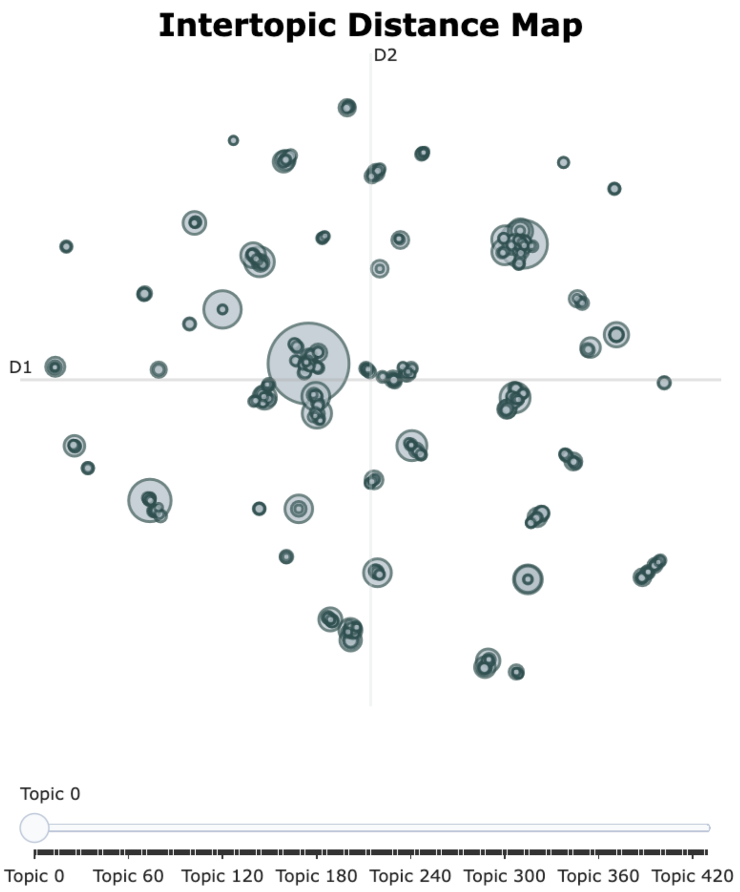
This time, we will set the `min_topic_size` to 100 to reduce the number of topics it finds and to make the visualization simpler.

```
[ ] from sentence_transformers import SentenceTransformer
    from umap import UMAP

    # Prepare embeddings
    sentence_model = SentenceTransformer("all-MiniLM-L6-v2")
    embeddings = sentence_model.encode(docs, show_progress_bar=False)

    topic_model = BERTopic(min_topic_size=100).fit(docs, embeddings)

    # Run the visualization with the original embeddings
    topic_model.visualize_documents(docs, embeddings=embeddings)

    # Reduce dimensionality of embeddings, this step is optional but much faster to perform iteratively:
    reduced_embeddings = UMAP(n_neighbors=10, n_components=2, min_dist=0.0, metric='cosine').fit_transform(embeddings)
    topic_model.visualize_documents(docs, reduced_embeddings=reduced_embeddings)
```

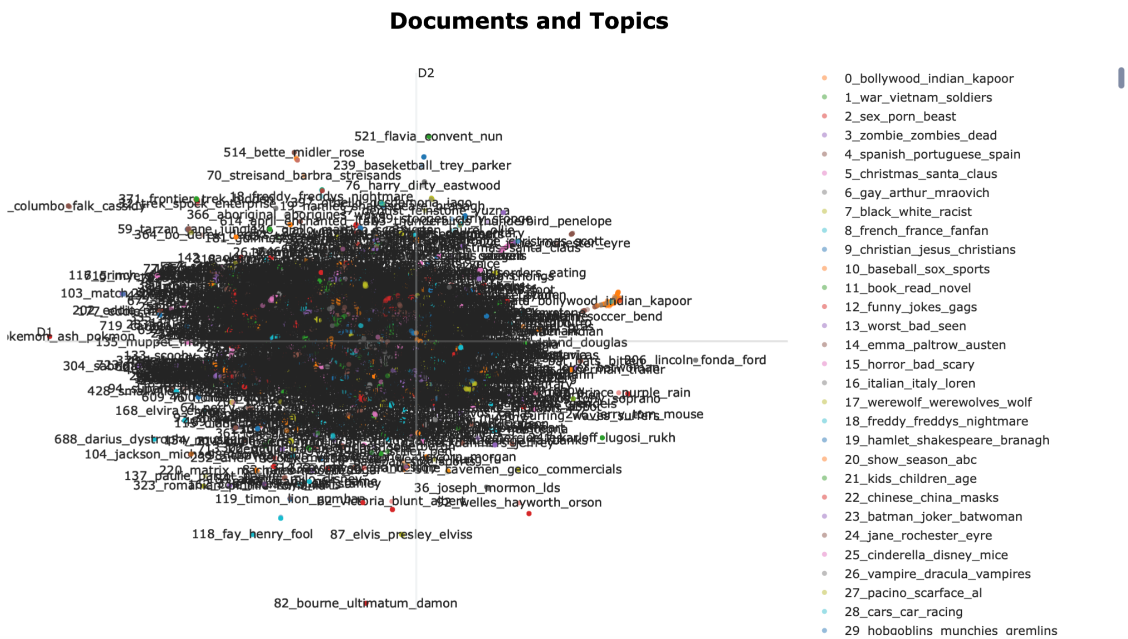


```
25000
```

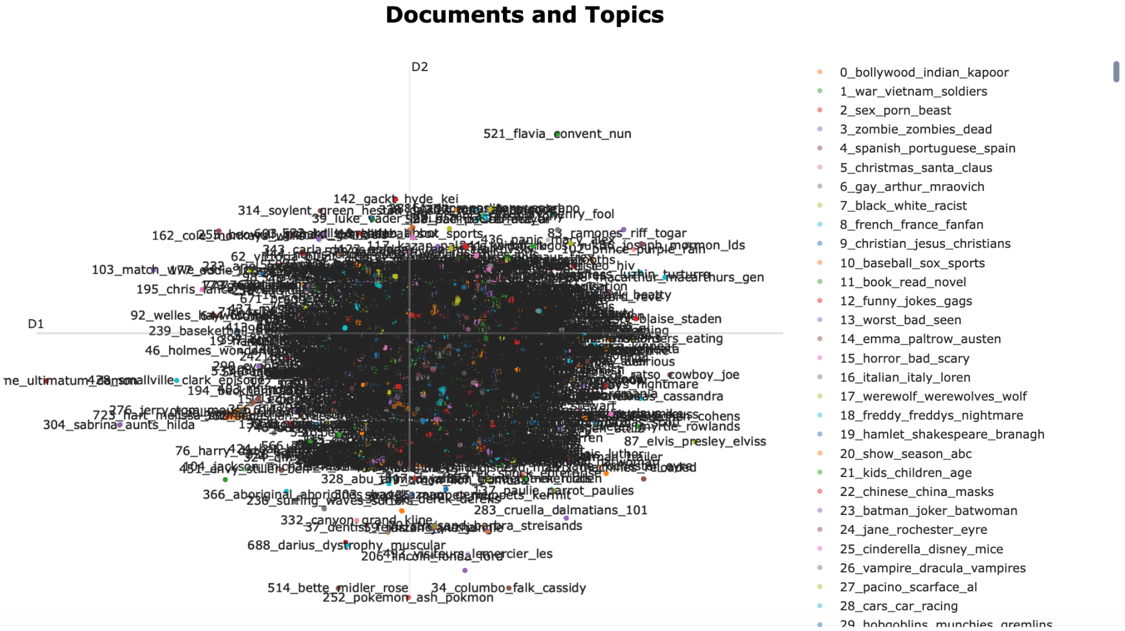
```
topic_model = BERTopic()
topics, probs = topic_model.fit_transform(docs)

topic_model.visualize_topics()
```



```
topic_model.visualize_documents(docs, embeddings=embeddings)
```



```
reduced_embeddings = UMAP(n_neighbors=10, n_components=2, min_dist=0.0, metric='cosine').fit_transform(embeddings)
topic_model.visualize_documents(docs, reduced_embeddings=reduced_embeddings)
```


Documents and Topics
D2
D1
521_flavia_convent_nun
142_gackt_hyde_kei
314_soylent_green_heston
39_luke_vader
83_ramones_riff_togar
ph_mormon_lds
162_co
103_match_
195_chris_
92_welles_
239_basketball
46_holmes_wonder
ne_ultimatum
smallville
clark
194_
723_hart_melissa
304_sabrina_aunts_hilda
76_harry
451_andy
366_aboriginal_aborig
236_surfing_waves
328_abu
blaise_staden
eating
ratso_cowboy_joe
cassandra
cohens
87_elvis_presley_elviss
137_paulie_parrot_paulies
283_cruella_dalmatians_101
332_canyon_grand_kline
37_dentist
barbra_streisands
688_darius_dystrophy_muscular
206_lincoln_fonda_ford
lemercier_les
514_bette_midler_rose
34_columbo_falk_cassidy
252_pokemon_ash_pokmon
0_bollywood_indian_kapoor
1_war_vietnam_soldiers
2_sex_porn_beast
3_zombie_zombies_dead
4_spanish_portuguese_spain
5_christmas_santa_claus
6_gay_arthur_mraovich
7_black_white_racist
8_french_france_fanfan
9_christian_jesus_christians
10_baseball_sox_sports
11_book_read_novel
12_funny_jokes_gags
13_worst_bad_seen
14_emma_paltrow_austen
15_horror_bad_scary
16_italian_italy_loren
17_werewolf_werewolves_wolf
18_freddy_freddys_nightmare
19_hamlet_shakespeare_branagh
20_show_season_abc
21_kids_children_age
22_chinese_china_masks
23_batman_joker_batwoman
24_jane_rochester_eyre
25_cinderella_disney_mice
26_vampire_dracula_vampires
27_pacino_scarface_al
28_cars_car_racing
29_hobgoblins_munchies_gremlins