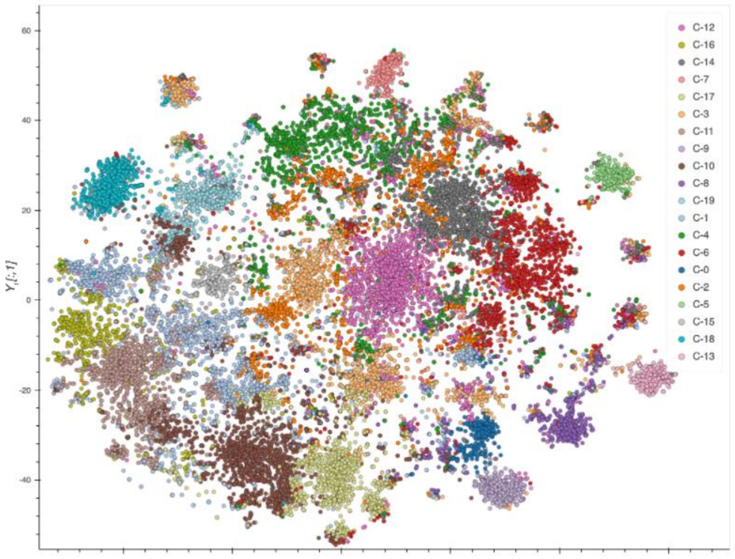
Some Description of the Clustering Literature

```
from IPython.display import Image
Image(url='https://raw.githubusercontent.com/MaksimEkin/COVID19-Literature-Clustering/master/cover/bokeh_plot.png', width=800, height=800)
```



```
from tqdm import tqdm
all_json_clean = list()
for index, entry in tqdm(enumerate(all_json), total=len(all_json)):
    try:
        content = FileReader(entry)
    except Exception as e:
        continue

    if len(content.body_text) == 0:
        continue
    all_json_clean.append(all_json[index])
all_json = all_json_clean
len(all_json)
```

```
100%|██████████| 401214/401214 [1:05:14<00:00, 102.50it/s]
401214
```

```
df_covid = pd.DataFrame(dict_, columns=['paper_id', 'doi', 'abstract', 'body_text',
df_covid.head()
```

```
100%|██████████| 15000/15000 [21:43<00:00, 11.51it/s]
```

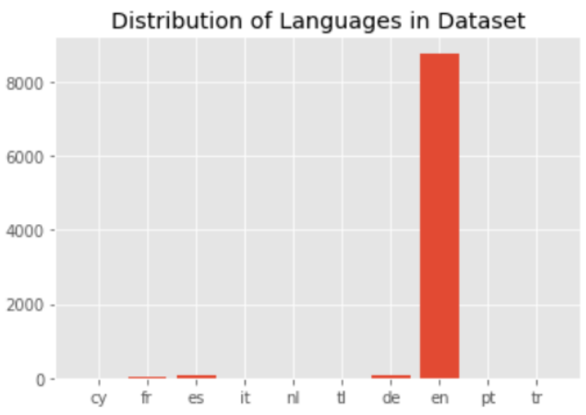
```
DetectorFactory.seed = 0
languages = []

for i in tqdm(range(0, len(df))):
    text = df.iloc[i]['body_text'].split(" ")
    lang = 'en'

    try:
        if len(text) > 50:
            lang = detect(" ".join(text[:50]))
        elif len(text) > 0:
            lang = detect(" ".join(text[:len(text)]))
    except Exception as e:
        all_words = set(text)
        try:
            lang = detect(" ".join(all_words))
        except Exception as e:
            try:
                lang = detect(df.iloc[i]['abstract_summary'])
            except Exception as e:
                lang = 'unknown'
                pass
    languages.append(lang)
```

```
100%|██████████| 8987/8987 [00:49<00:00, 179.75it/s]
```

```
df['language'] = languages
plt.bar(range(len(language_dict)), list(language_dict.values()), align='center')
plt.xticks(range(len(language_dict)), list(language_dict.keys()))
plt.title('Distribution of Languages in Dataset')
plt.show()
```



```
tqdm.pandas()
df['processed_text'] = df['body_text'].progress_apply(spacy_tokenizer)
```

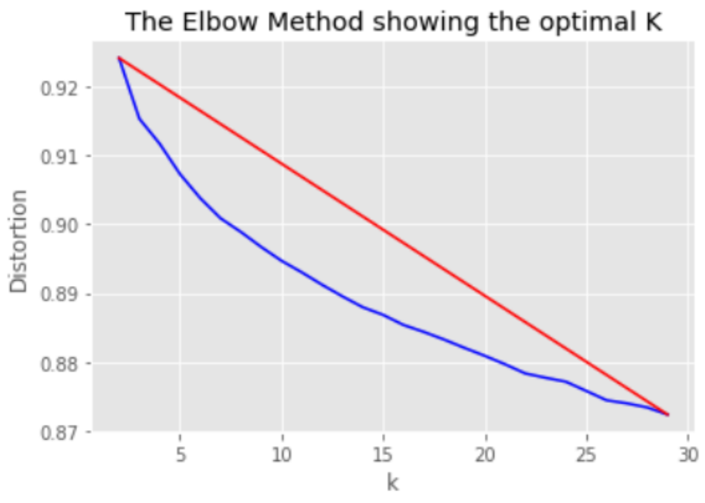
```
100%|██████████| 8781/8781 [43:27<00:00,  3.37it/s]
```

```
pca = PCA(n_components=0.95, random_state=42)
X_reduced = pca.fit_transform(X.toarray())
X_reduced.shape
```

```
(8781, 2295)
```

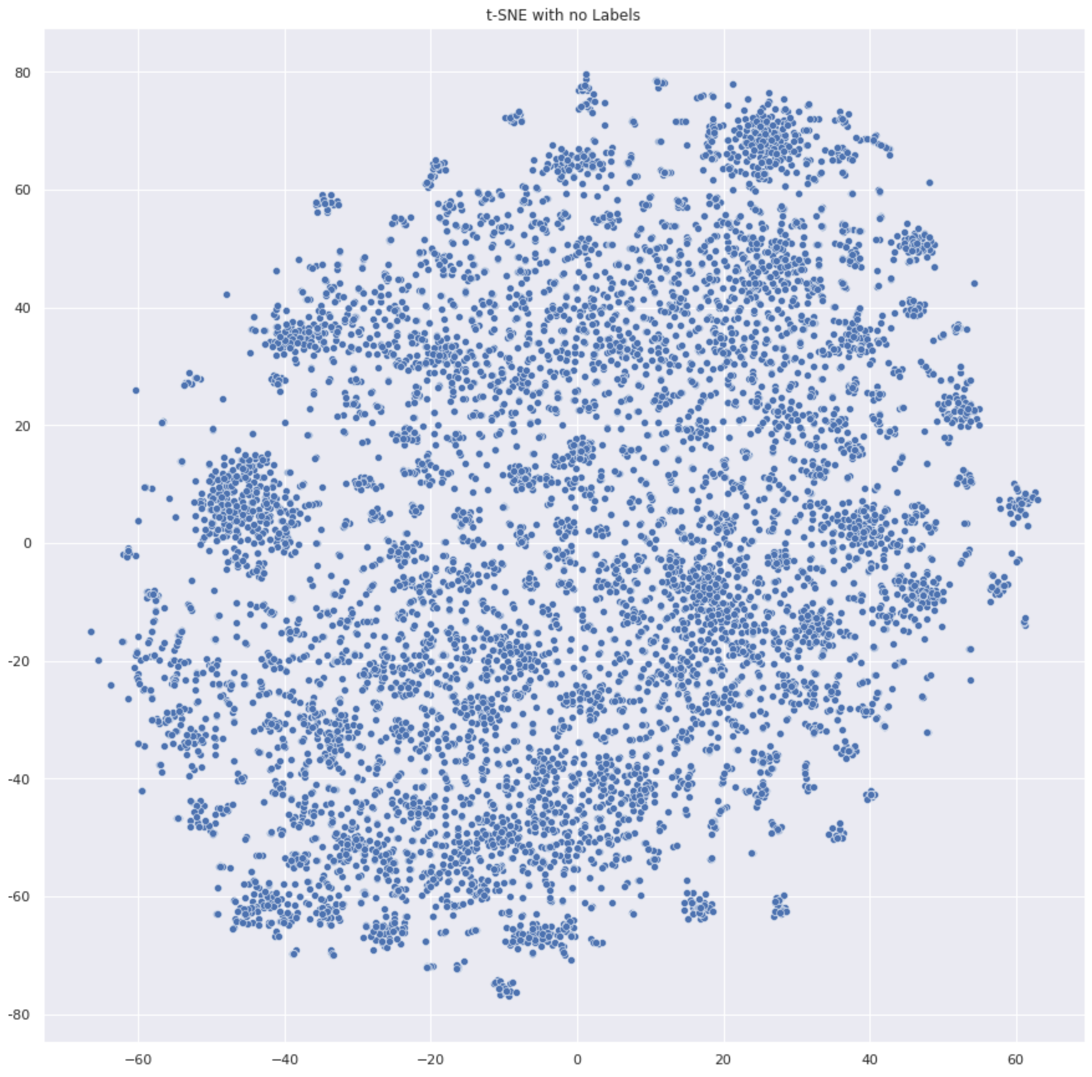
```
X_line = [K[0], K[-1]]
Y_line = [distortions[0], distortions[-1]]

plt.plot(K, distortions, 'b-')
plt.plot(X_line, Y_line, 'r')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal K')
plt.show()
```



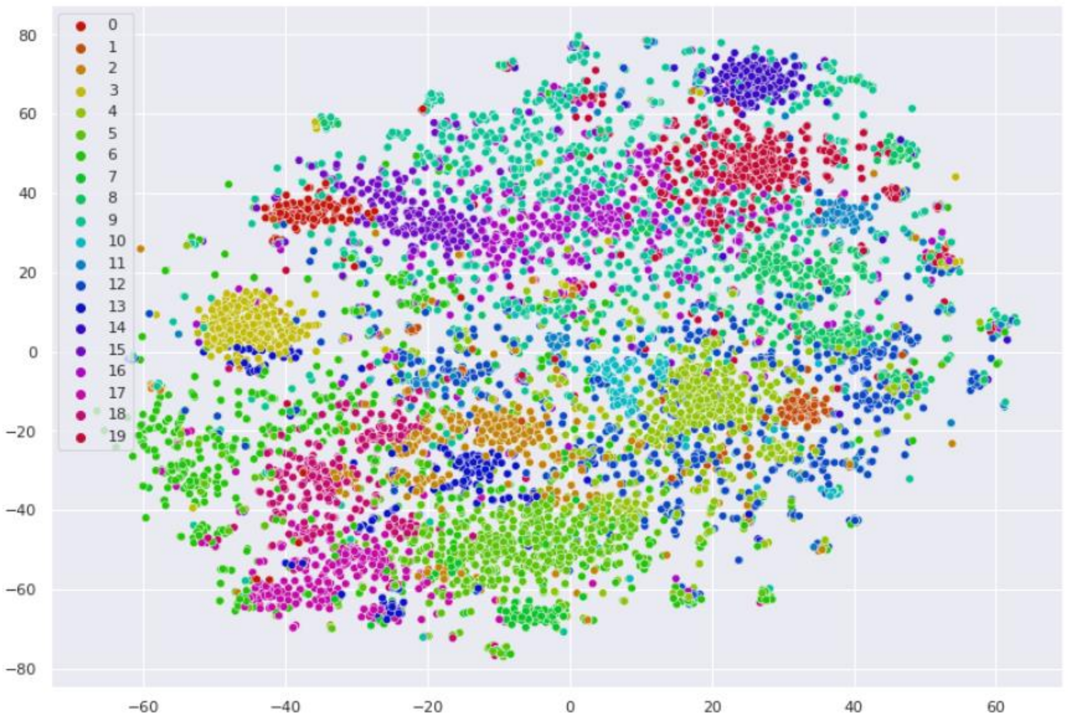
```
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(rc={'figure.figsize':(15, 15)})
palette = sns.color_palette('bright', 1)
sns.scatterplot(X_embedded[:,0], X_embedded[:,1], palette=palette)
plt.title('t-SNE with no Labels')
plt.savefig('t-SNE covid-19.png')
plt.show()
```

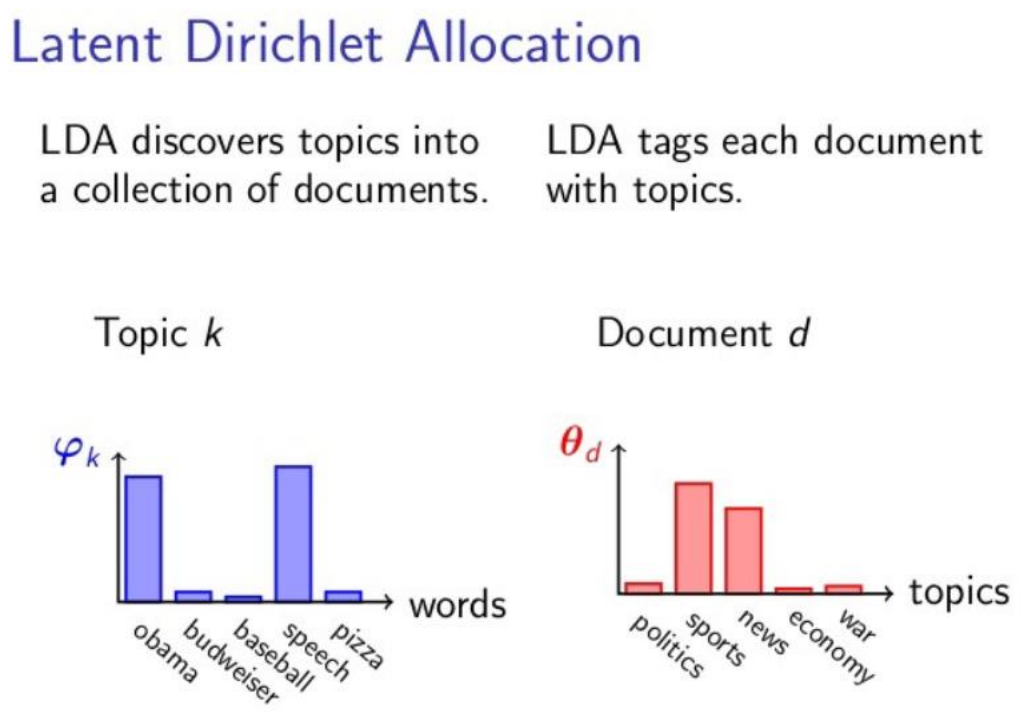


```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(rc={'figure.figsize':(13, 9)})
palette = sns.hls_palette(20, l=0.4, s=0.9)
sns.scatterplot(X_embedded[:,0], X_embedded[:,1], hue=y_pred, legend='full', palette=palette)
plt.savefig('improved cluster tsne.png')
plt.show()
```



```python
from sklearn.decomposition import LatentDirichletAllocation
from sklearn.feature_extraction.text import CountVectorizer
Image(url='https://miro.medium.com/max/1276/0*Sj65xR38wDwuxhtr.jpg', width=800, height=800)
```



Latent Dirichlet Allocation

LDA discovers topics into a collection of documents.

LDA tags each document with topics.