15th June, 2022

Youth Climate Activism

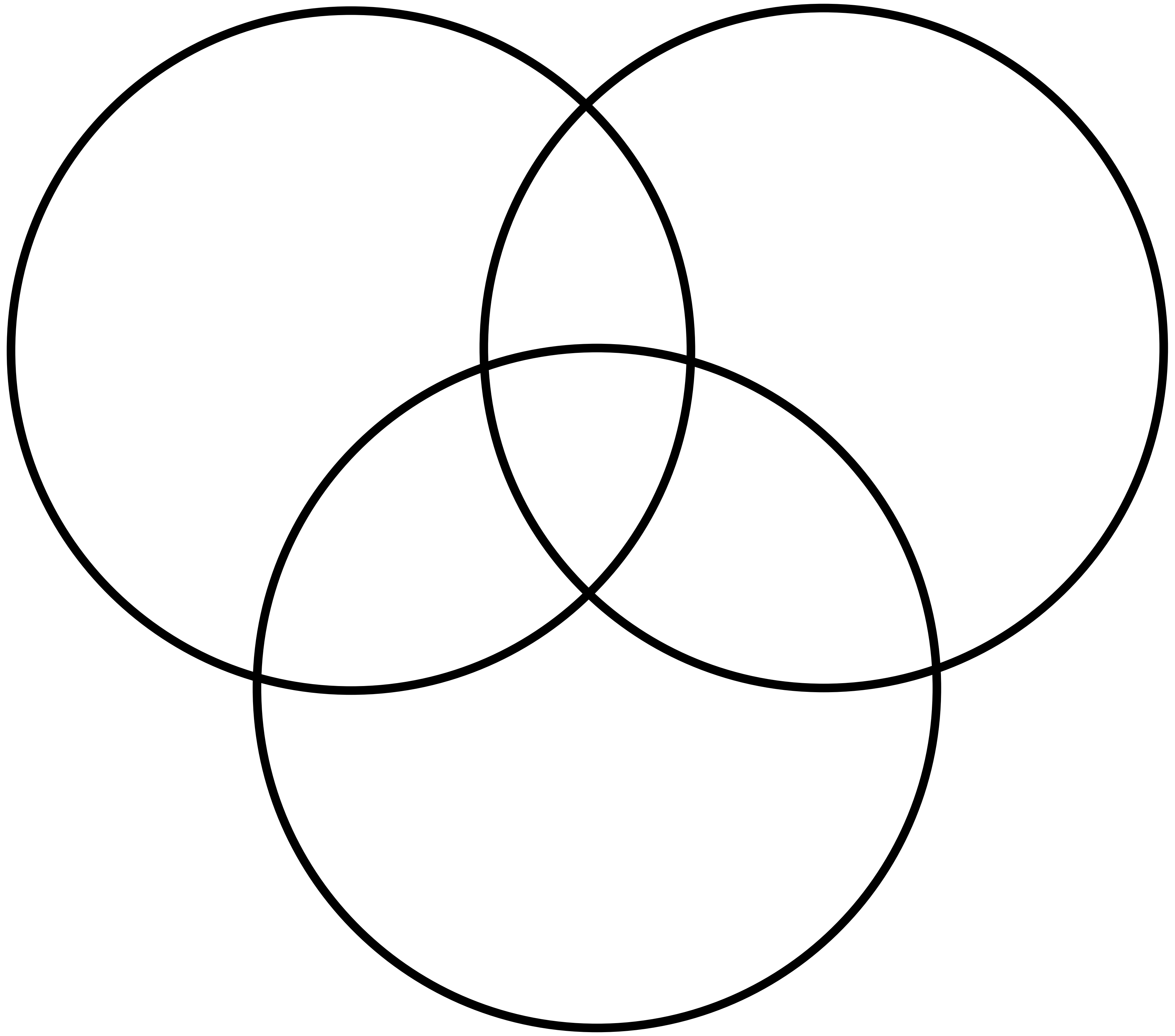
Twitter text clustering and classification

Group members: Diego Farías, Ping Chang, Zhen Li (All participants contributed evenly)

Contents

- Introduction
- Dataset
- Methods and models
- Results
- Conclusion
- Appendix

Introduction



Social media data

Research project:

Planning with Youth

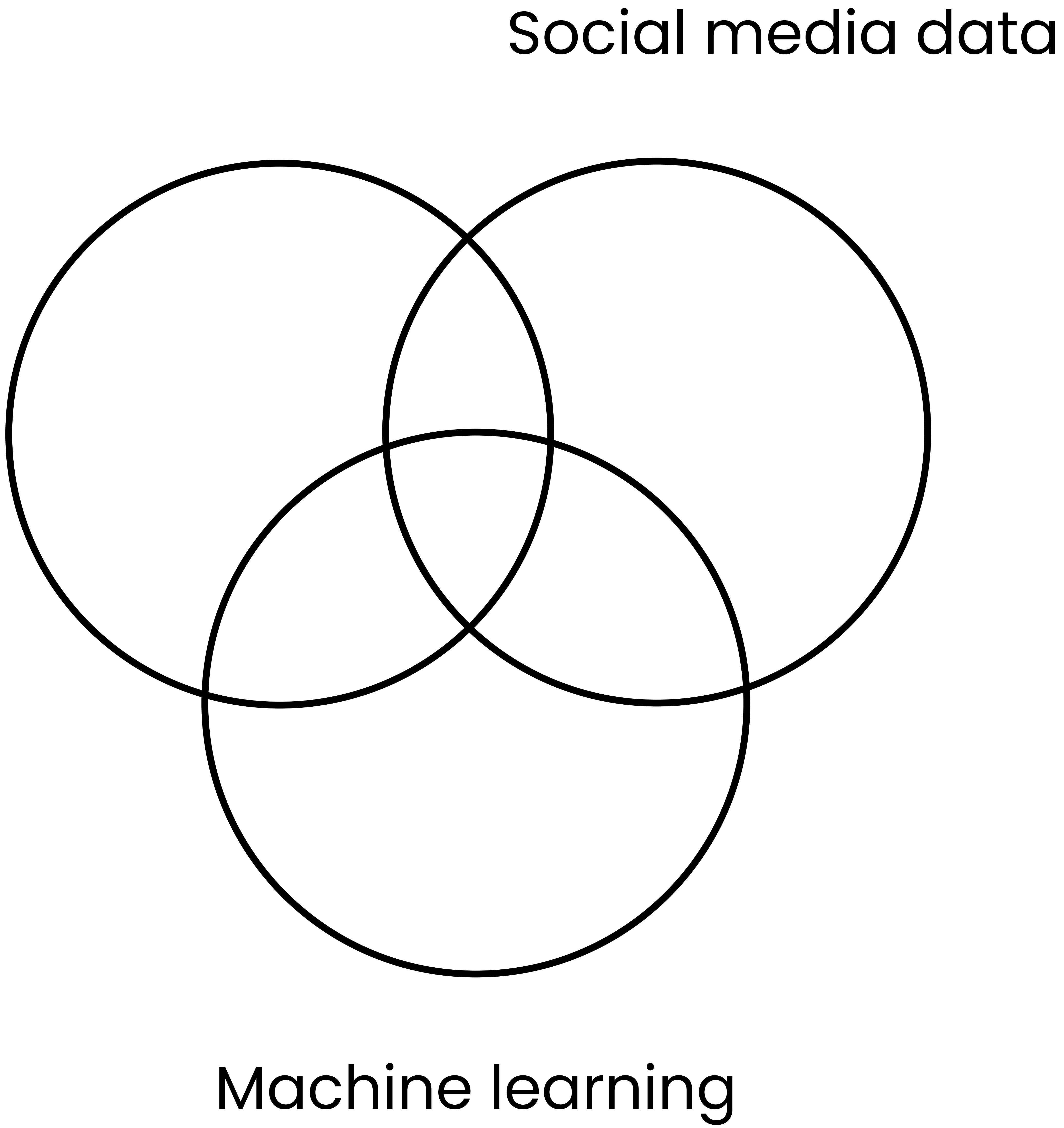
Machine learning

Introduction

Research project:

Planning with Youth

Problem: Can we identify and what **themes** of youth activism on the topic of climate change can be identified from social media (twitter)?



Social media data

Machine learning

Dataset

Dataset description

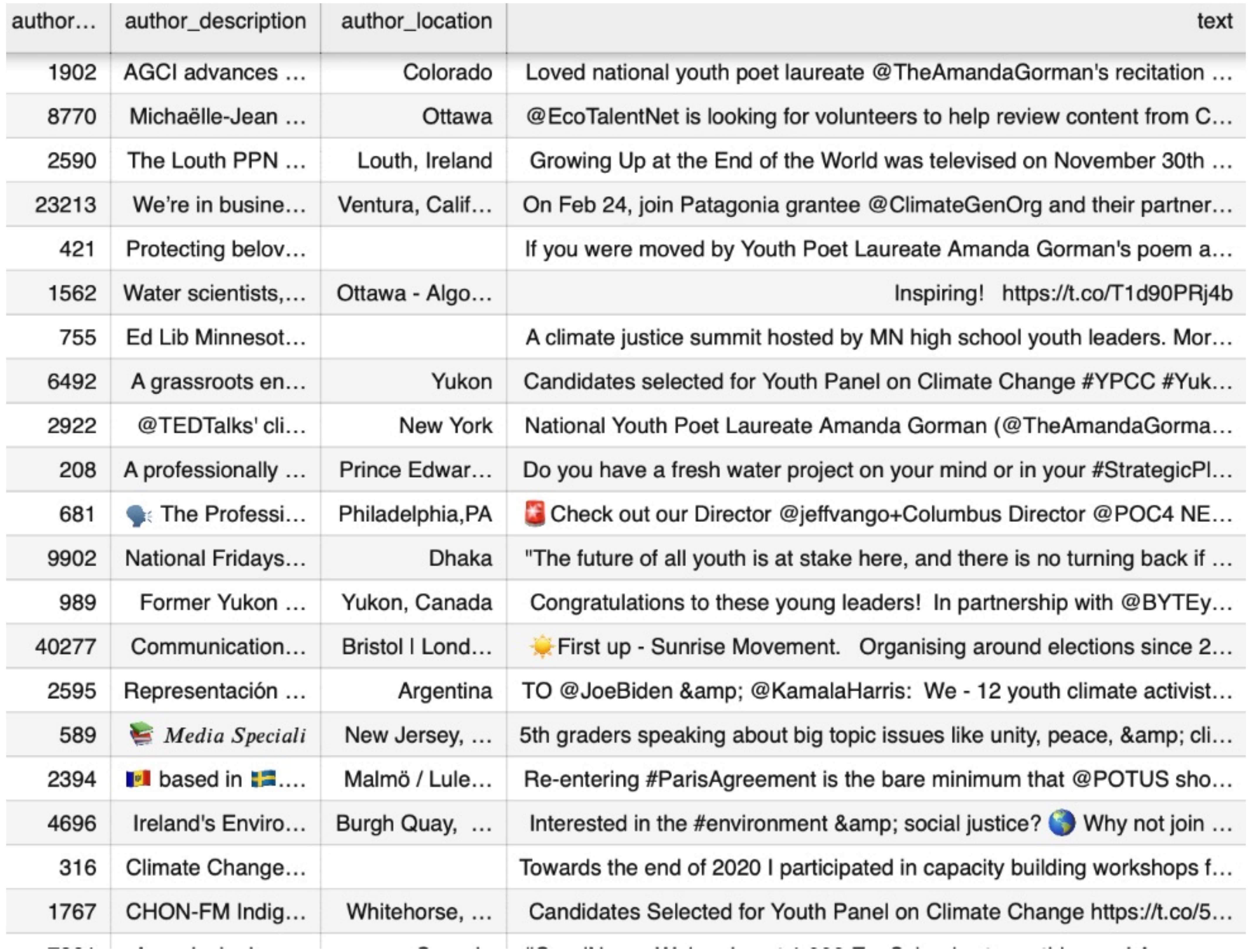
Data scaping: Twitter API v2

Searching words: youth AND climate

Time period: 21/01/2020 –20/01/2021

Number of tweets: 47785

Size: 14.9 Mb



author…	author_description	author_location	text
1902	AGCI advances …	Colorado	Loved national youth poet laureate @TheAmandaGorman's recitation …
8770	Michaëlle-Jean …	Ottawa	@EcoTalentNet is looking for volunteers to help review content from C…
2590	The Louth PPN …	Louth, Ireland	Growing Up at the End of the World was televised on November 30th …
23213	We're in busine…	Ventura, Calif…	On Feb 24, join Patagonia grantee @ClimateGenOrg and their partner…
421	Protecting belov…		If you were moved by Youth Poet Laureate Amanda Gorman's poem a…
1562	Water scientists,…	Ottawa - Algo…	Inspiring! https://t.co/T1d90PRj4b
755	Ed Lib Minnesot…		A climate justice summit hosted by MN high school youth leaders. Mor…
6492	A grassroots en…	Yukon	Candidates selected for Youth Panel on Climate Change #YPCC #Yuk…
2922	@TEDTalks' cli…	New York	National Youth Poet Laureate Amanda Gorman (@TheAmandaGorma…
208	A professionally …	Prince Edwar…	Do you have a fresh water project on your mind or in your #StrategicPl…
681	🗣 The Professi…	Philadelphia,PA	🚨 Check out our Director @jeffvango+Columbus Director @POC4 NE…
9902	National Fridays…	Dhaka	"The future of all youth is at stake here, and there is no turning back if …
989	Former Yukon …	Yukon, Canada	Congratulations to these young leaders! In partnership with @BYTEy…
40277	Communication…	Bristol I Lond…	☀️ First up - Sunrise Movement. Organising around elections since 2…
2595	Representación …	Argentina	TO @JoeBiden &amp; @KamalaHarris: We - 12 youth climate activist…
589	📚 *Media Speciali*	New Jersey, …	5th graders speaking about big topic issues like unity, peace, &amp; cli…
2394	🇲🇩 based in 🇸🇪….	Malmö / Lule…	Re-entering #ParisAgreement is the bare minimum that @POTUS sho…
4696	Ireland's Enviro…	Burgh Quay, …	Interested in the #environment &amp; social justice? 🌎 Why not join …
316	Climate Change…		Towards the end of 2020 I participated in capacity building workshops f…
1767	CHON-FM Indig…	Whitehorse, …	Candidates Selected for Youth Panel on Climate Change https://t.co/5…

Data preprocessing

Data cleaning *Regular expression + spaCy*

duplicate texts

Stop words: default and universal words in tweets

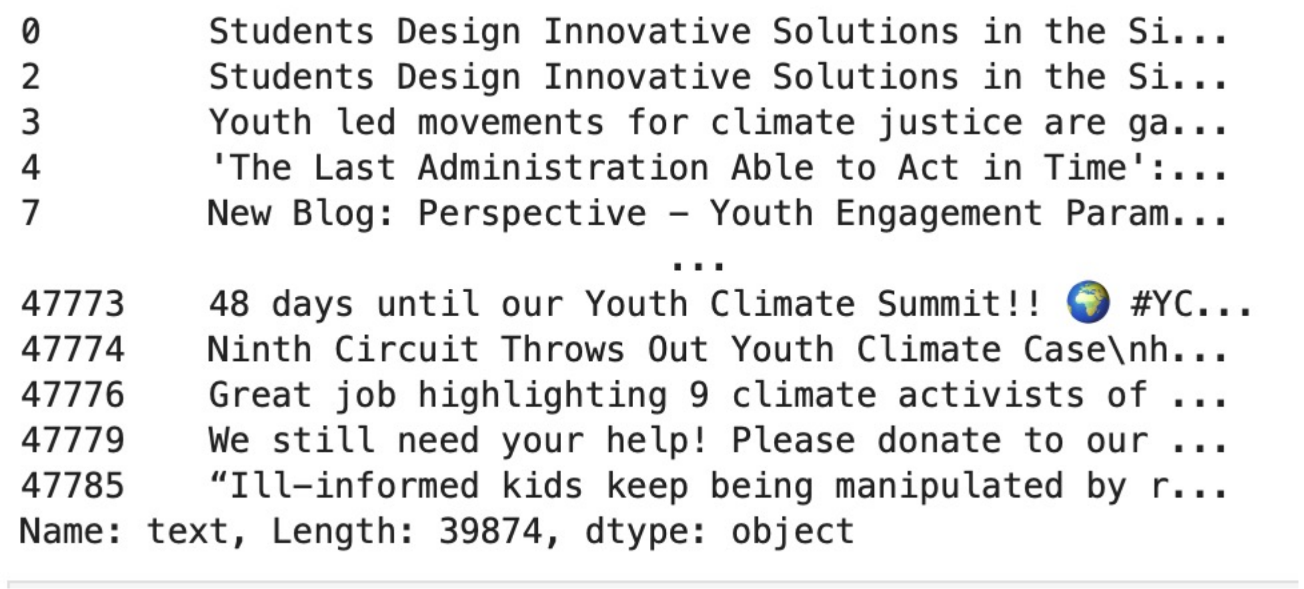
Numbers

email

URL

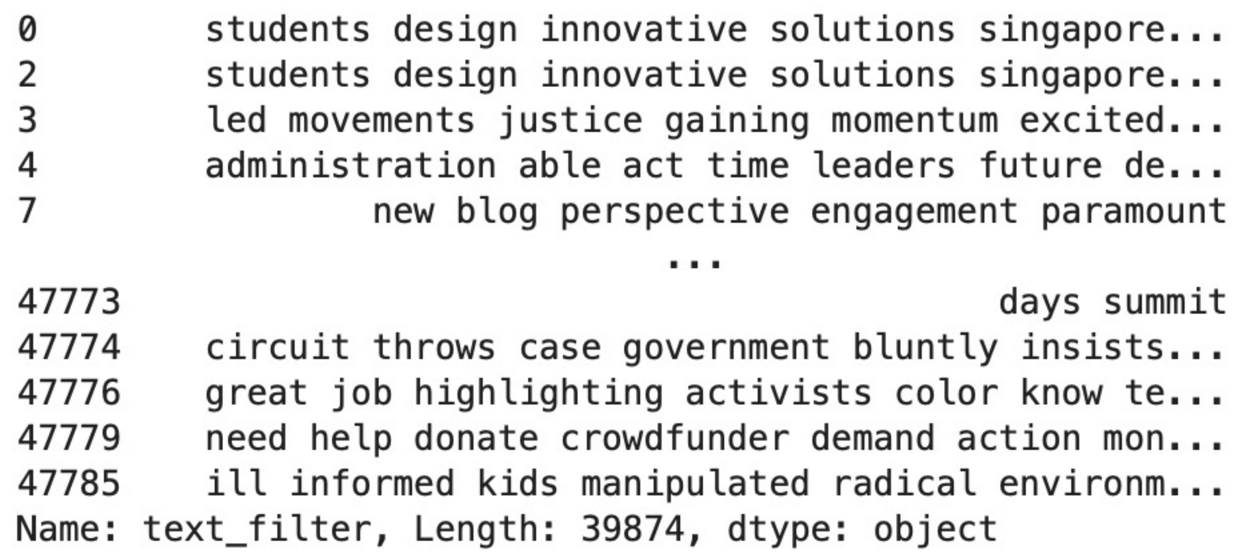
emoji

Punctuations



```
0        Students Design Innovative Solutions in the Si...
2        Students Design Innovative Solutions in the Si...
3        Youth led movements for climate justice are ga...
4        'The Last Administration Able to Act in Time':...
7        New Blog: Perspective – Youth Engagement Param...
                               ...
47773    48 days until our Youth Climate Summit!! 🌍 #YC...
47774    Ninth Circuit Throws Out Youth Climate Case\nh...
47776    Great job highlighting 9 climate activists of ...
47779    We still need your help! Please donate to our ...
47785    “Ill–informed kids keep being manipulated by r...
Name: text, Length: 39874, dtype: object
```

Raw texts

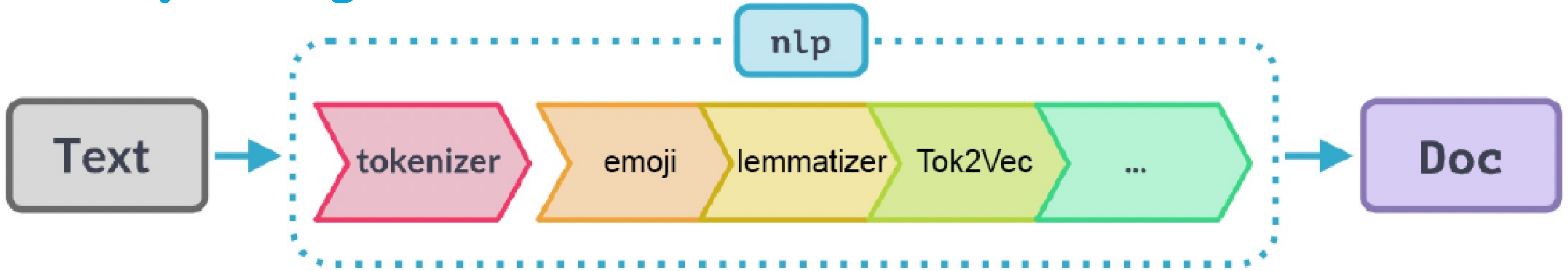


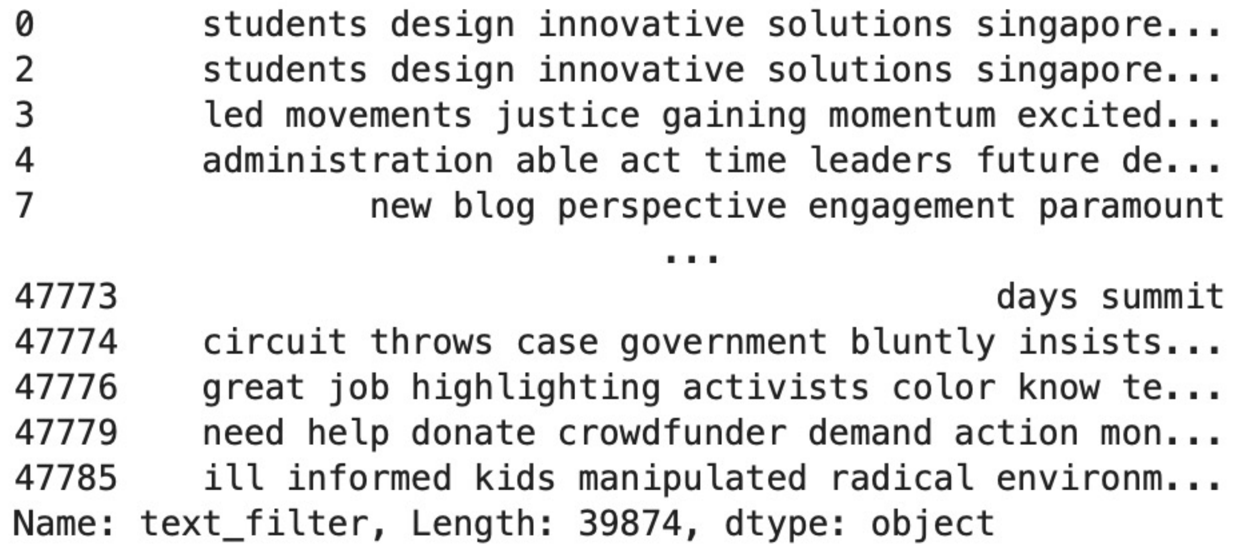
```
0        students design innovative solutions singapore...
2        students design innovative solutions singapore...
3        led movements justice gaining momentum excited...
4        administration able act time leaders future de...
7                new blog perspective engagement paramount
                               ...
47773                                          days summit
47774    circuit throws case government bluntly insists...
47776    great job highlighting activists color know te...
47779    need help donate crowdfunder demand action mon...
47785    ill informed kids manipulated radical environm...
Name: text_filter, Length: 39874, dtype: object
```

Cleaned texts

Data preprocessing

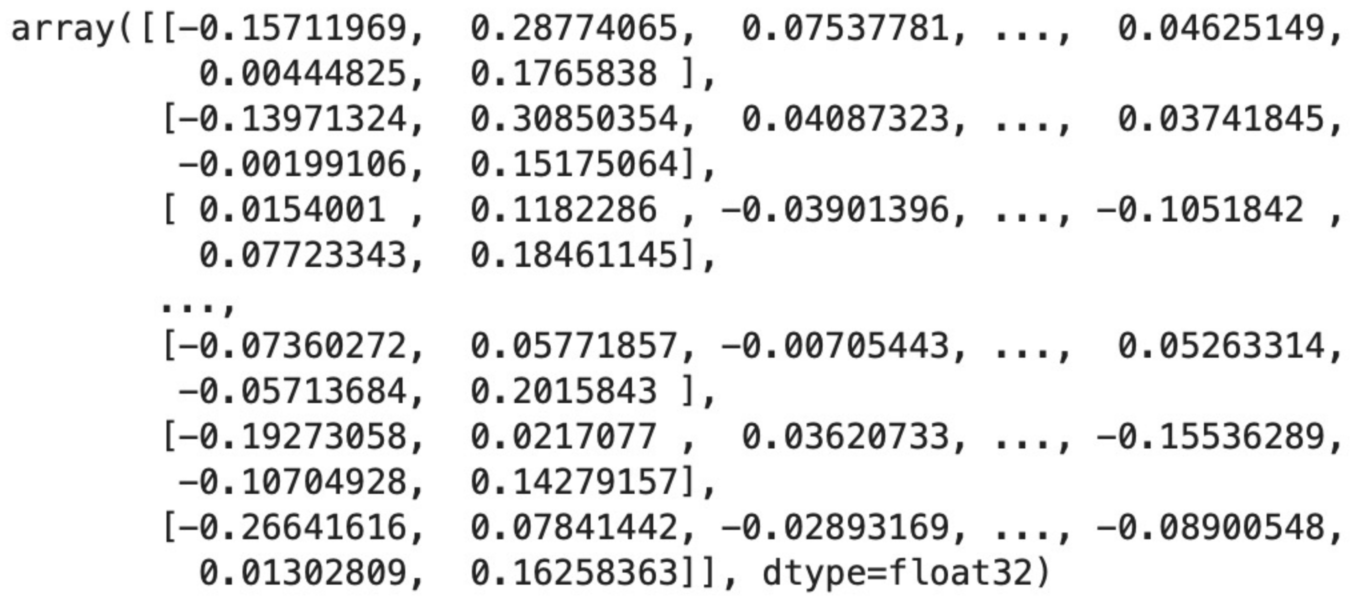
Text vector with spaCy





```
0        students design innovative solutions singapore...
2        students design innovative solutions singapore...
3        led movements justice gaining momentum excited...
4        administration able act time leaders future de...
7                new blog perspective engagement paramount
                               ...
47773                                          days summit
47774    circuit throws case government bluntly insists...
47776    great job highlighting activists color know te...
47779    need help donate crowdfunder demand action mon...
47785    ill informed kids manipulated radical environm...
Name: text_filter, Length: 39874, dtype: object
```

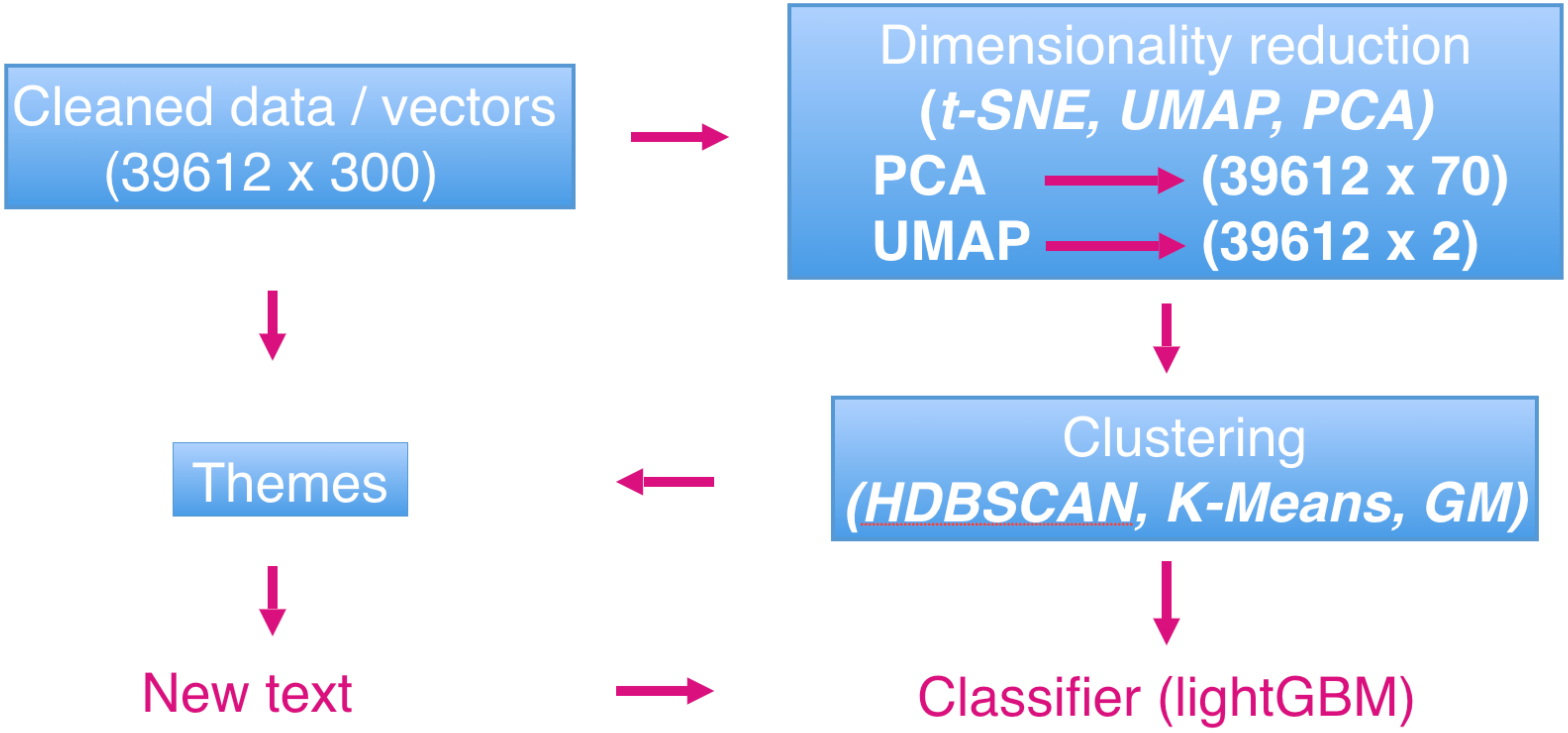
Textual data



```
array([[-0.15711969,  0.28774065,  0.07537781, ...,  0.04625149,
         0.00444825,  0.1765838 ],
       [-0.13971324,  0.30850354,  0.04087323, ...,  0.03741845,
        -0.00199106,  0.15175064],
       [ 0.0154001 ,  0.1182286 , -0.03901396, ..., -0.1051842 ,
         0.07723343,  0.18461145],
       ...,
       [-0.07360272,  0.05771857, -0.00705443, ...,  0.05263314,
        -0.05713684,  0.2015843 ],
       [-0.19273058,  0.0217077 ,  0.03620733, ..., -0.15536289,
        -0.10704928,  0.14279157],
       [-0.26641616,  0.07841442, -0.02893169, ..., -0.08900548,
         0.01302809,  0.16258363]], dtype=float32)
```

39612 x 300 vectors

Workflow



Cleaned data / vectors (39612 x 300)

Dimensionality reduction *(t-SNE, UMAP, PCA)*

PCA → **(39612 x 70)**

UMAP → **(39612 x 2)**

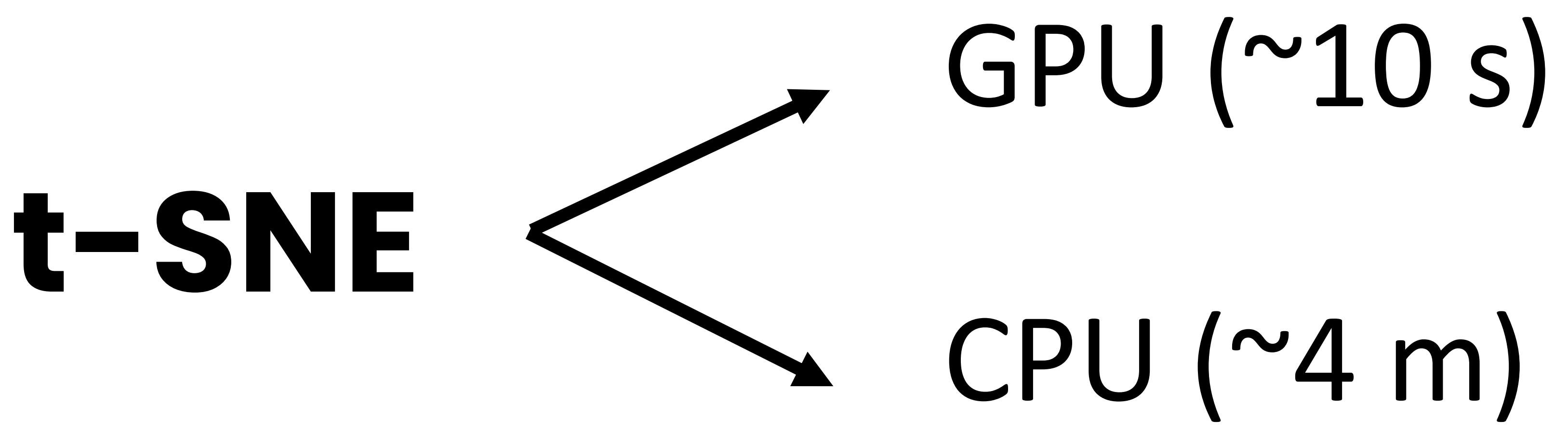
Clustering *(HDBSCAN, K-Means, GM)*

Themes

New text

Classifier (lightGBM)

Dimensionality Reduction

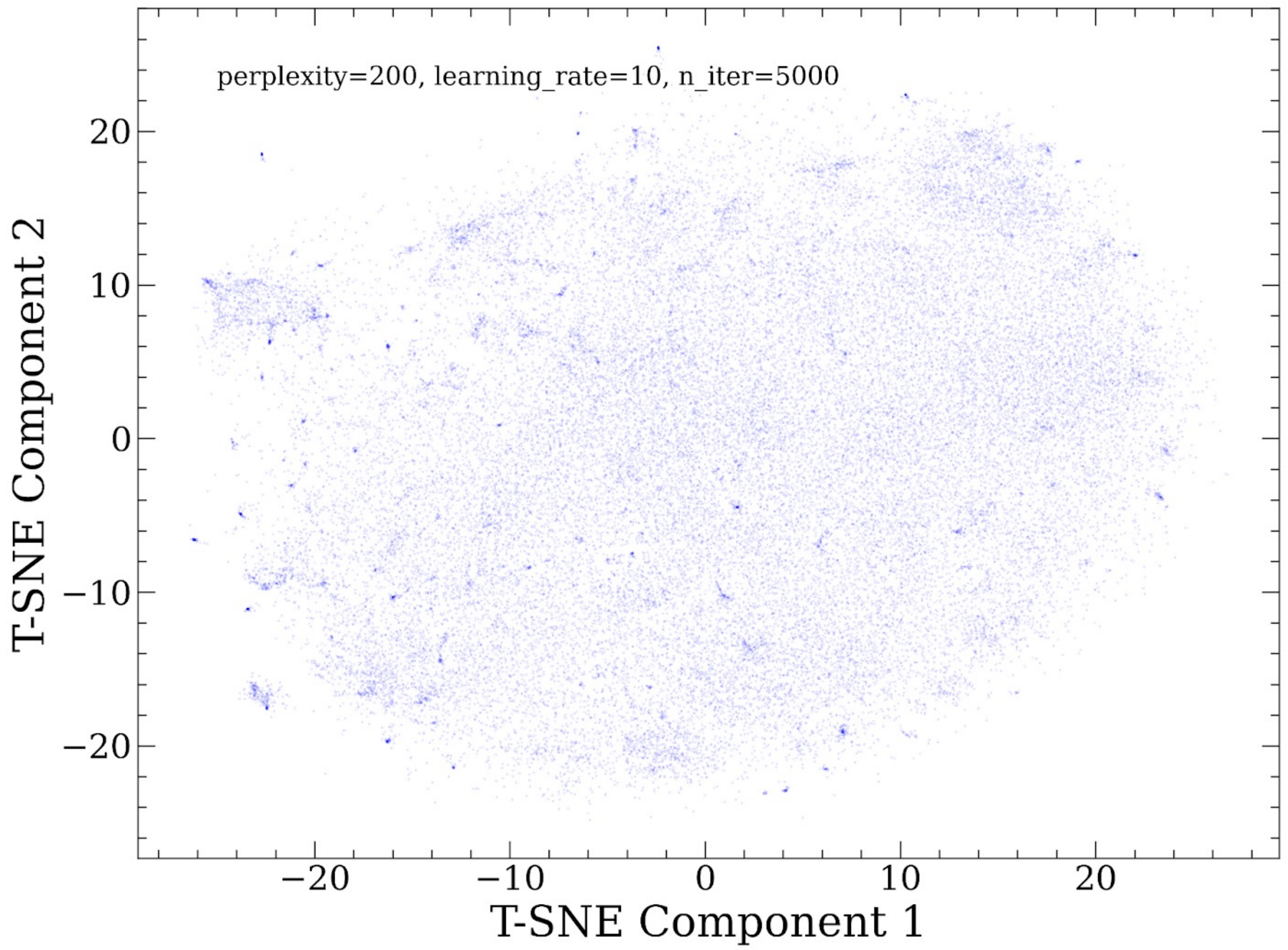


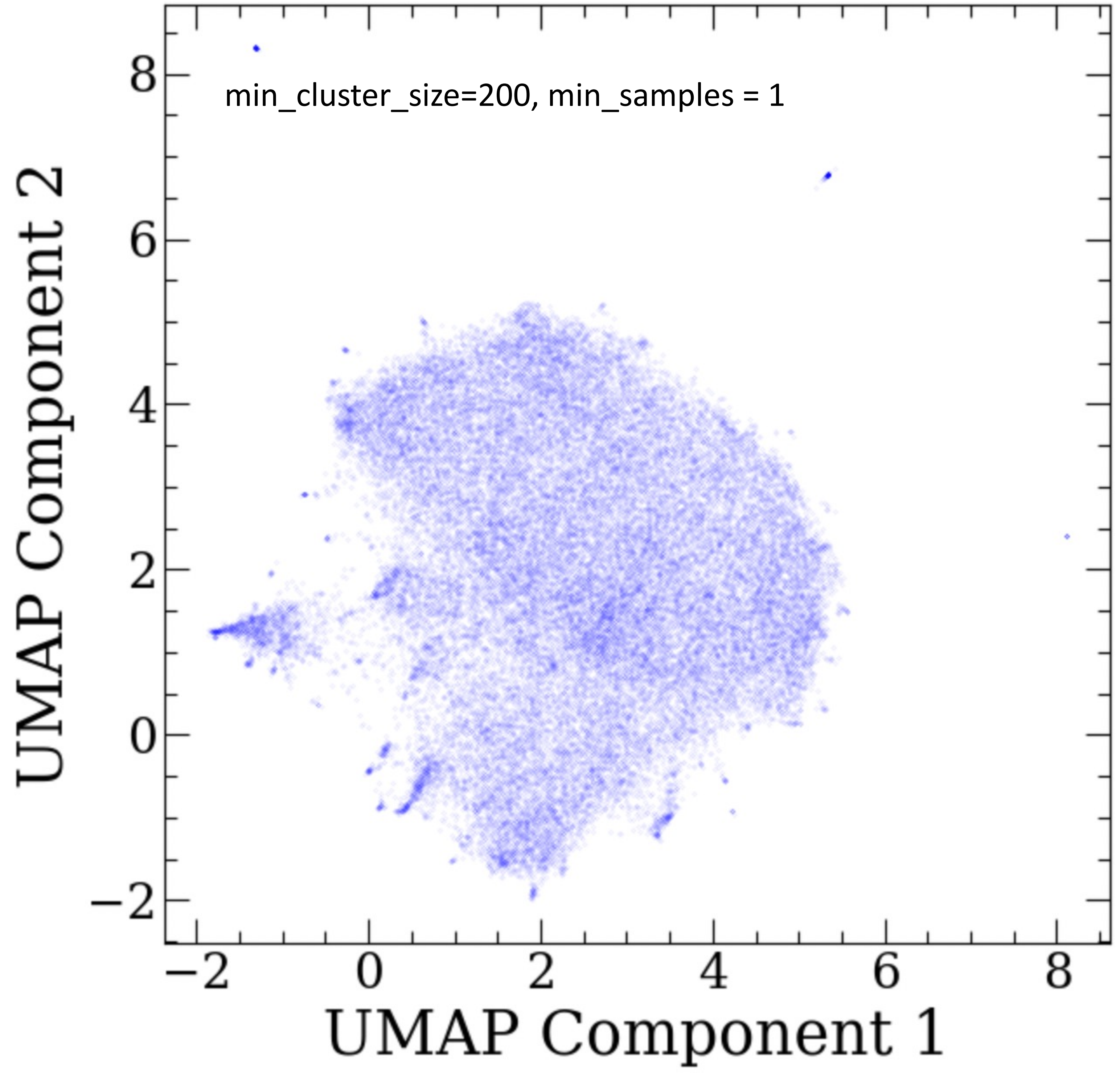
t-SNE

GPU (~10 s)

CPU (~4 m)

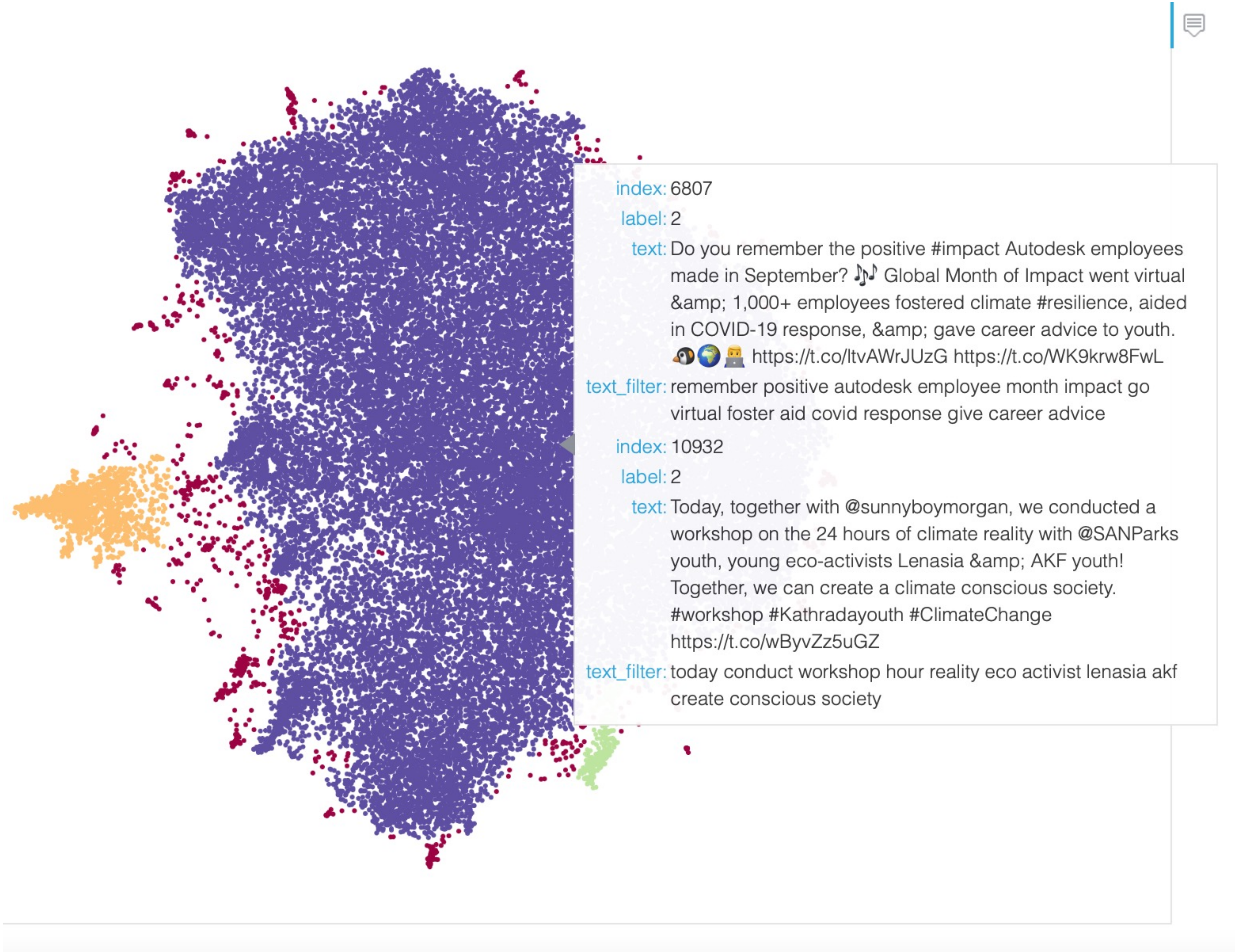
UMAP (~1 m)





Text visualization

Interactive maps (UMAP-plot)



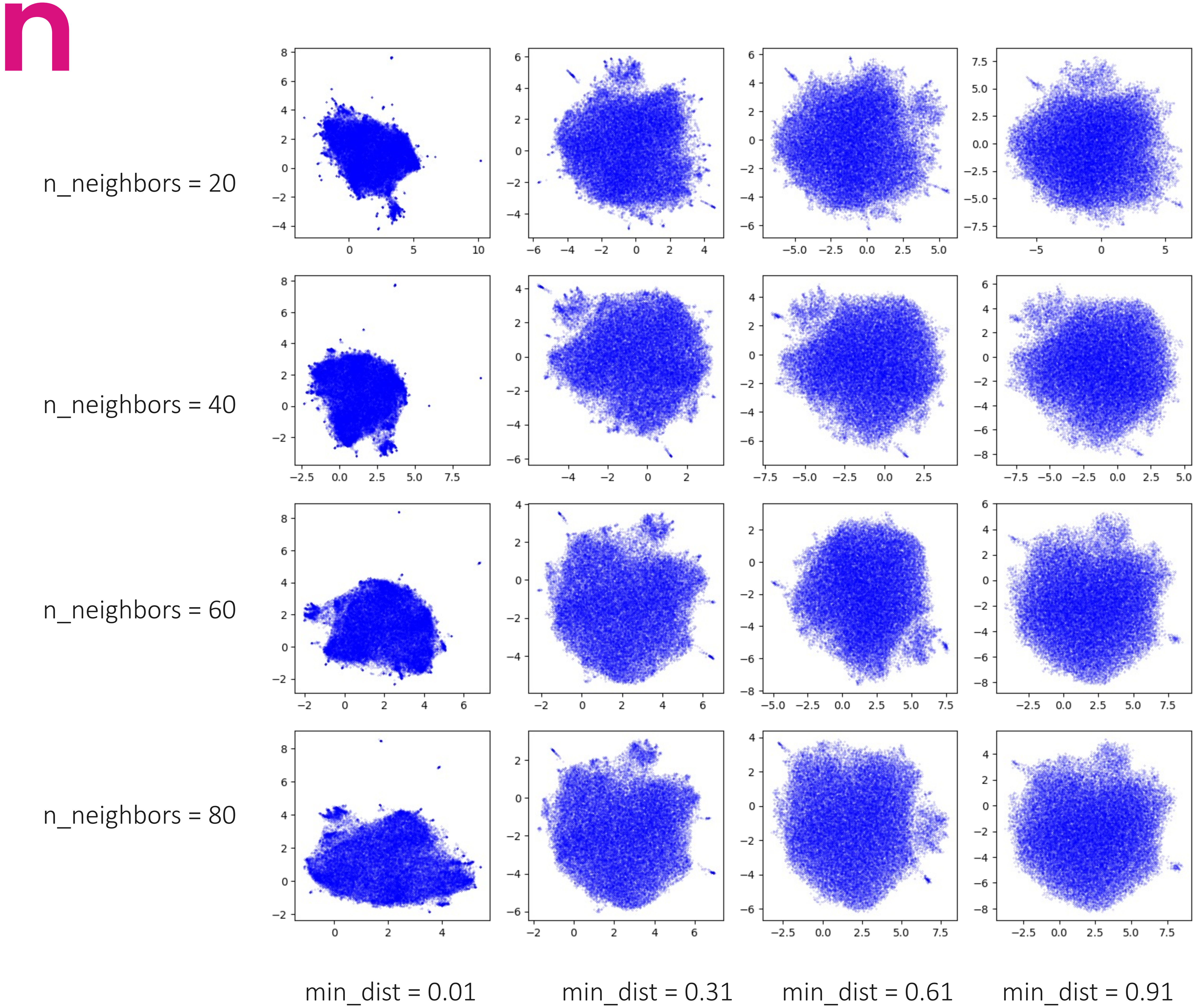
index: 6807
label: 2
text: Do you remember the positive #impact Autodesk employees made in September? 🎶 Global Month of Impact went virtual &amp; 1,000+ employees fostered climate #resilience, aided in COVID-19 response, &amp; gave career advice to youth. 🐧🌍👨‍💻 https://t.co/ltvAWrJUzG https://t.co/WK9krw8FwL
text_filter: remember positive autodesk employee month impact go virtual foster aid covid response give career advice

index: 10932
label: 2
text: Today, together with @sunnyboymorgan, we conducted a workshop on the 24 hours of climate reality with @SANParks youth, young eco-activists Lenasia &amp; AKF youth! Together, we can create a climate conscious society. #workshop #Kathradayouth #ClimateChange https://t.co/wByvZz5uGZ
text_filter: today conduct workshop hour reality eco activist lenasia akf create conscious society

1) Hyperparameter Optimisation

UMAP

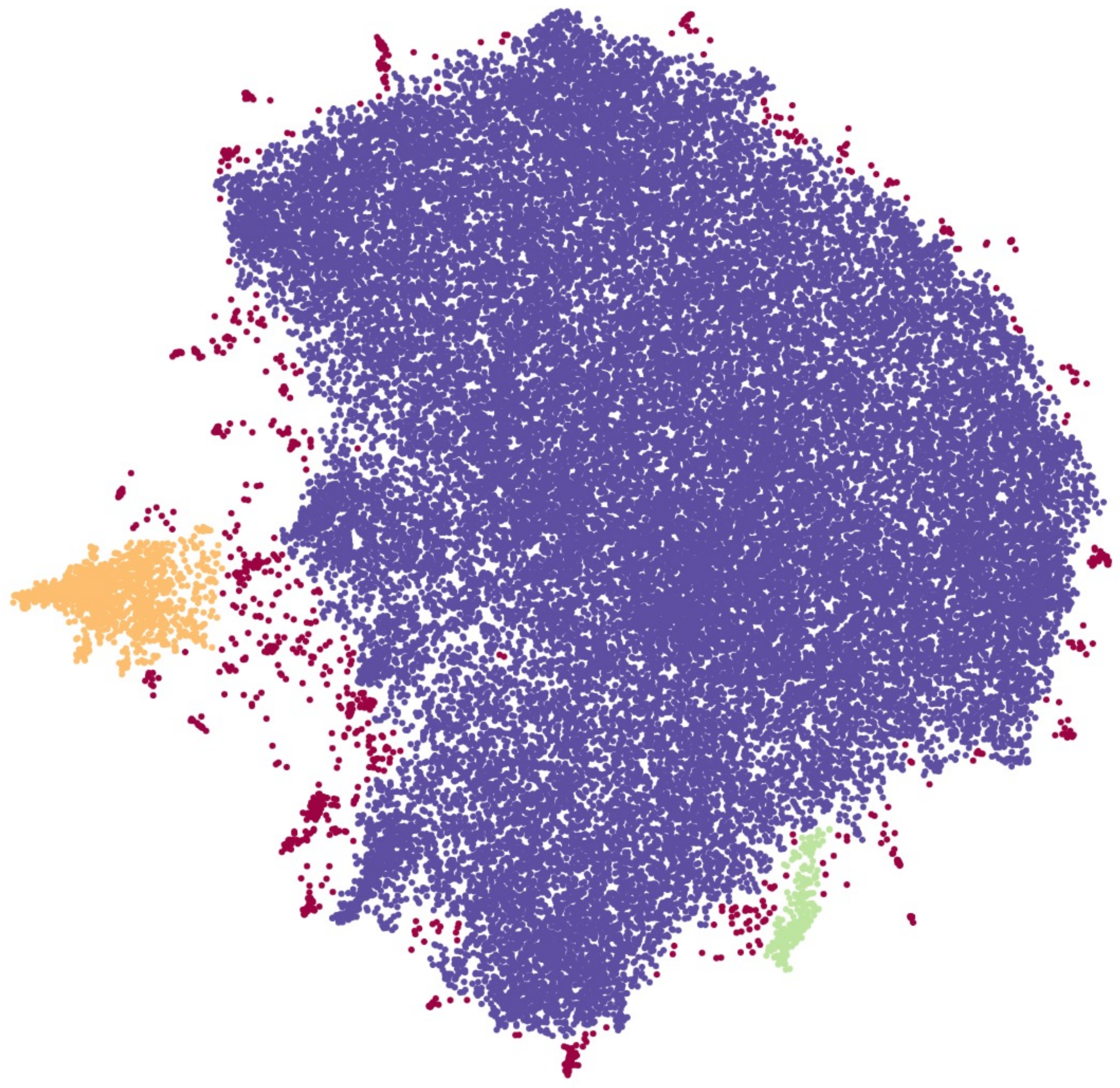
Grid search



Clustering algorithms

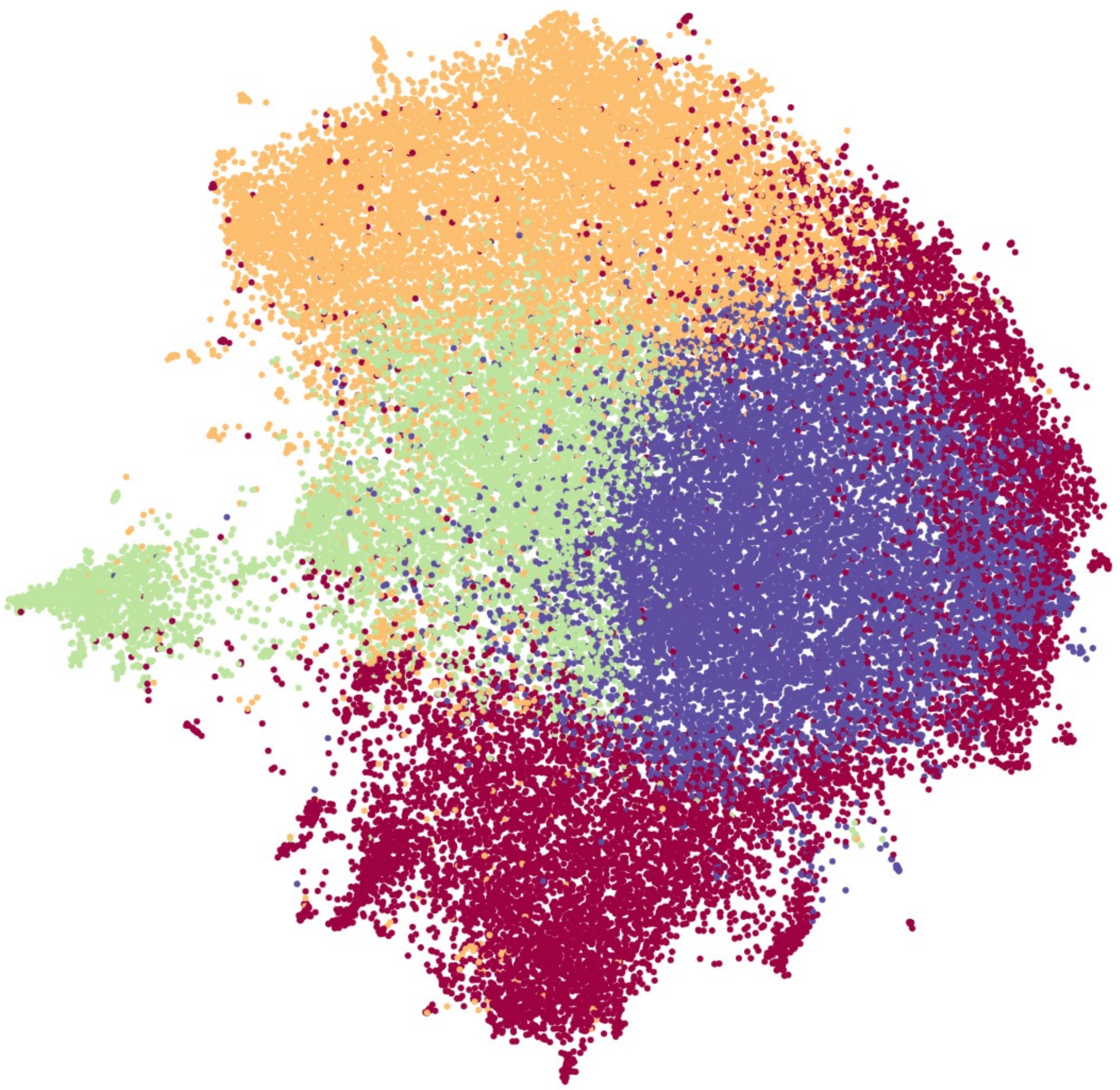
HDBSCAN

(min_cluster_size=200,
min_samples=1)



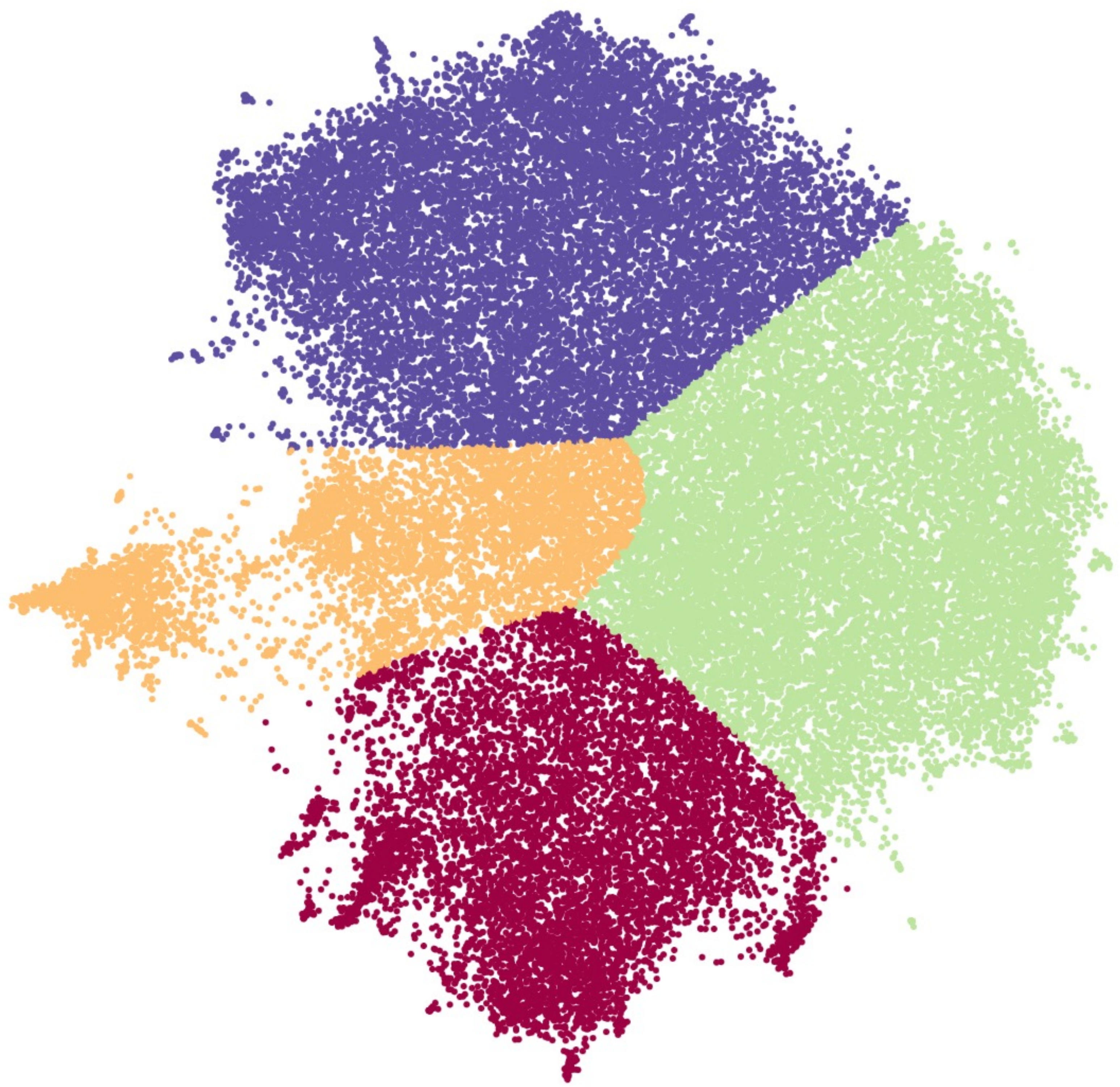
K-means

（n_clusters=4）



Gaussian Mixture

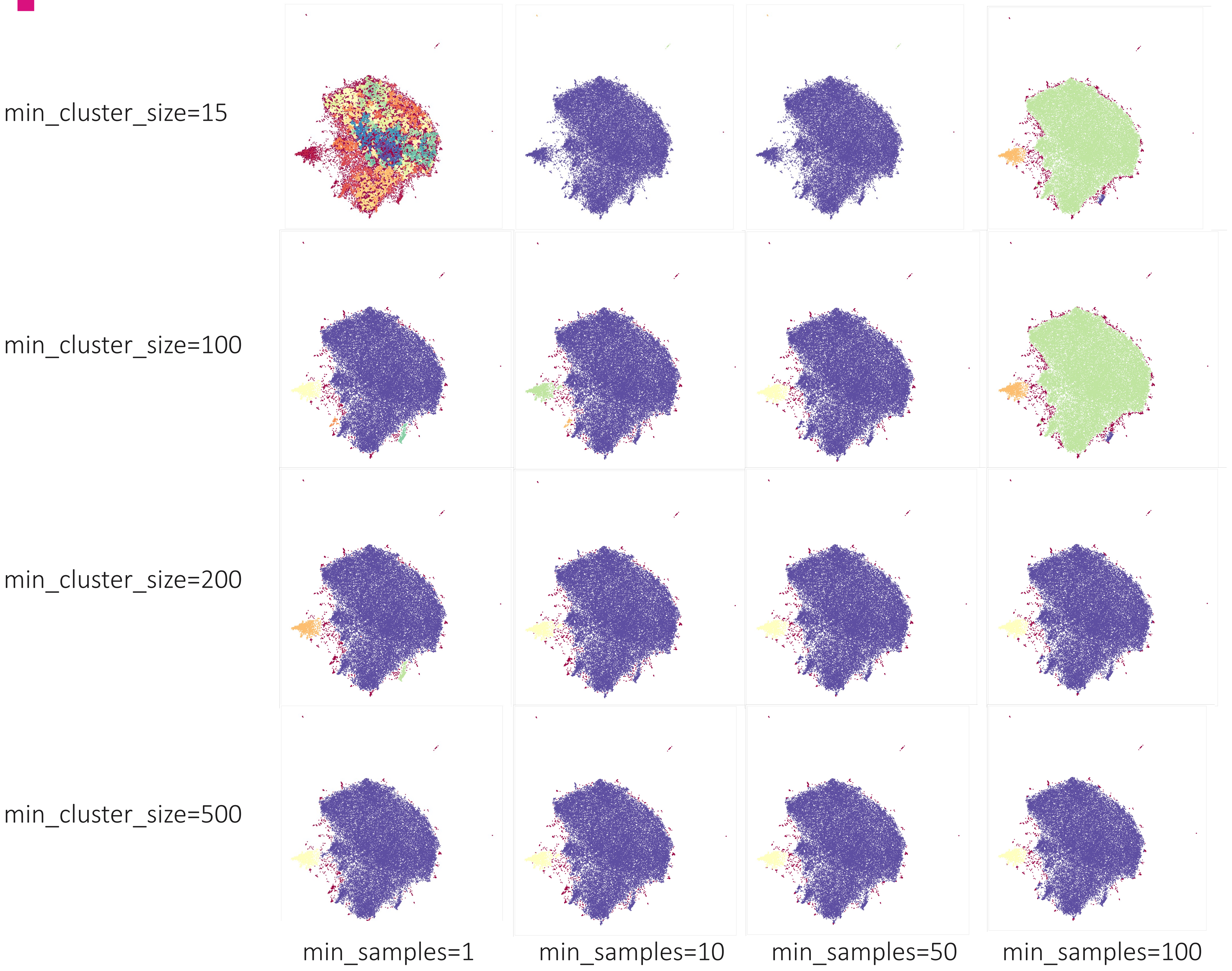
(n_components=4, n_init=100)



2) Hyperparameter Optimisation

HDBSCAN

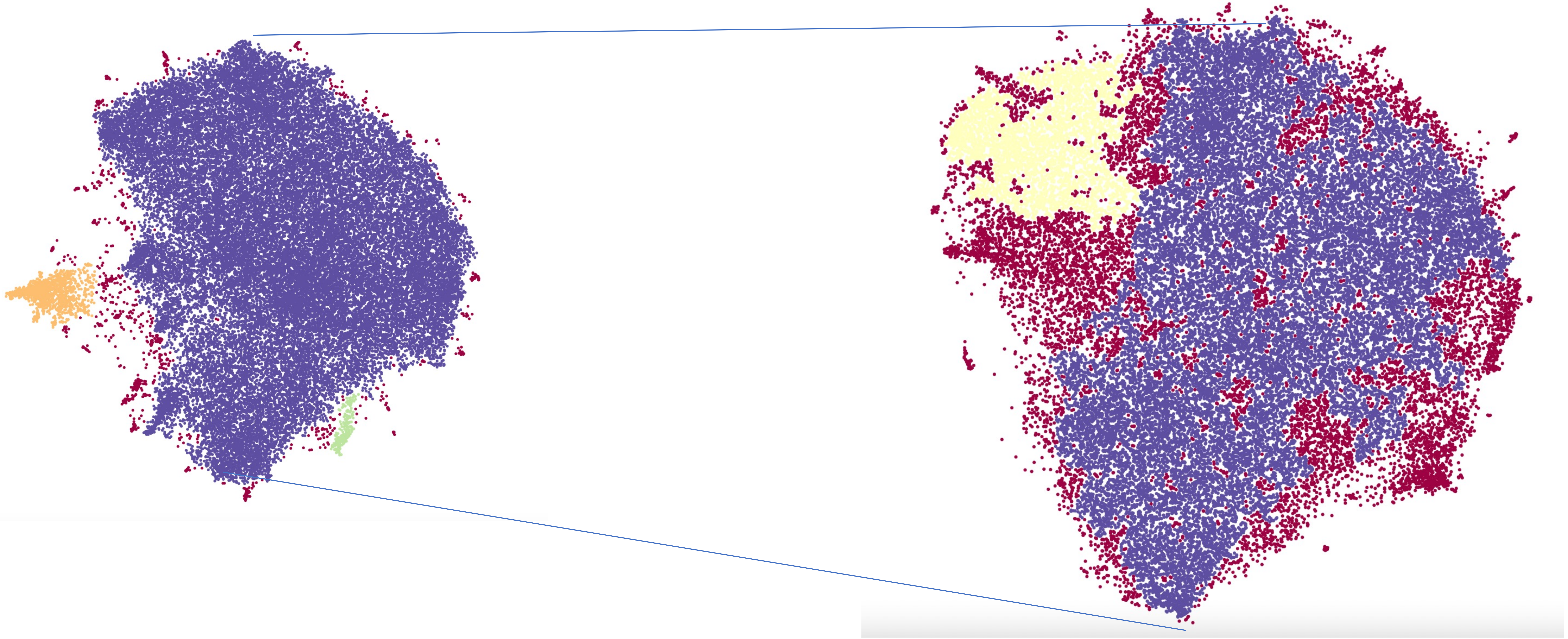
Grid search



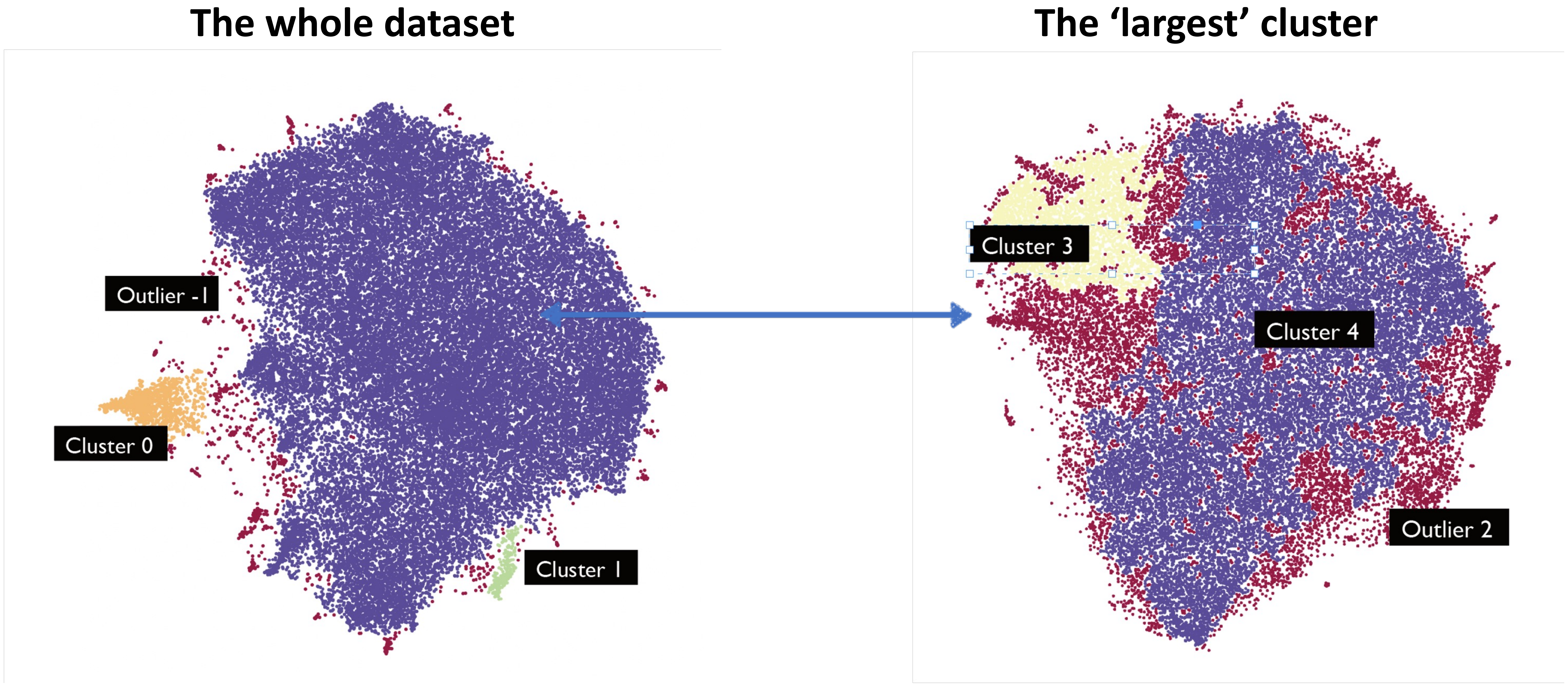
Re-Clustering the largest one

HDBSCAN for the largest cluster

(min_cluster_size=1200, min_samples=1)



Clustering

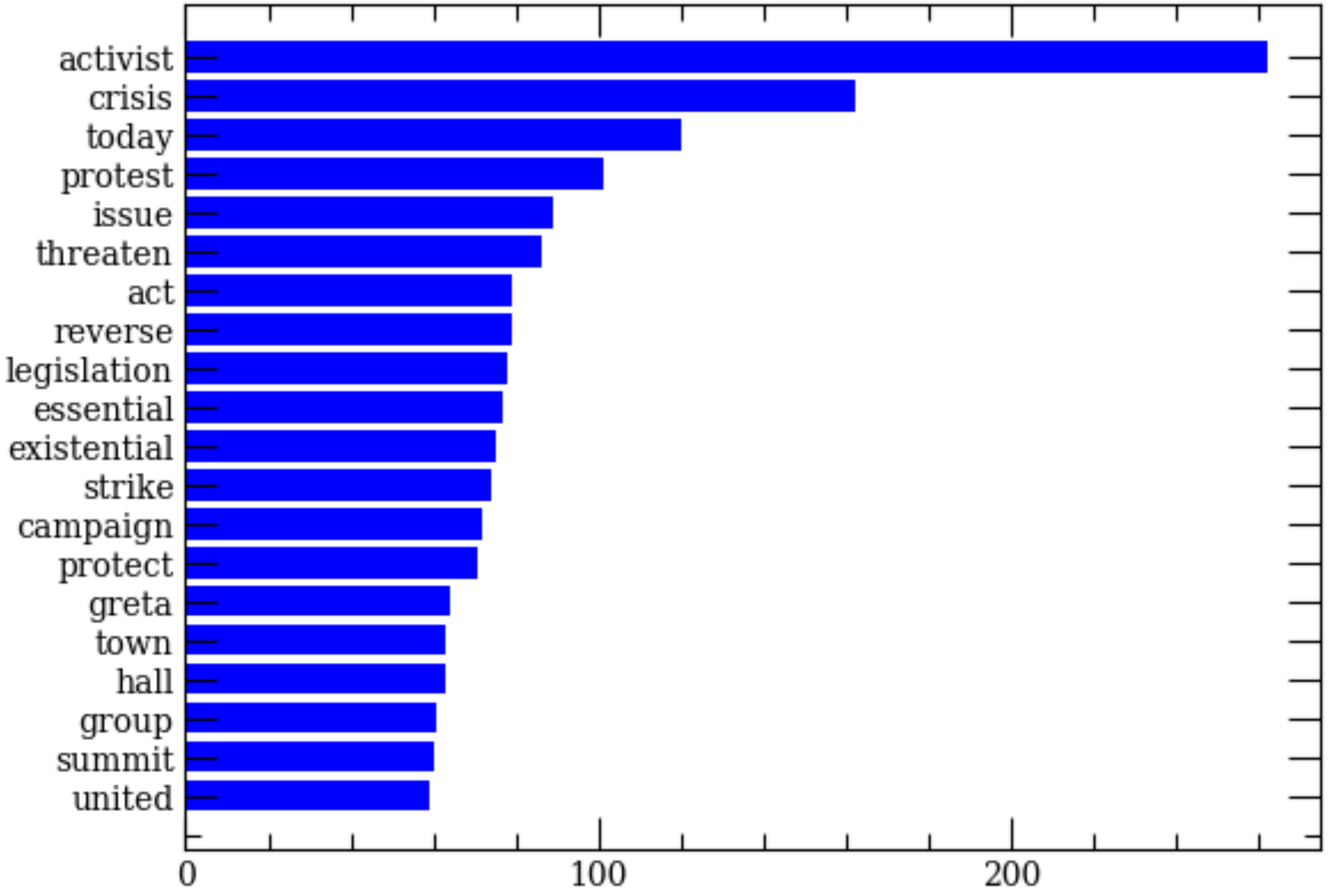


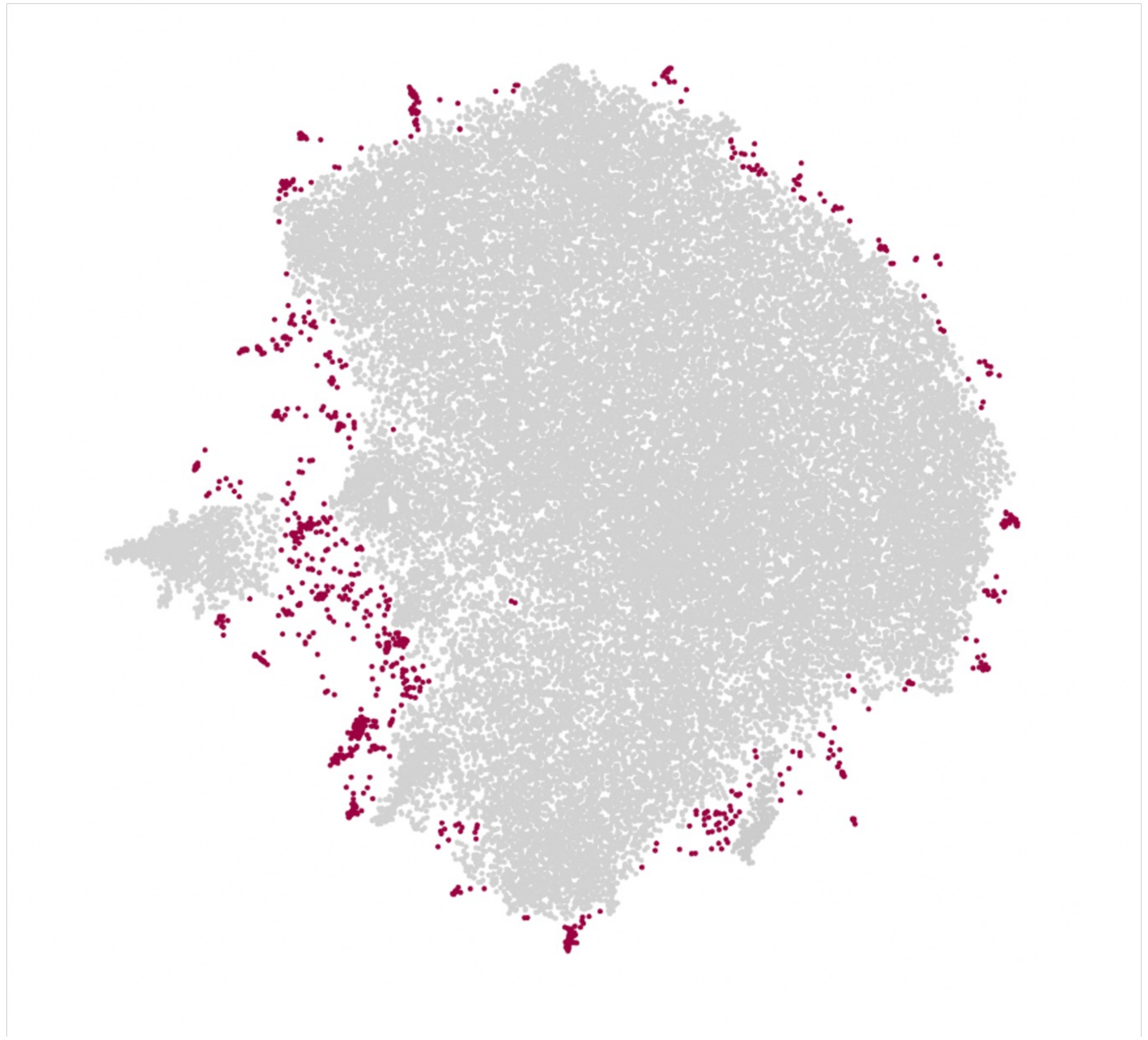
The whole dataset

The ‘largest’ cluster

Clustering

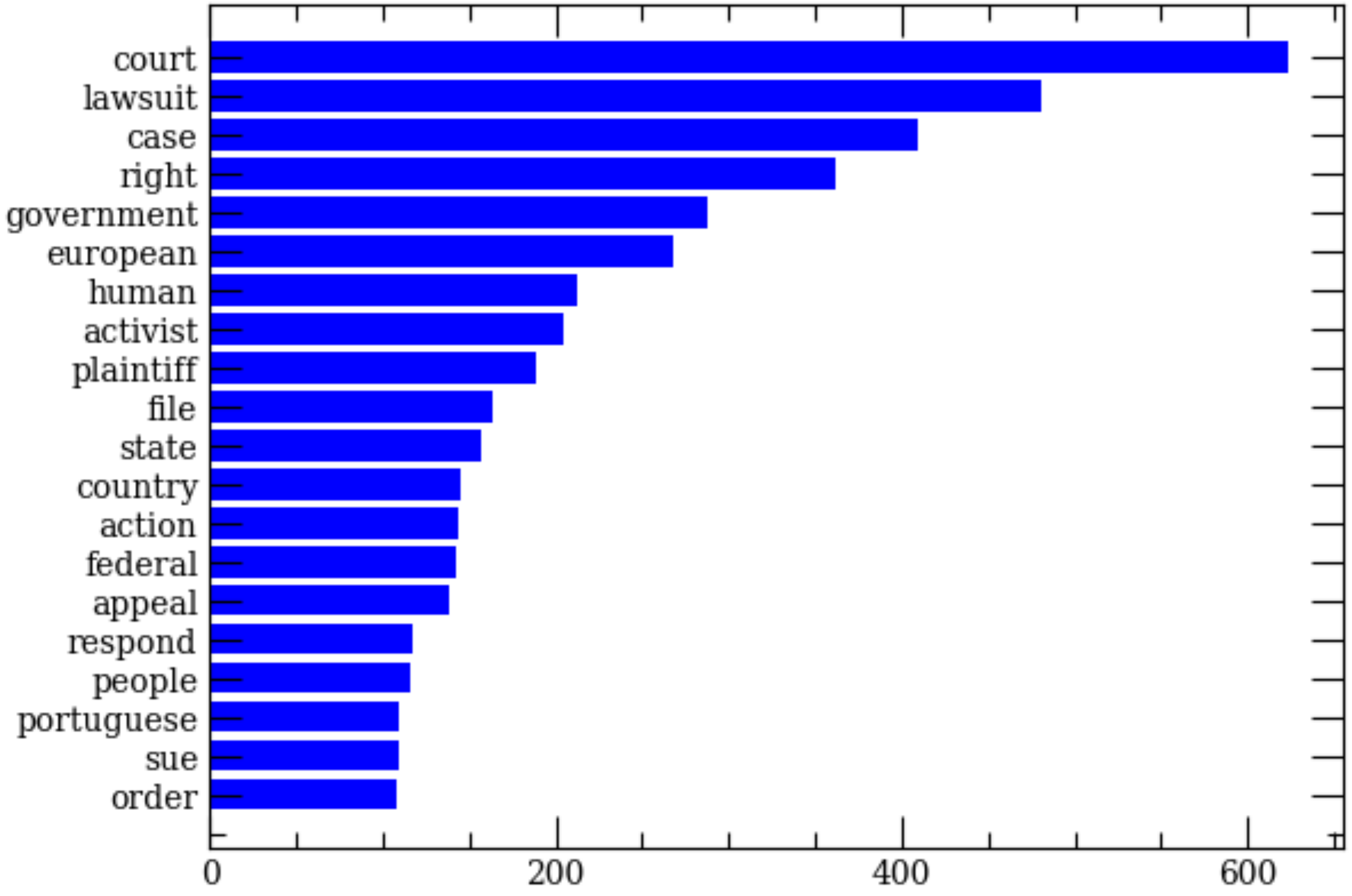
Word frequency and Outlier -1

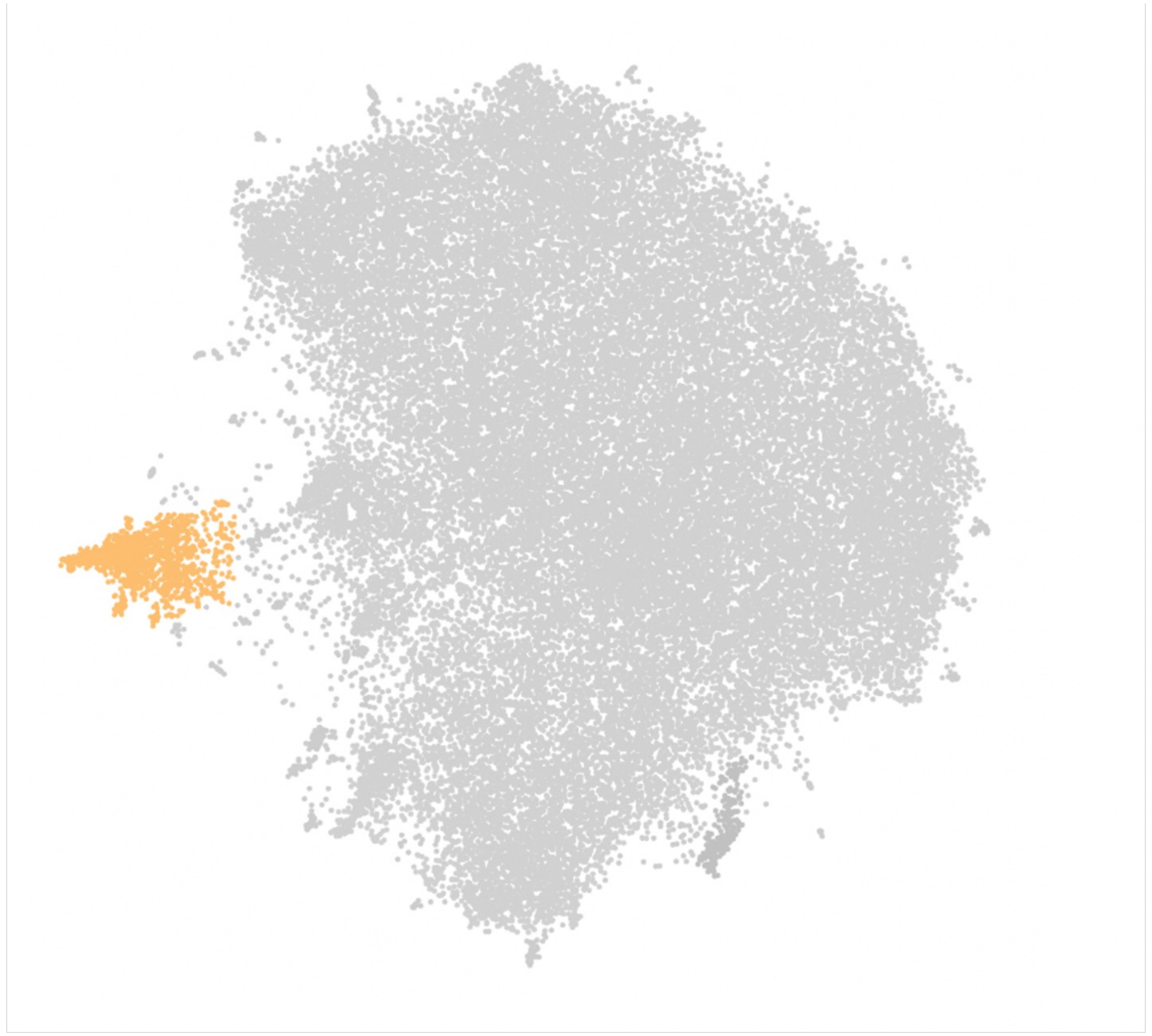




Clustering

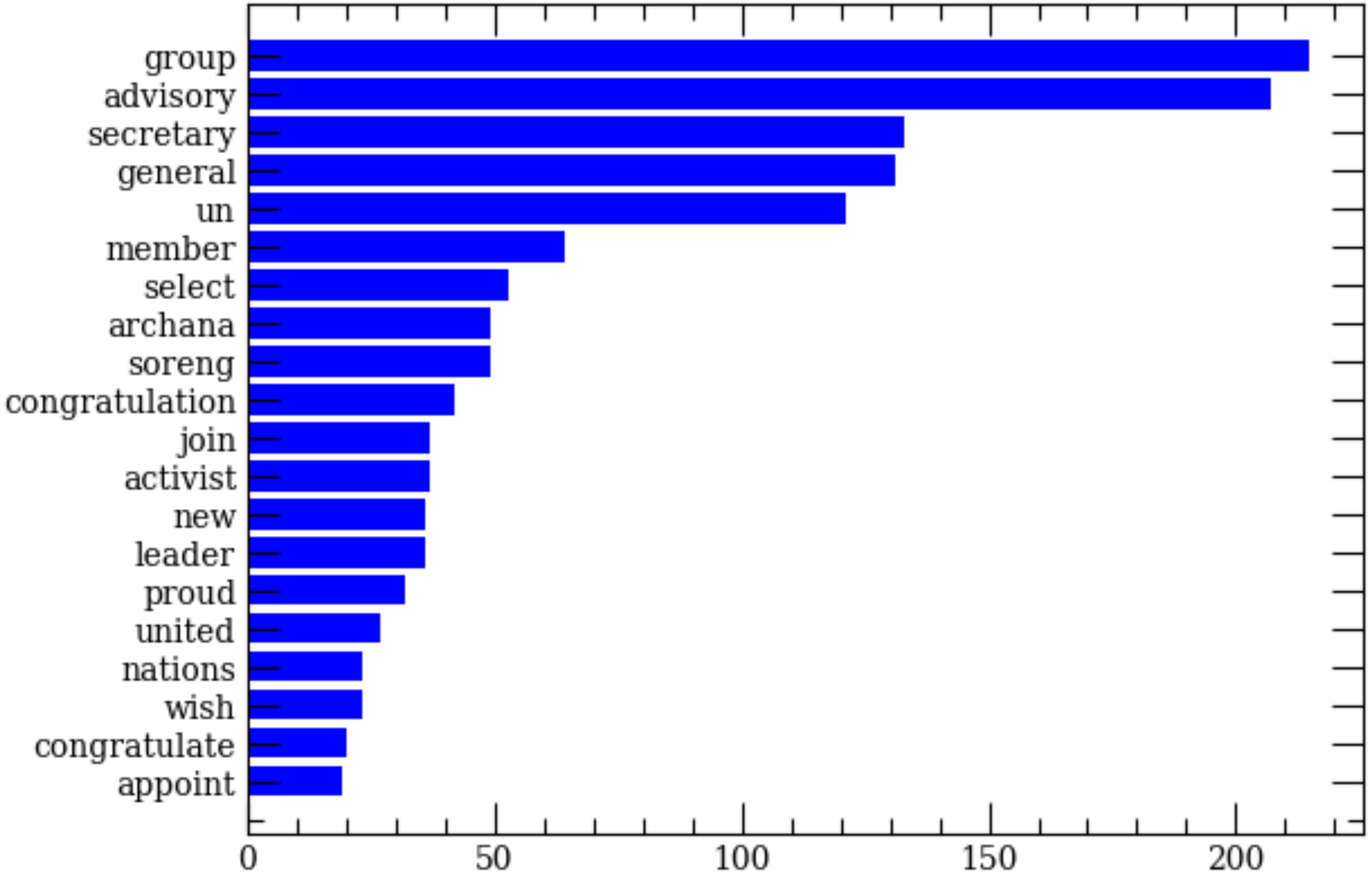
Word frequency and Cluster 0: Legal mobilization

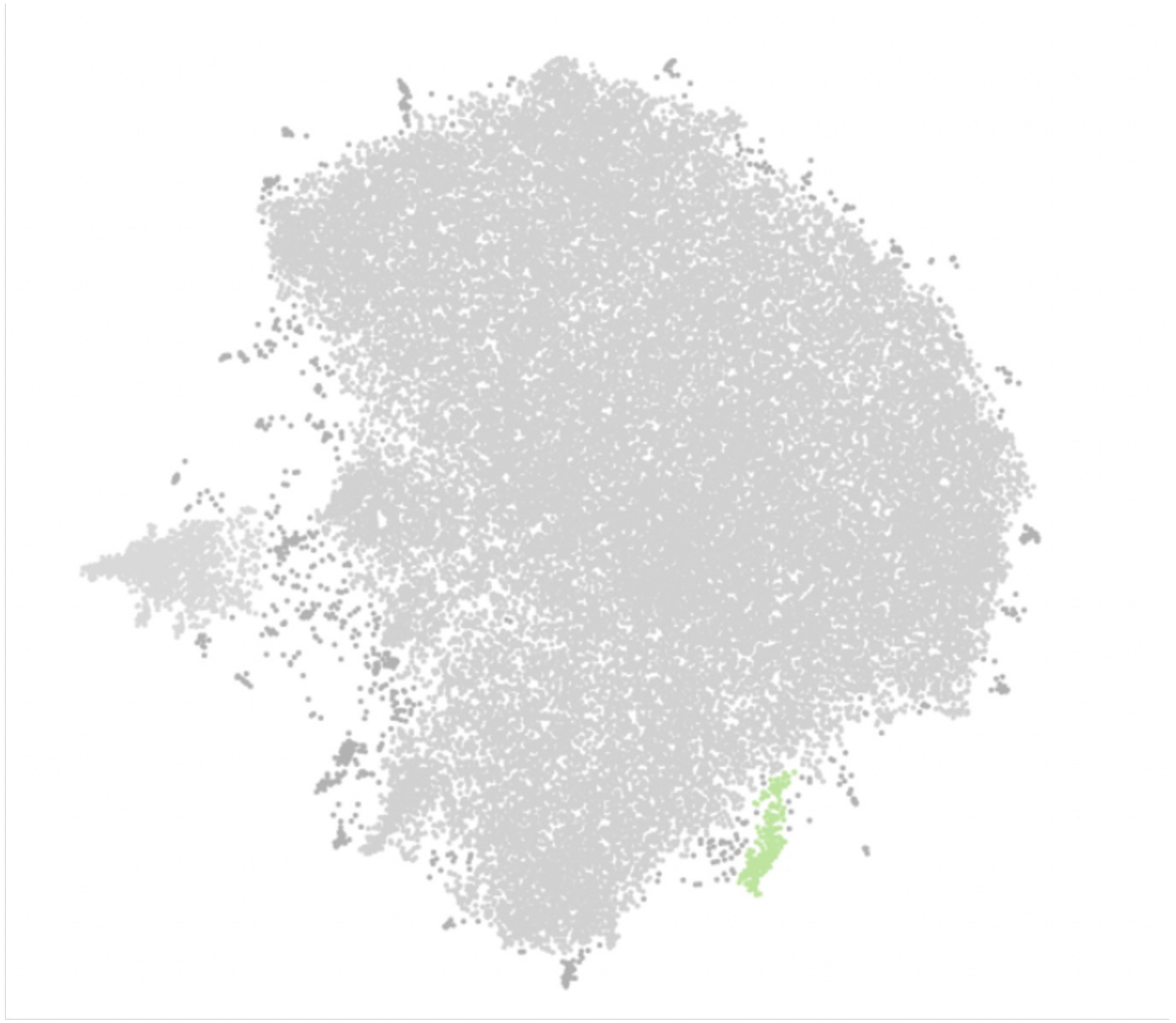




Clustering

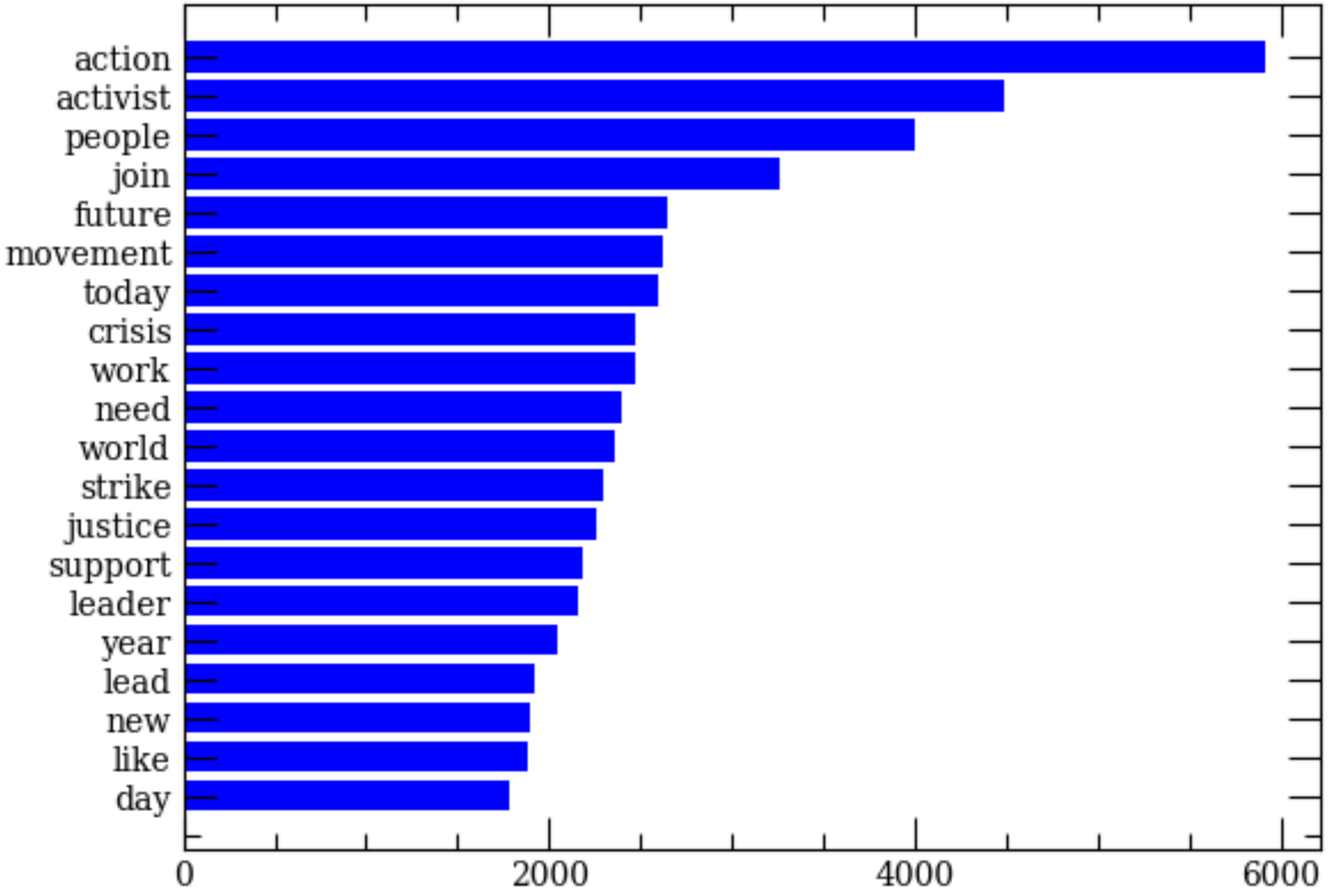
Word frequency and Cluster 1: leadership up-scaling

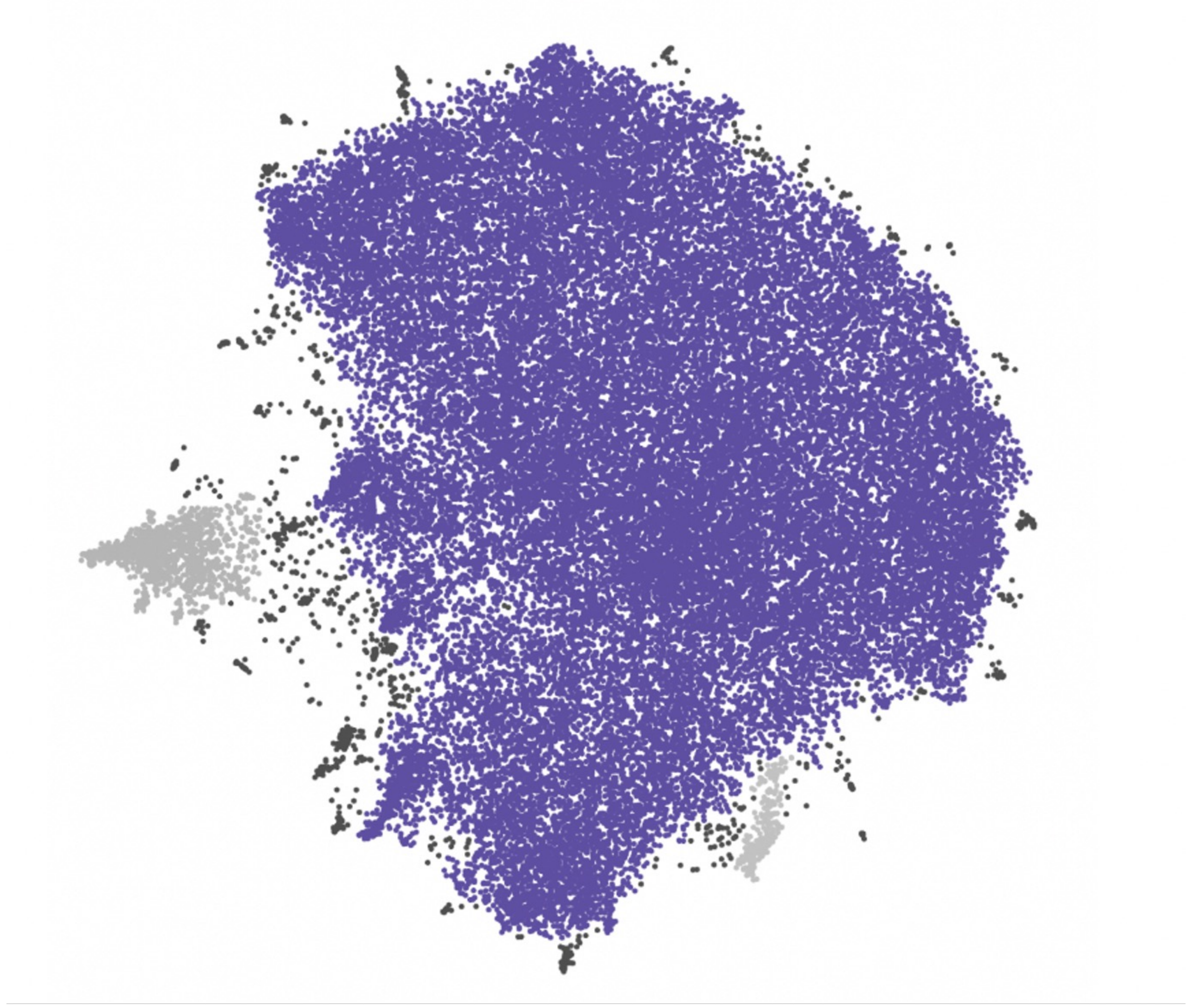




Clustering

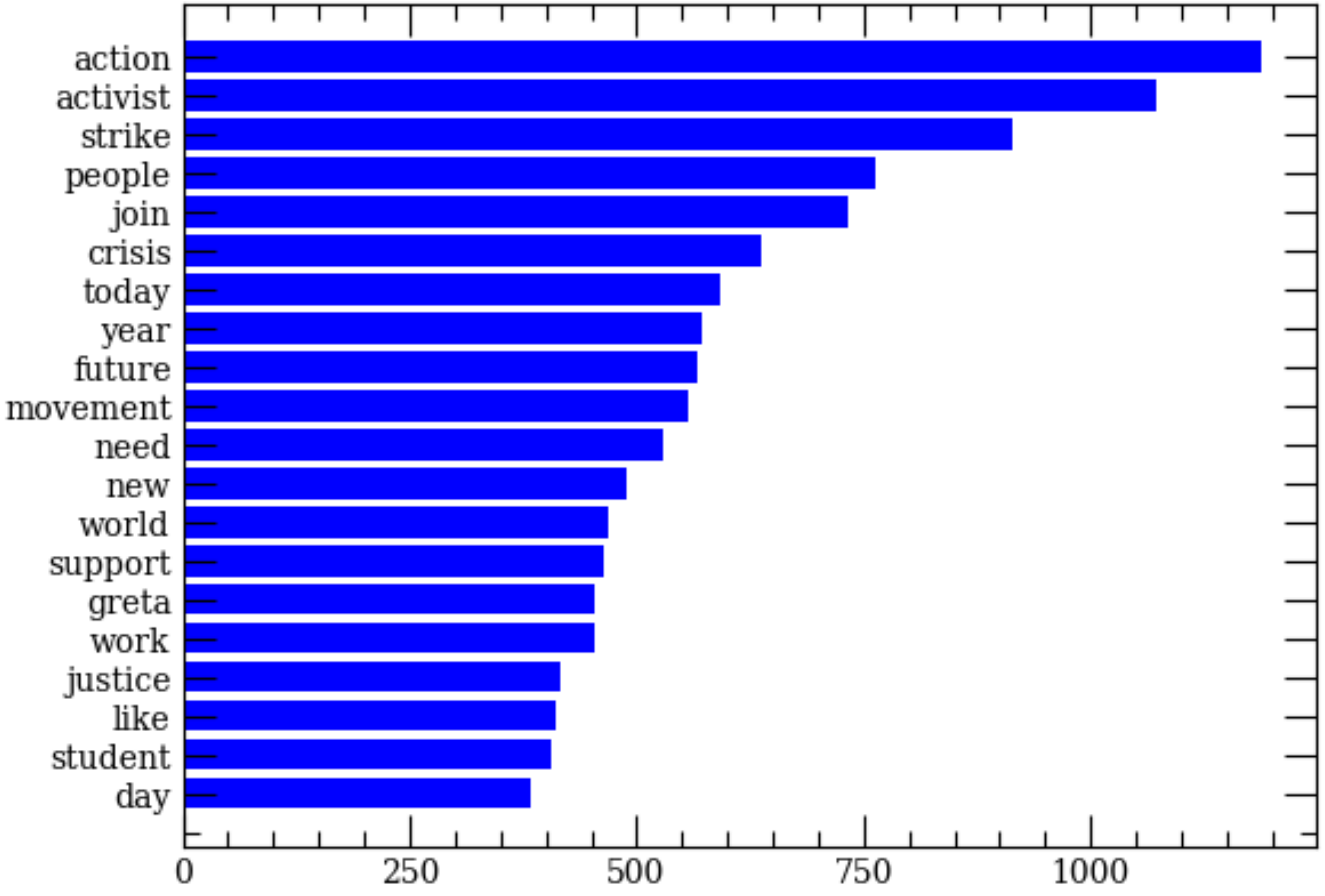
Word frequency and largest cluster

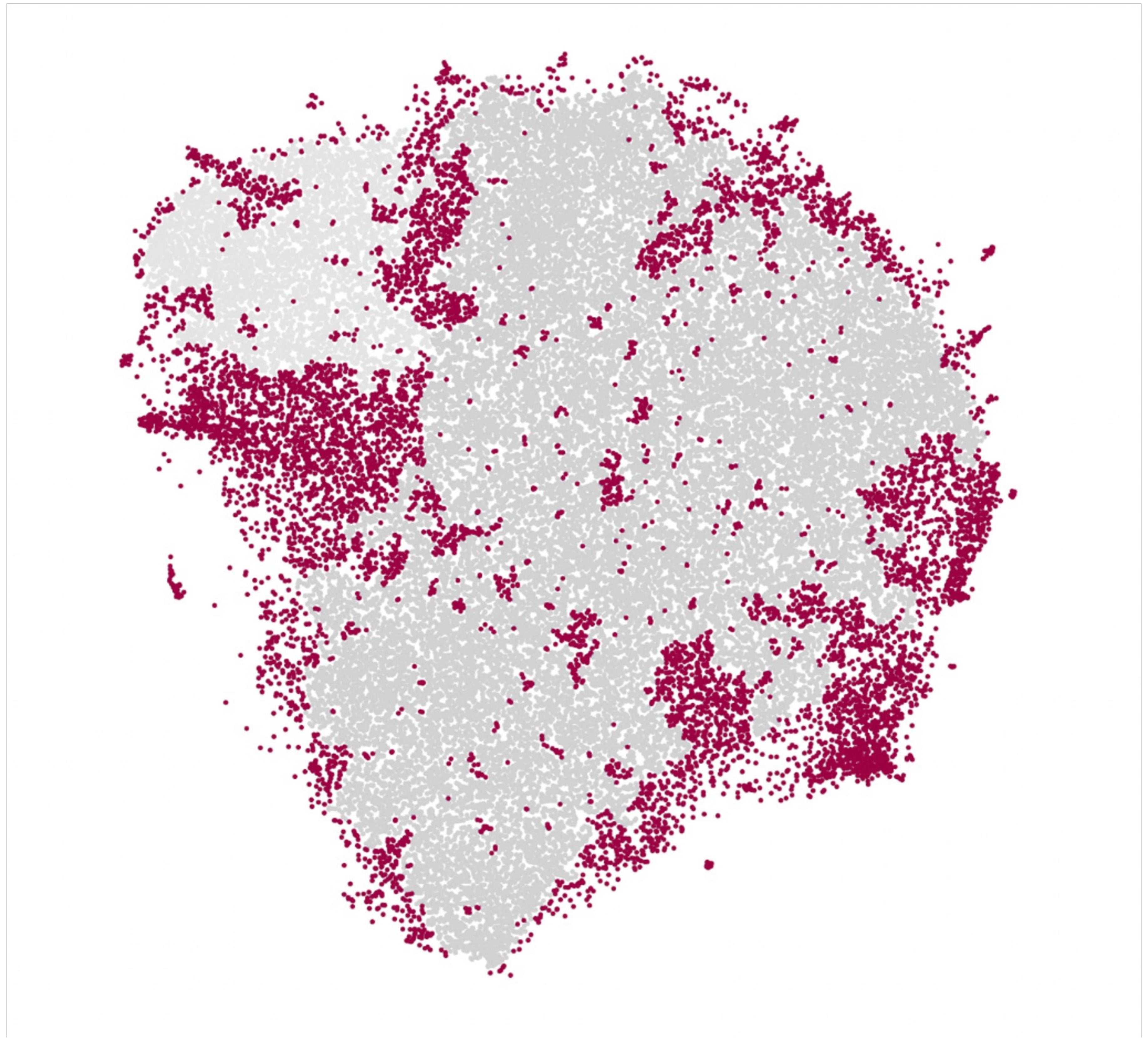




Clustering

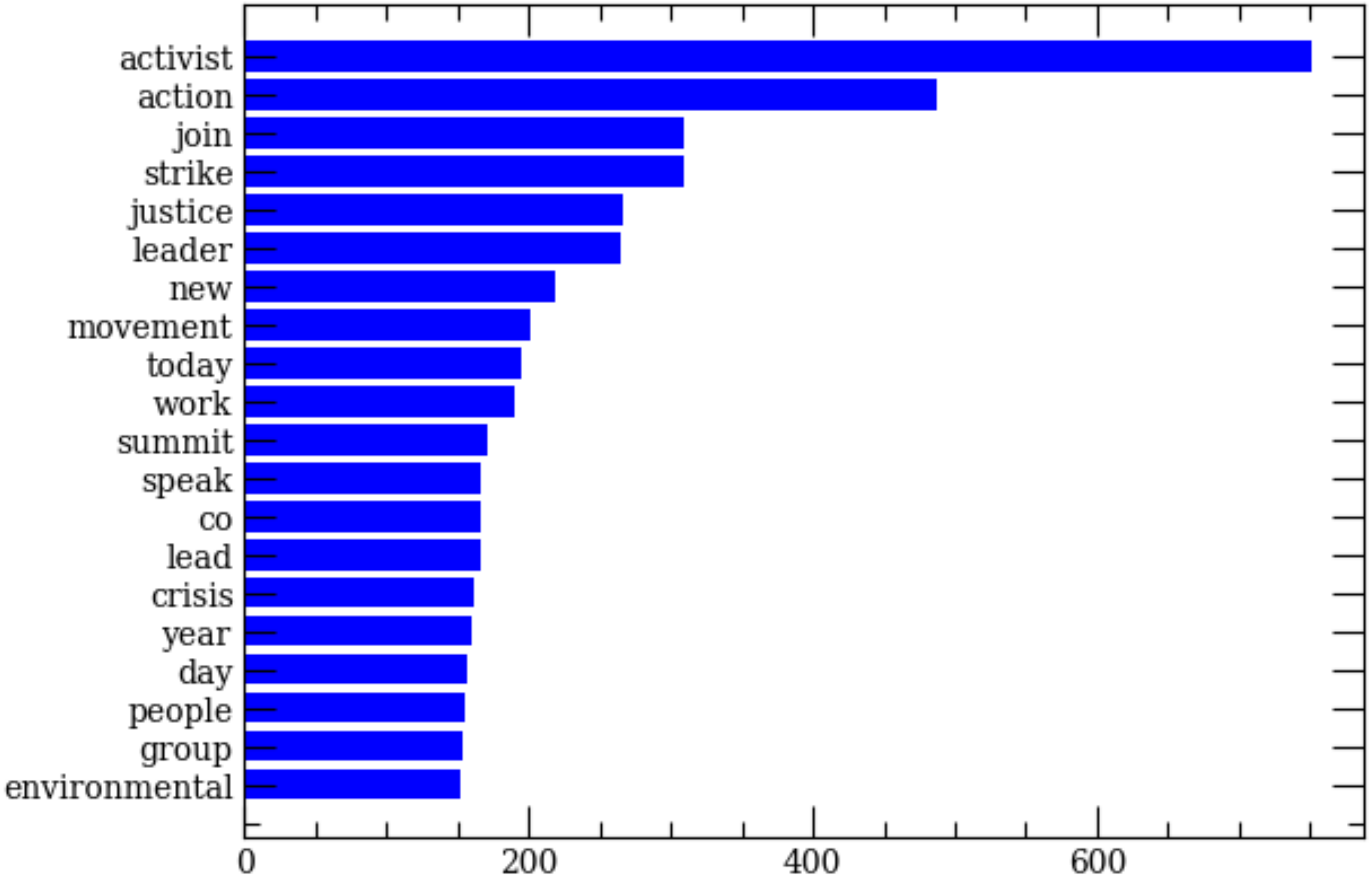
Word frequency and Outlier cluster 2

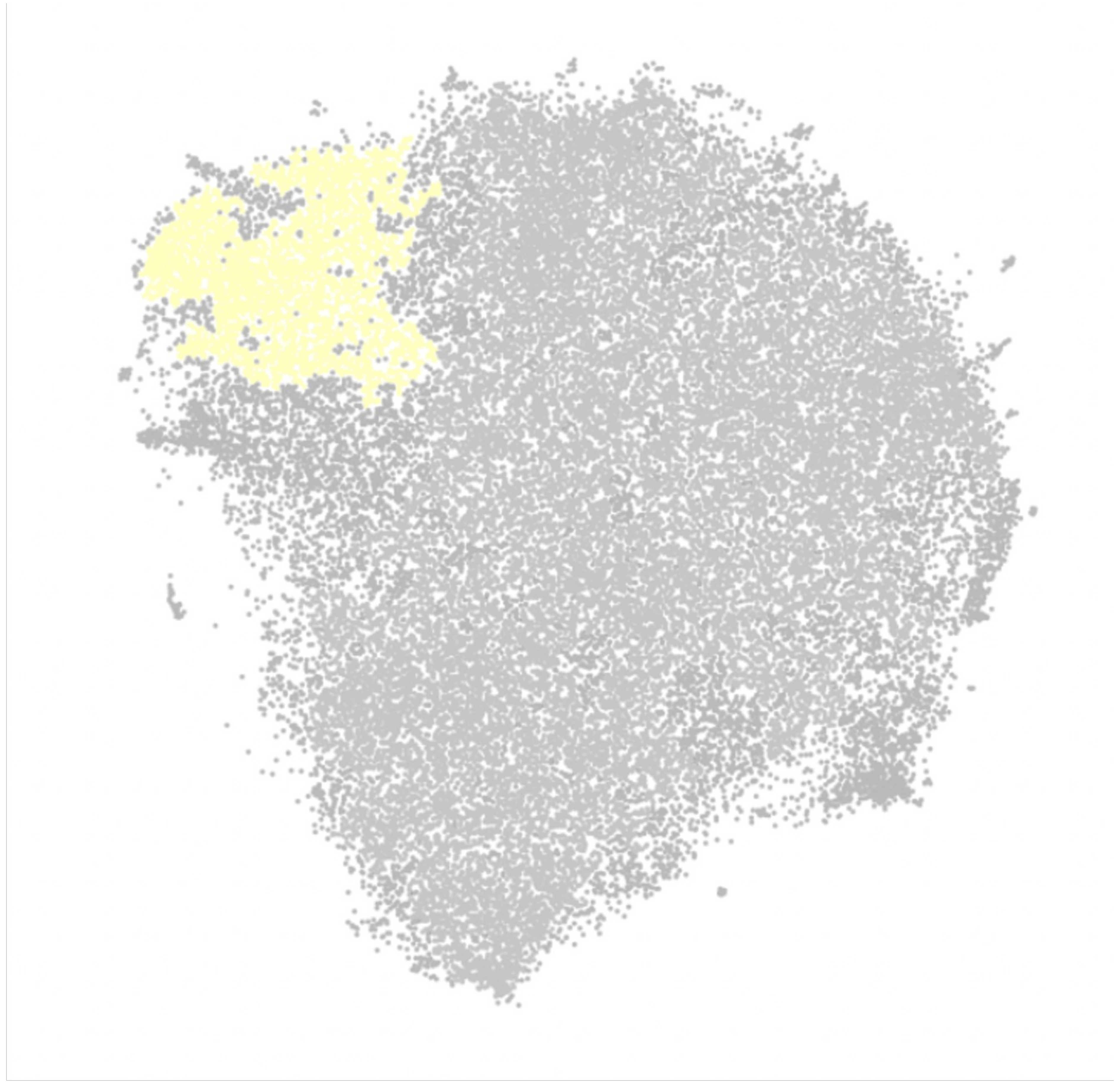




Clustering

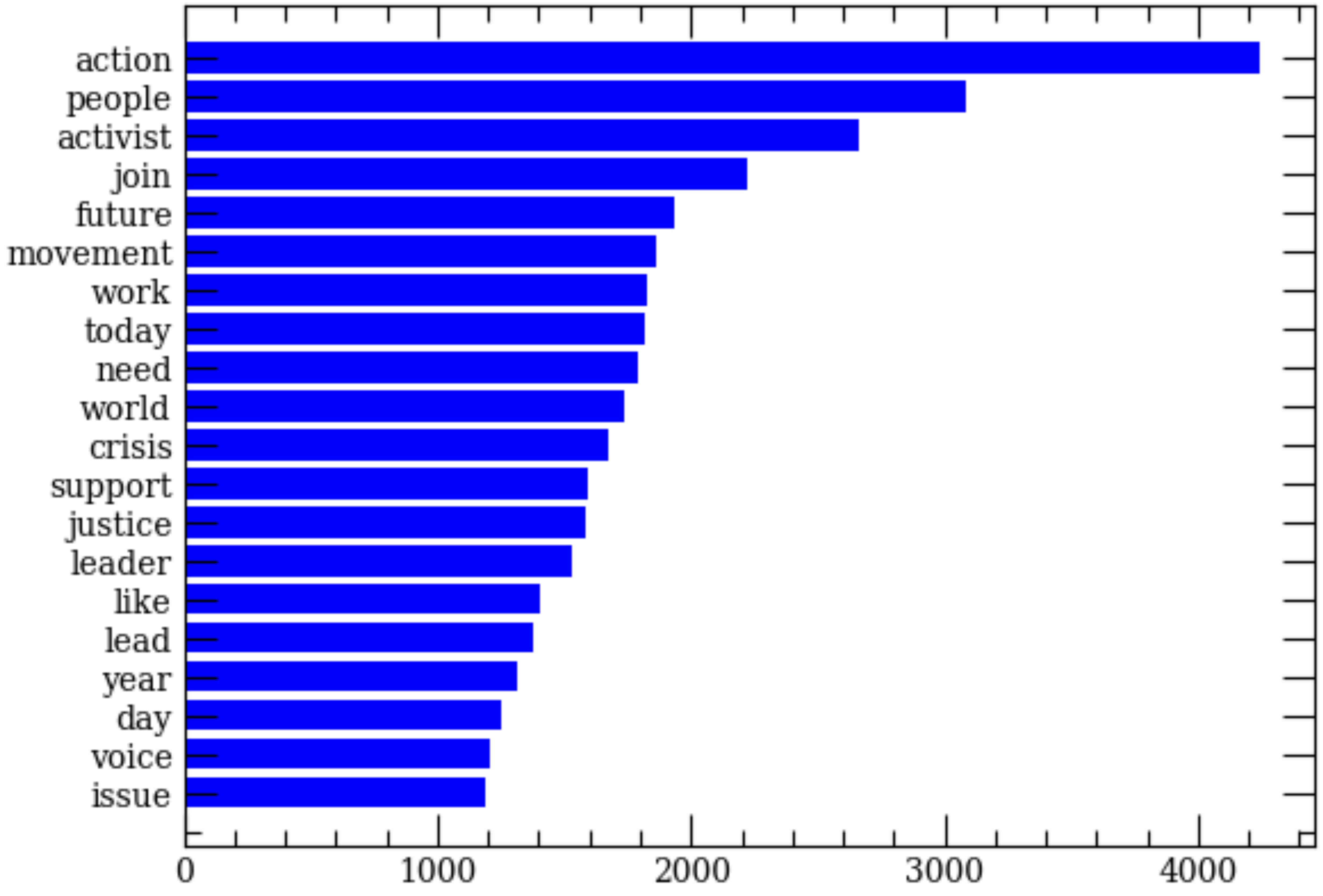
Word frequency and Cluster 3: movement events

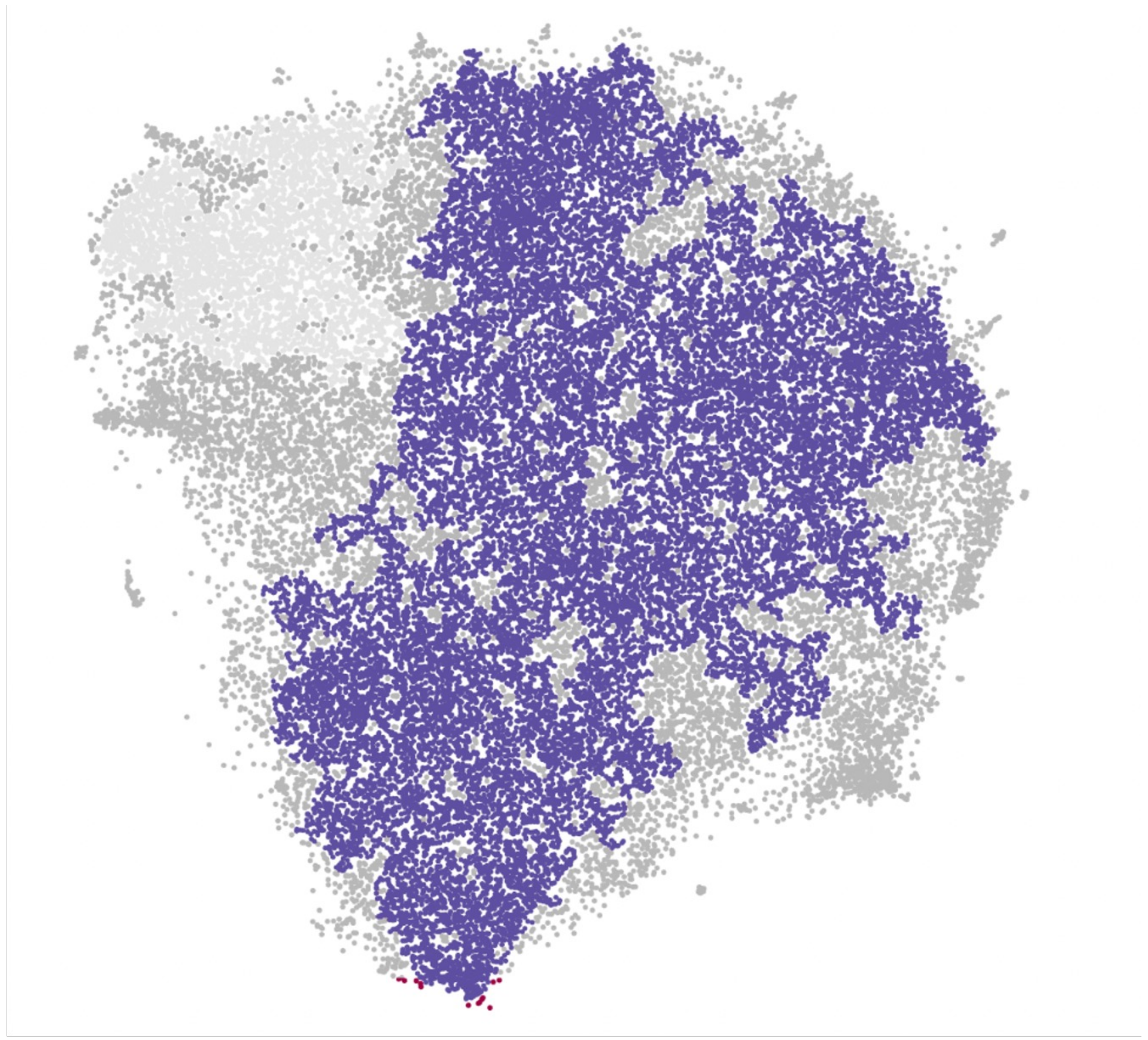




Clustering

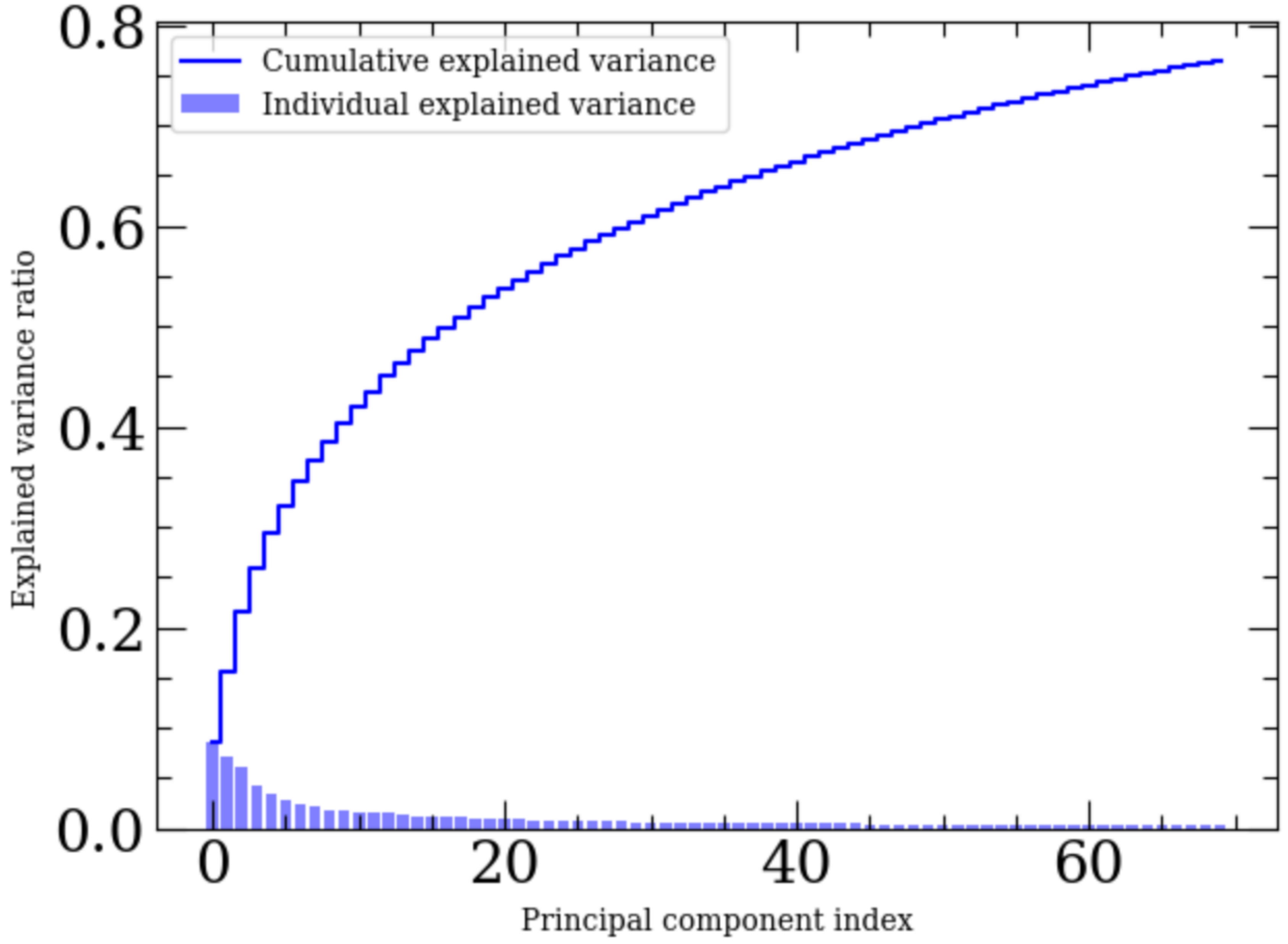
Word frequency and Cluster 4





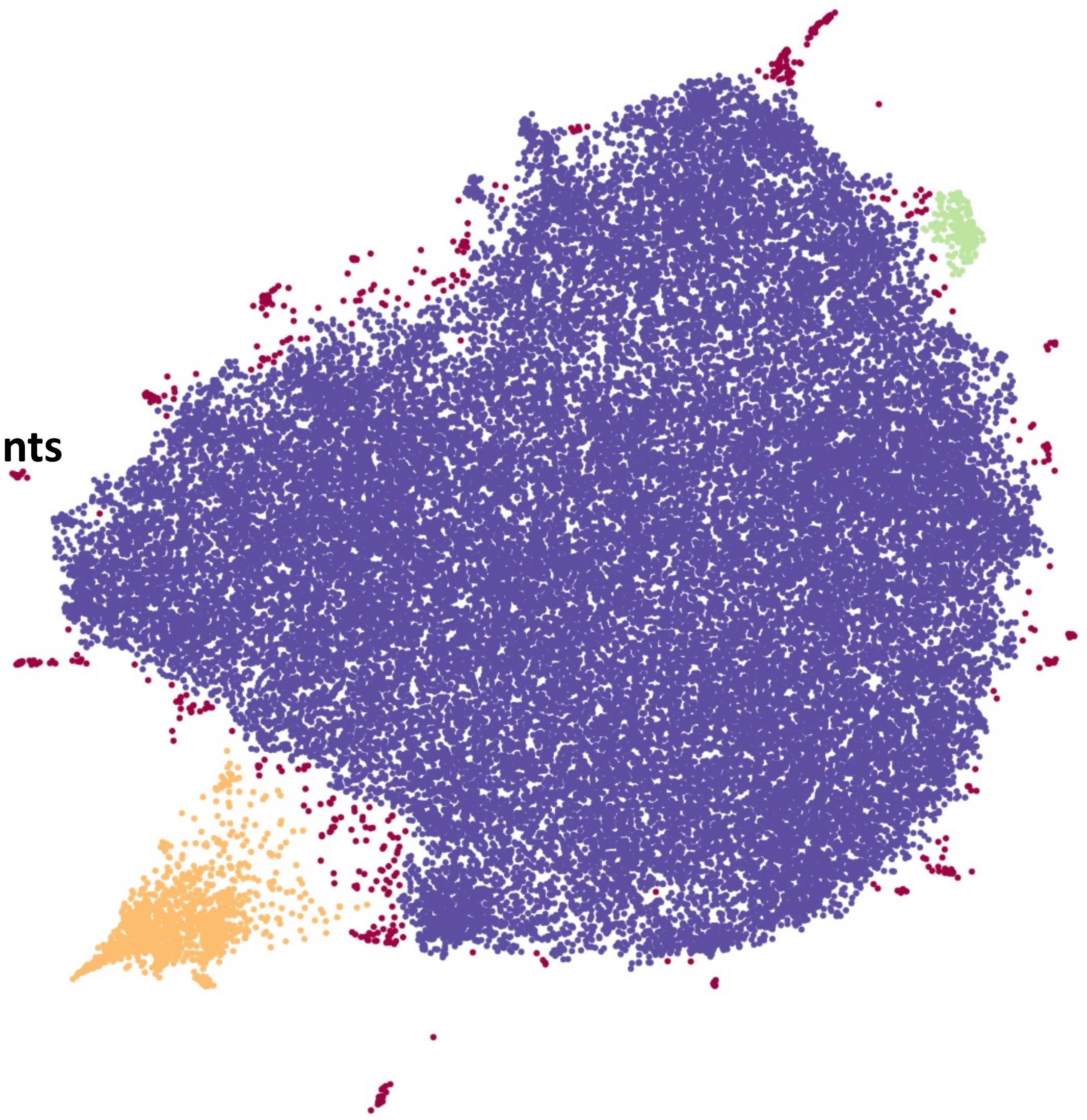
PCA + UMAP

PCA before HDBSCAN

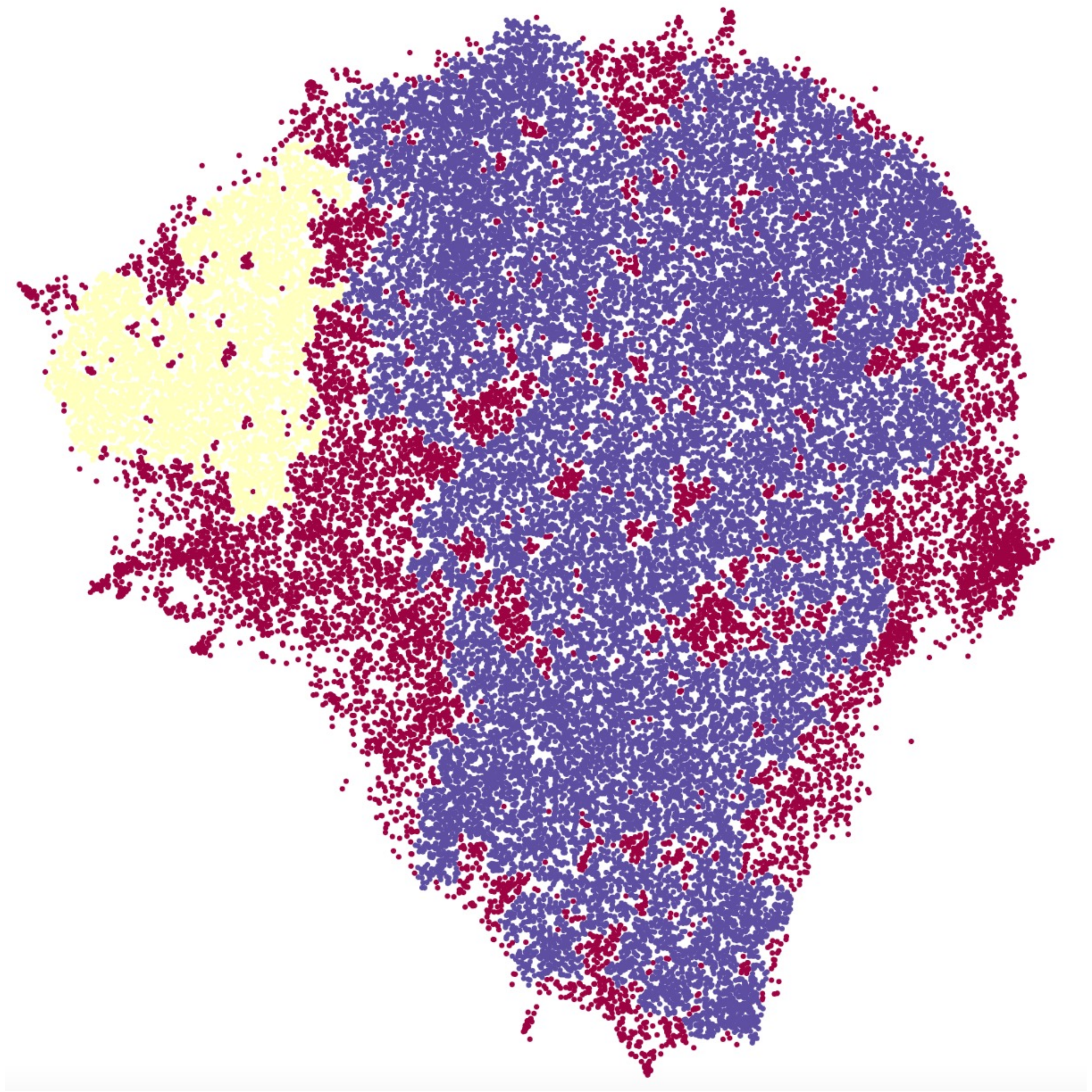


70 components

HDBSCAN



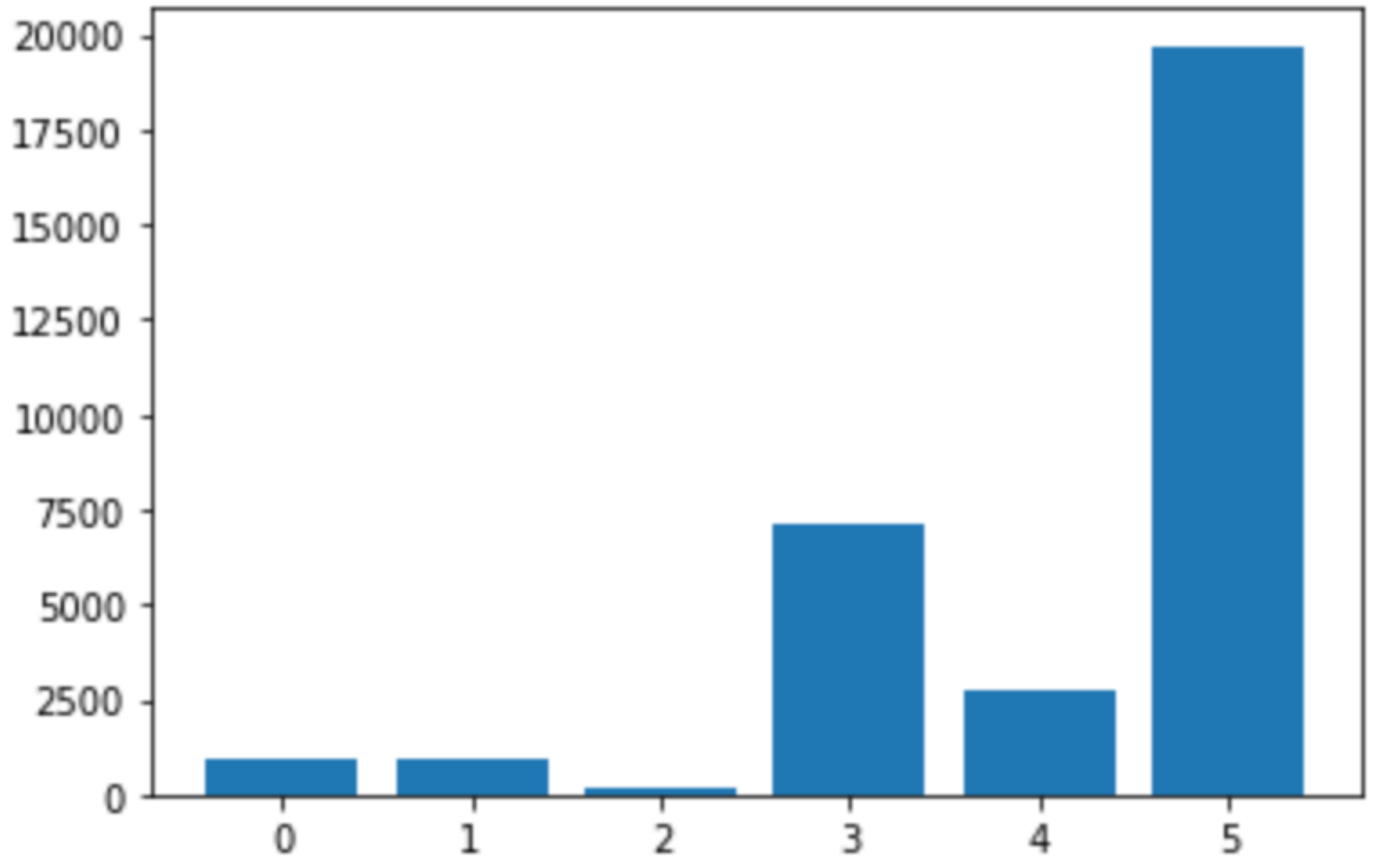
Reclustering the largest



Classification

Oversampling the imbalanced training dataset using SMOTE

number of tweets



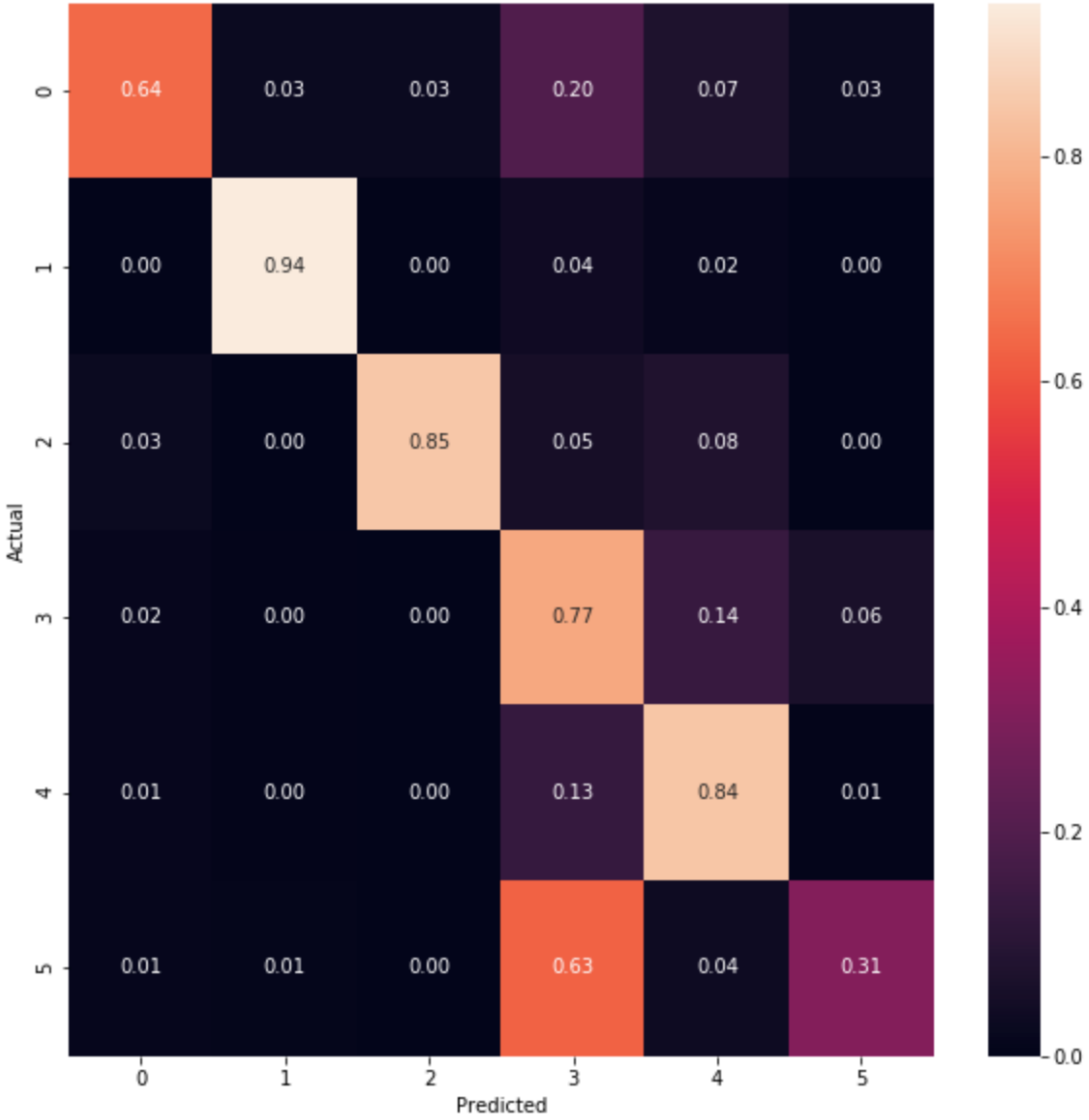
cluster labels

Classification

Algorithim: LightGBM

Hyperparameter optimizer: Optuna

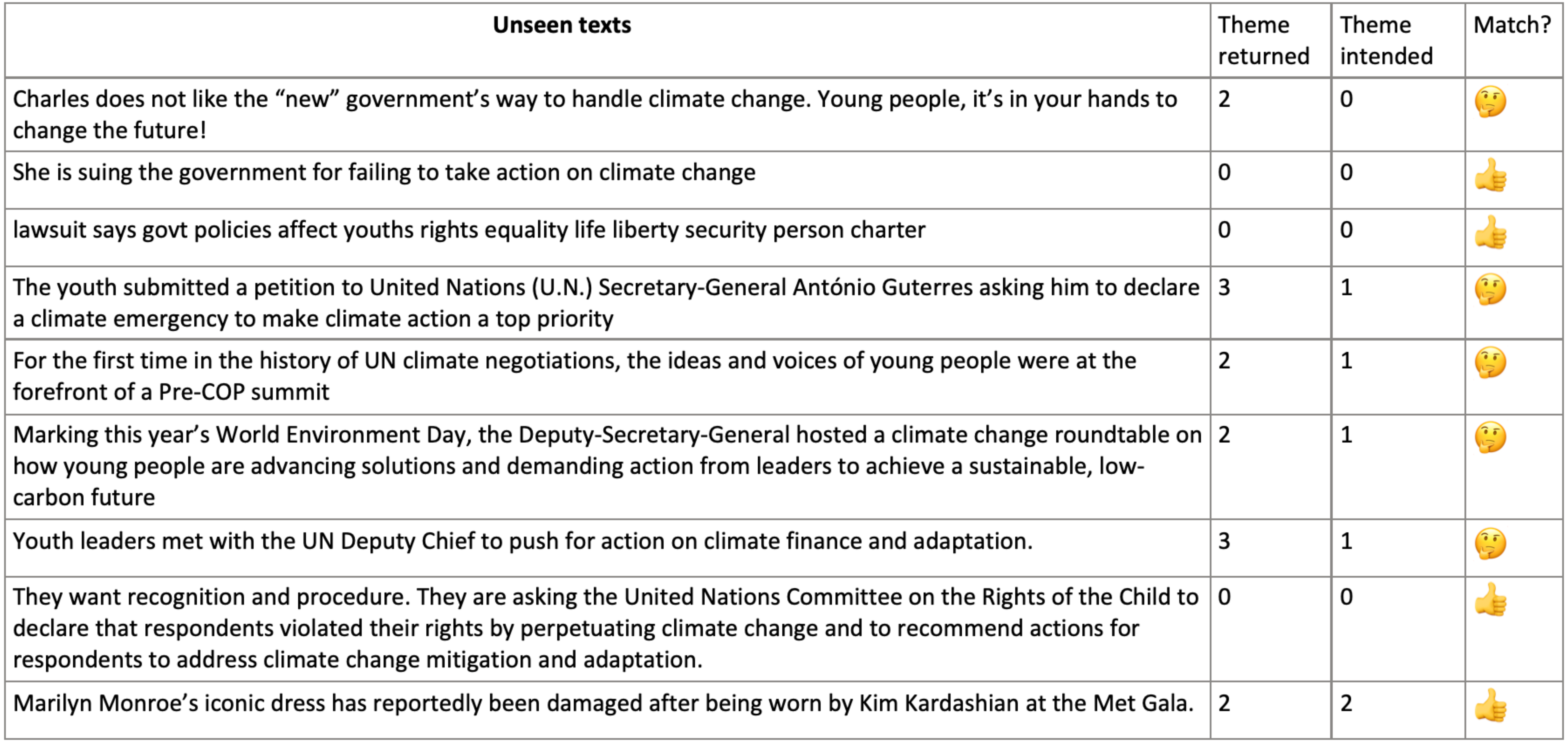
Evaluation of classifier: Confusion Matrix



Actual \ Predicted	0	1	2	3	4	5
0	0.64	0.03	0.03	0.20	0.07	0.03
1	0.00	0.94	0.00	0.04	0.02	0.00
2	0.03	0.00	0.85	0.05	0.08	0.00
3	0.02	0.00	0.00	0.77	0.14	0.06
4	0.01	0.00	0.00	0.13	0.84	0.01
5	0.01	0.01	0.00	0.63	0.04	0.31

Classification

Classifier performance with unseen texts



Unseen texts	Theme returned	Theme intended	Match?
Charles does not like the “new” government’s way to handle climate change. Young people, it’s in your hands to change the future!	2	0	🤔
She is suing the government for failing to take action on climate change	0	0	👍
lawsuit says govt policies affect youths rights equality life liberty security person charter	0	0	👍
The youth submitted a petition to United Nations (U.N.) Secretary-General António Guterres asking him to declare a climate emergency to make climate action a top priority	3	1	🤔
For the first time in the history of UN climate negotiations, the ideas and voices of young people were at the forefront of a Pre-COP summit	2	1	🤔
Marking this year’s World Environment Day, the Deputy-Secretary-General hosted a climate change roundtable on how young people are advancing solutions and demanding action from leaders to achieve a sustainable, low-carbon future	2	1	🤔
Youth leaders met with the UN Deputy Chief to push for action on climate finance and adaptation.	3	1	🤔
They want recognition and procedure. They are asking the United Nations Committee on the Rights of the Child to declare that respondents violated their rights by perpetuating climate change and to recommend actions for respondents to address climate change mitigation and adaptation.	0	0	👍
Marilyn Monroe’s iconic dress has reportedly been damaged after being worn by Kim Kardashian at the Met Gala.	2	2	👍

Conclusion

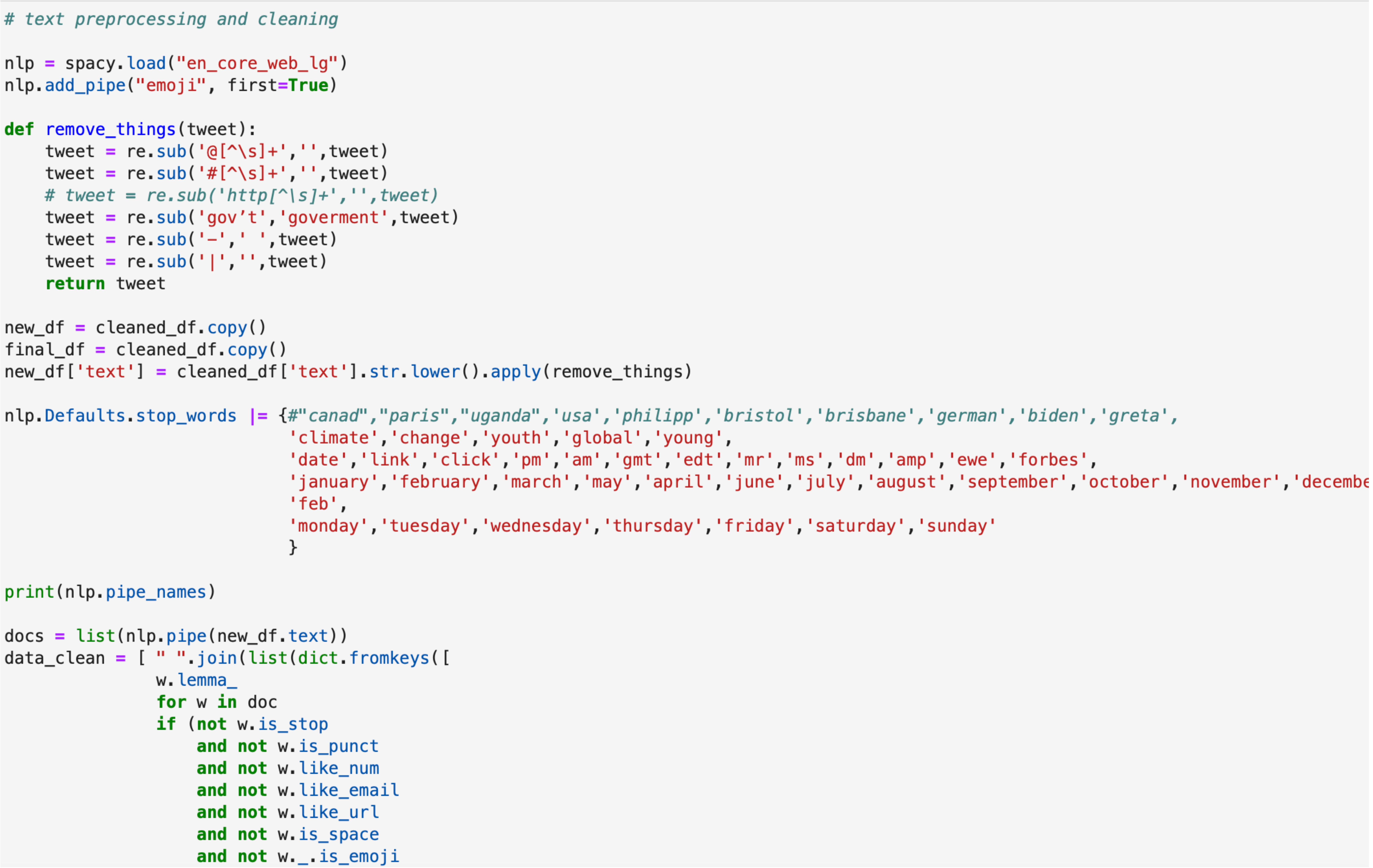
- The classifier only partially succeeded in classifying the unseen texts
- The latent themes from twitter are detected and visualised
- However, the classifier is based on the reliability of clustering, whereas changes in clustering output happened when rerunning the codes
- Unsupervised learning and the vectorised text as input data limits the ways to evaluate the model accuracy
- Could word frequency be the major factor of the resulting clustering here?

Thank you

include
issue
start
learn
voice
environment
activist
thank
need
country
call
say
live
hear
group
vote
find
community
movement
crisi
register
current
political
engage
support
solution
fight
discuss
week
power
plan
talk
help
future
government
meet
action
join
speak
education
year
lead
new
great
impact
project
activism
inspire
right
summit
generation
good
know
child
student
justice
today
focus
time
bring
policy
people
school
hope
think
read
work
let
look
listen
change
day
create
go
world
leader
want
come
way
strike
event
share
environmental
check
advocate
happen

Appendix

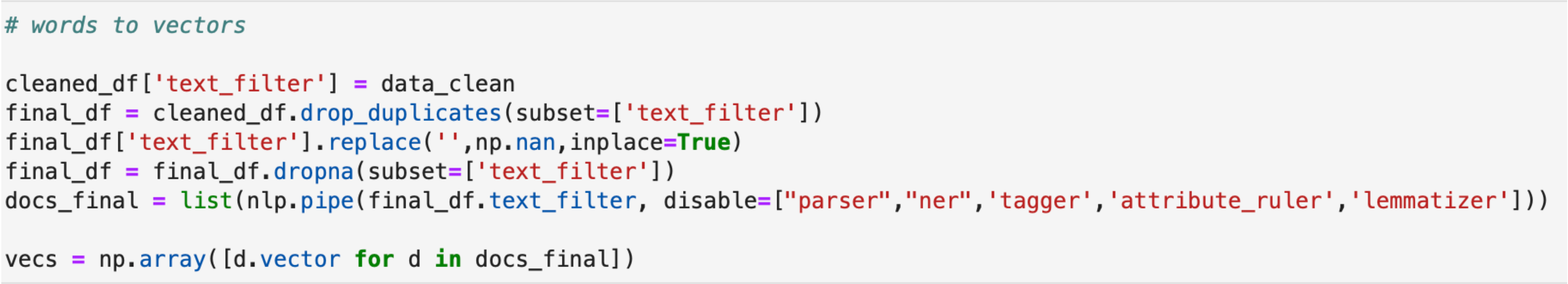
Codes & descriptions



```
# text preprocessing and cleaning

nlp = spacy.load("en_core_web_lg")
nlp.add_pipe("emoji", first=True)

def remove_things(tweet):
    tweet = re.sub('@[^\s]+','',tweet)
    tweet = re.sub('#[^\s]+','',tweet)
    # tweet = re.sub('http[^\s]+','',tweet)
    tweet = re.sub('gov’t','goverment',tweet)
    tweet = re.sub('–',' ',tweet)
    tweet = re.sub('|','',tweet)
    return tweet

new_df = cleaned_df.copy()
final_df = cleaned_df.copy()
new_df['text'] = cleaned_df['text'].str.lower().apply(remove_things)

nlp.Defaults.stop_words |= {#"canad","paris","uganda",'usa','philipp','bristol','brisbane','german','biden','greta',
                            'climate','change','youth','global','young',
                            'date','link','click','pm','am','gmt','edt','mr','ms','dm','amp','ewe','forbes',
                            'january','february','march','may','april','june','july','august','september','october','november','decembe
                            'feb',
                            'monday','tuesday','wednesday','thursday','friday','saturday','sunday'
                            }

print(nlp.pipe_names)

docs = list(nlp.pipe(new_df.text))
data_clean = [ " ".join(list(dict.fromkeys([
               w.lemma_
               for w in doc
               if (not w.is_stop
                   and not w.is_punct
                   and not w.like_num
                   and not w.like_email
                   and not w.like_url
                   and not w.is_space
                   and not w._.is_emoji
```

Appendix

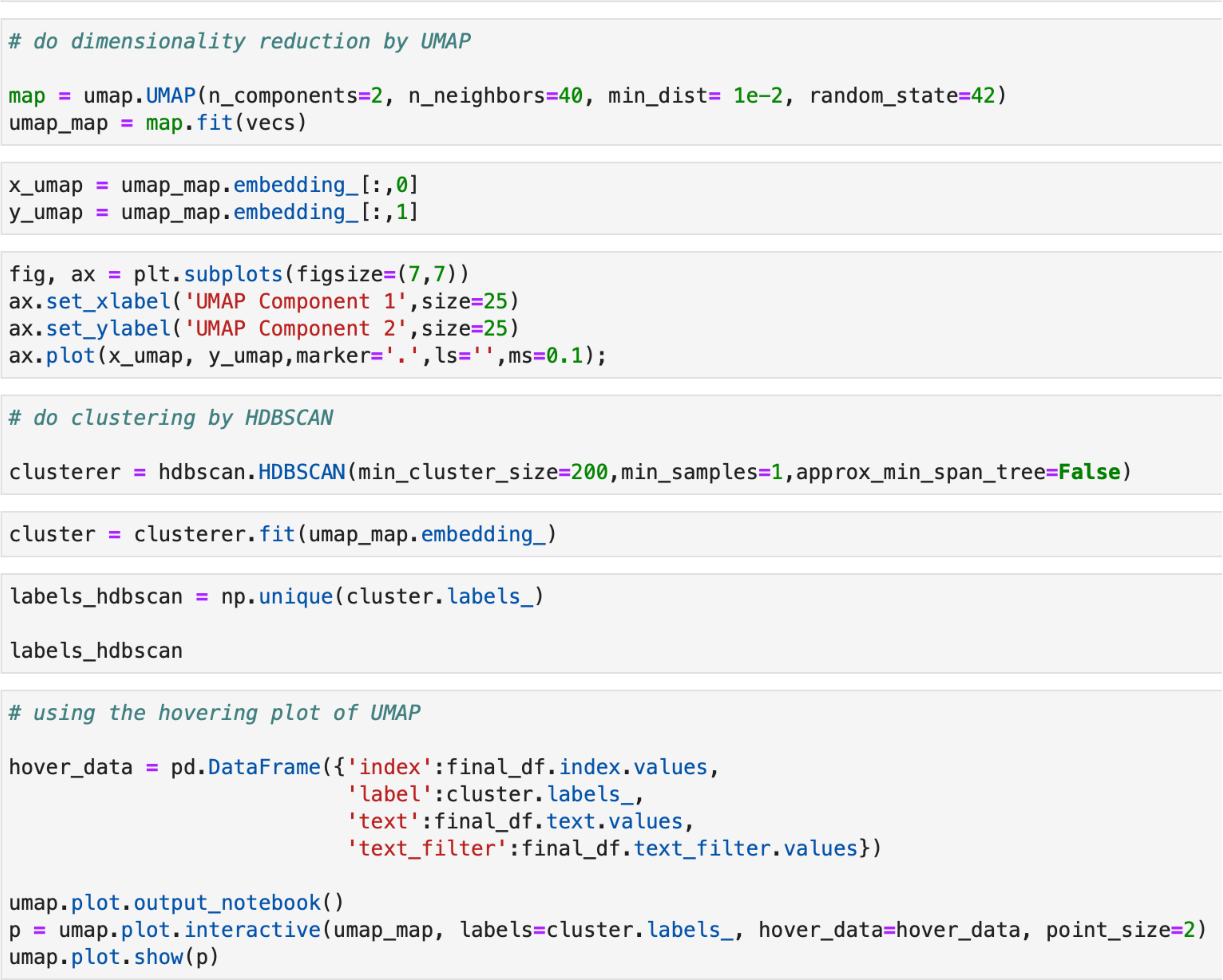
Codes & descriptions



```
# words to vectors

cleaned_df['text_filter'] = data_clean
final_df = cleaned_df.drop_duplicates(subset=['text_filter'])
final_df['text_filter'].replace('',np.nan,inplace=True)
final_df = final_df.dropna(subset=['text_filter'])
docs_final = list(nlp.pipe(final_df.text_filter, disable=["parser","ner",'tagger','attribute_ruler','lemmatizer']))

vecs = np.array([d.vector for d in docs_final])
```

Appendix

Codes & descriptions



```
# do dimensionality reduction by UMAP

map = umap.UMAP(n_components=2, n_neighbors=40, min_dist= 1e-2, random_state=42)
umap_map = map.fit(vecs)
```

```
x_umap = umap_map.embedding_[:,0]
y_umap = umap_map.embedding_[:,1]
```

```
fig, ax = plt.subplots(figsize=(7,7))
ax.set_xlabel('UMAP Component 1',size=25)
ax.set_ylabel('UMAP Component 2',size=25)
ax.plot(x_umap, y_umap,marker='.',ls='',ms=0.1);
```

```
# do clustering by HDBSCAN

clusterer = hdbscan.HDBSCAN(min_cluster_size=200,min_samples=1,approx_min_span_tree=False)
```

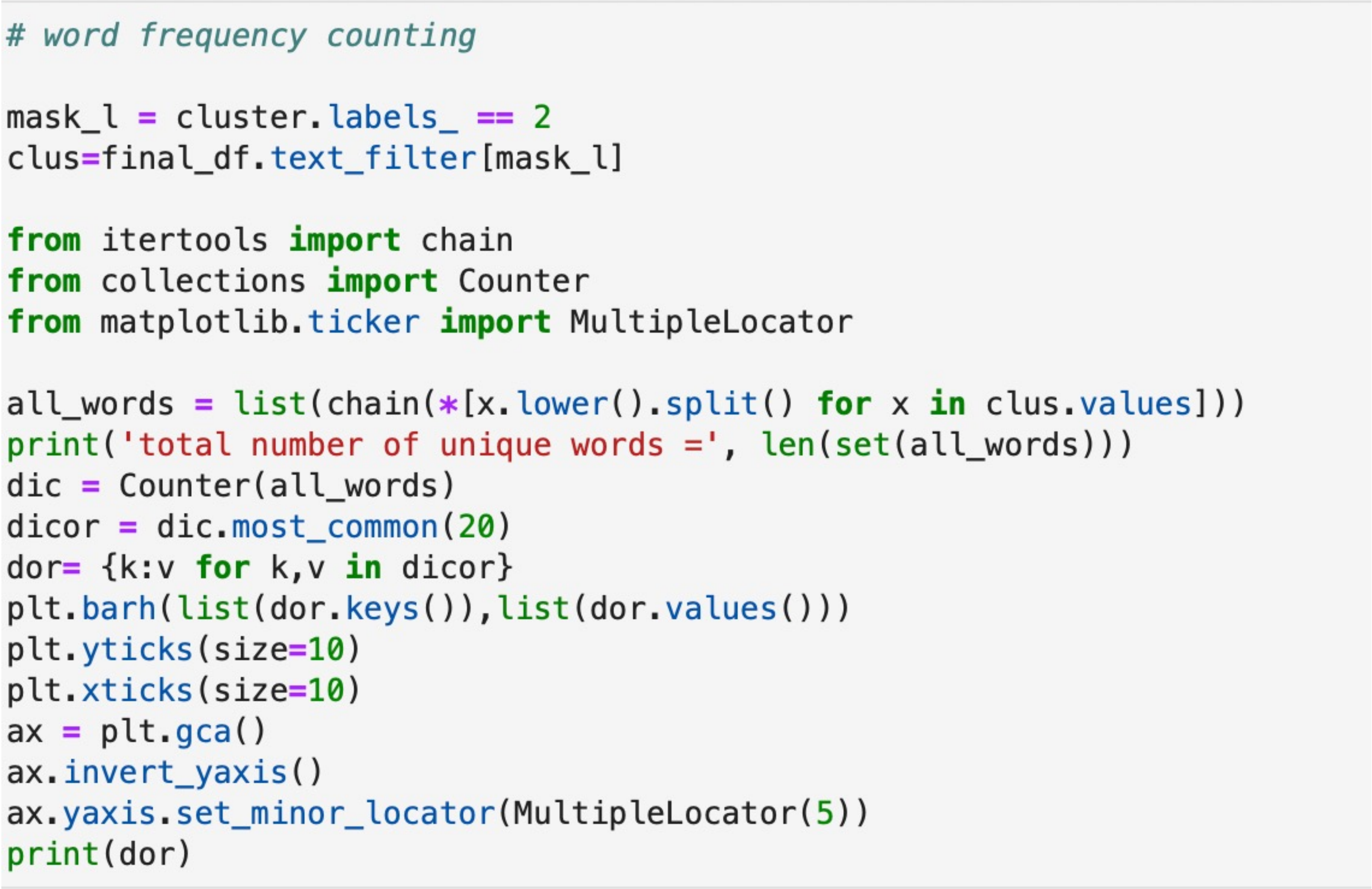
```
cluster = clusterer.fit(umap_map.embedding_)
```

```
labels_hdbscan = np.unique(cluster.labels_)

labels_hdbscan
```

```
# using the hovering plot of UMAP

hover_data = pd.DataFrame({'index':final_df.index.values,
                           'label':cluster.labels_,
                           'text':final_df.text.values,
                           'text_filter':final_df.text_filter.values})

umap.plot.output_notebook()
p = umap.plot.interactive(umap_map, labels=cluster.labels_, hover_data=hover_data, point_size=2)
umap.plot.show(p)
```

Appendix

Codes & descriptions

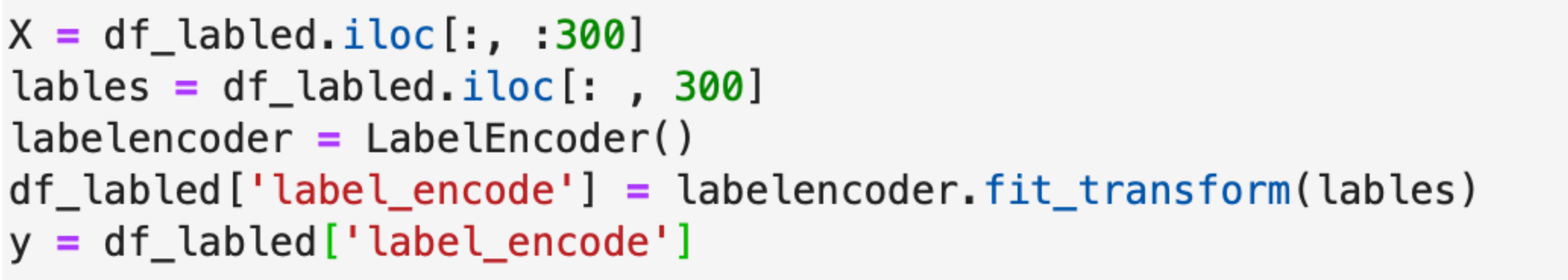


```
# word frequency counting

mask_l = cluster.labels_ == 2
clus=final_df.text_filter[mask_l]

from itertools import chain
from collections import Counter
from matplotlib.ticker import MultipleLocator

all_words = list(chain(*[x.lower().split() for x in clus.values]))
print('total number of unique words =', len(set(all_words)))
dic = Counter(all_words)
dicor = dic.most_common(20)
dor= {k:v for k,v in dicor}
plt.barh(list(dor.keys()),list(dor.values()))
plt.yticks(size=10)
plt.xticks(size=10)
ax = plt.gca()
ax.invert_yaxis()
ax.yaxis.set_minor_locator(MultipleLocator(5))
print(dor)
```

Appendix

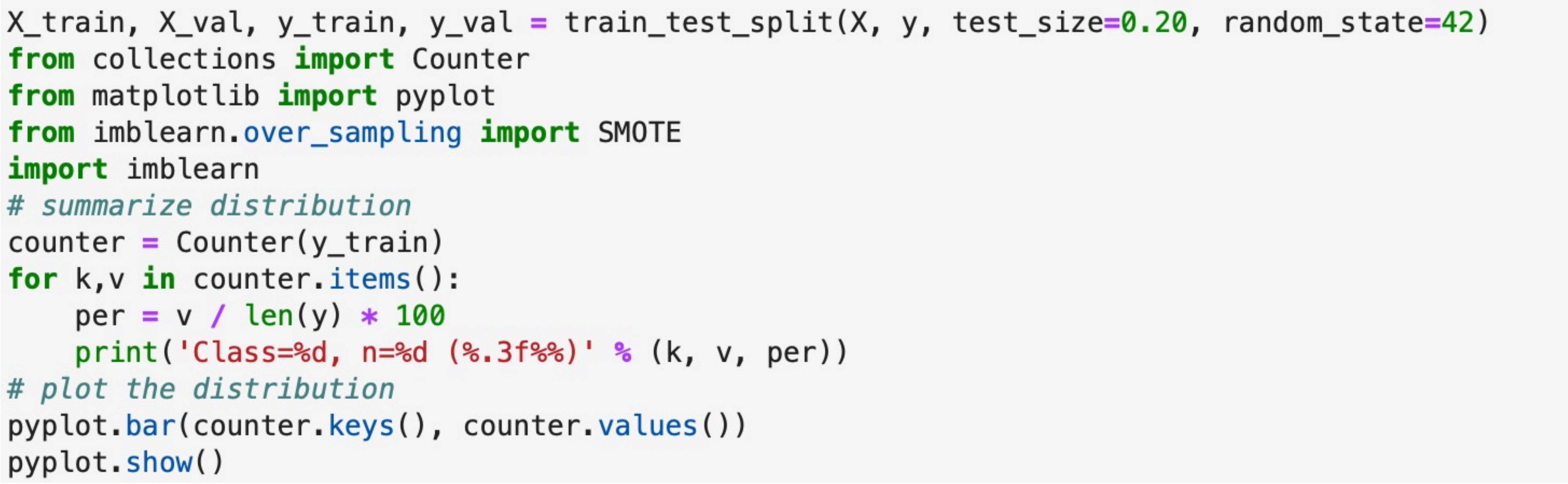
Codes & descriptions

Label encode

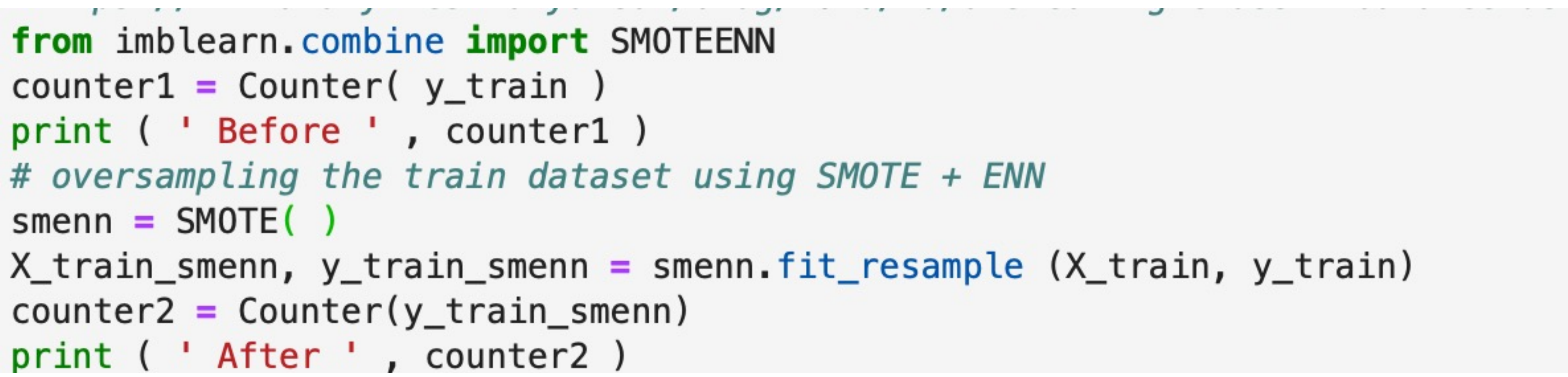


```
X = df_labled.iloc[:, :300]
lables = df_labled.iloc[: , 300]
labelencoder = LabelEncoder()
df_labled['label_encode'] = labelencoder.fit_transform(lables)
y = df_labled['label_encode']
```

SMOTE



```
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.20, random_state=42)
from collections import Counter
from matplotlib import pyplot
from imblearn.over_sampling import SMOTE
import imblearn
# summarize distribution
counter = Counter(y_train)
for k,v in counter.items():
    per = v / len(y) * 100
    print('Class=%d, n=%d (%.3f%%)' % (k, v, per))
# plot the distribution
pyplot.bar(counter.keys(), counter.values())
pyplot.show()
```

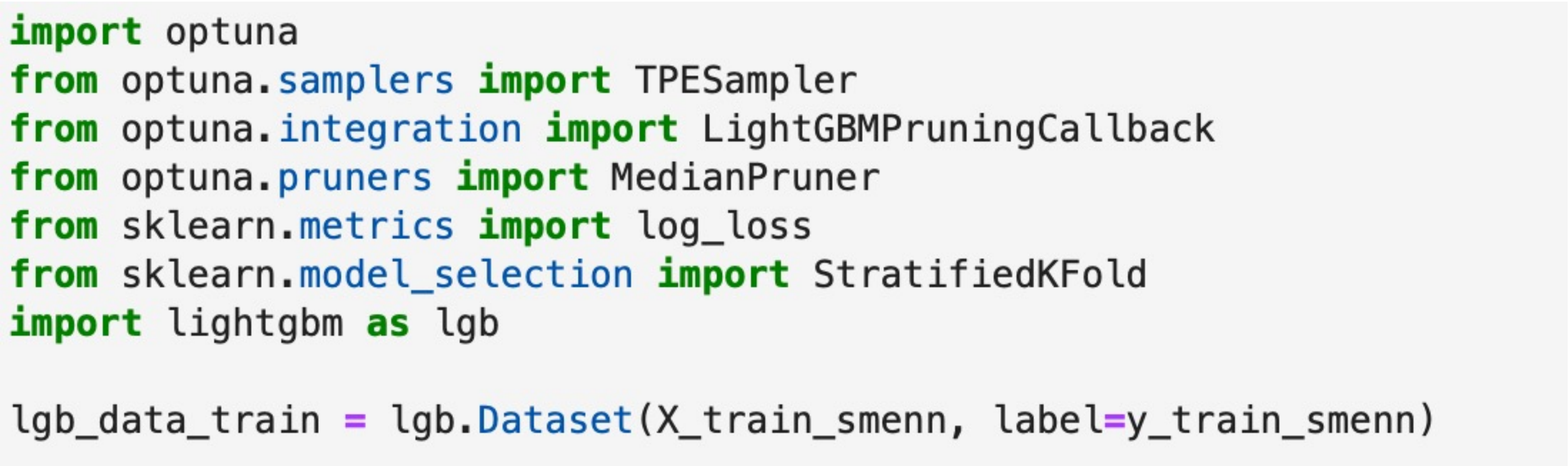


```
from imblearn.combine import SMOTEENN
counter1 = Counter( y_train )
print ( ' Before ' , counter1 )
# oversampling the train dataset using SMOTE + ENN
smenn = SMOTE( )
X_train_smenn, y_train_smenn = smenn.fit_resample (X_train, y_train)
counter2 = Counter(y_train_smenn)
print ( ' After ' , counter2 )
```

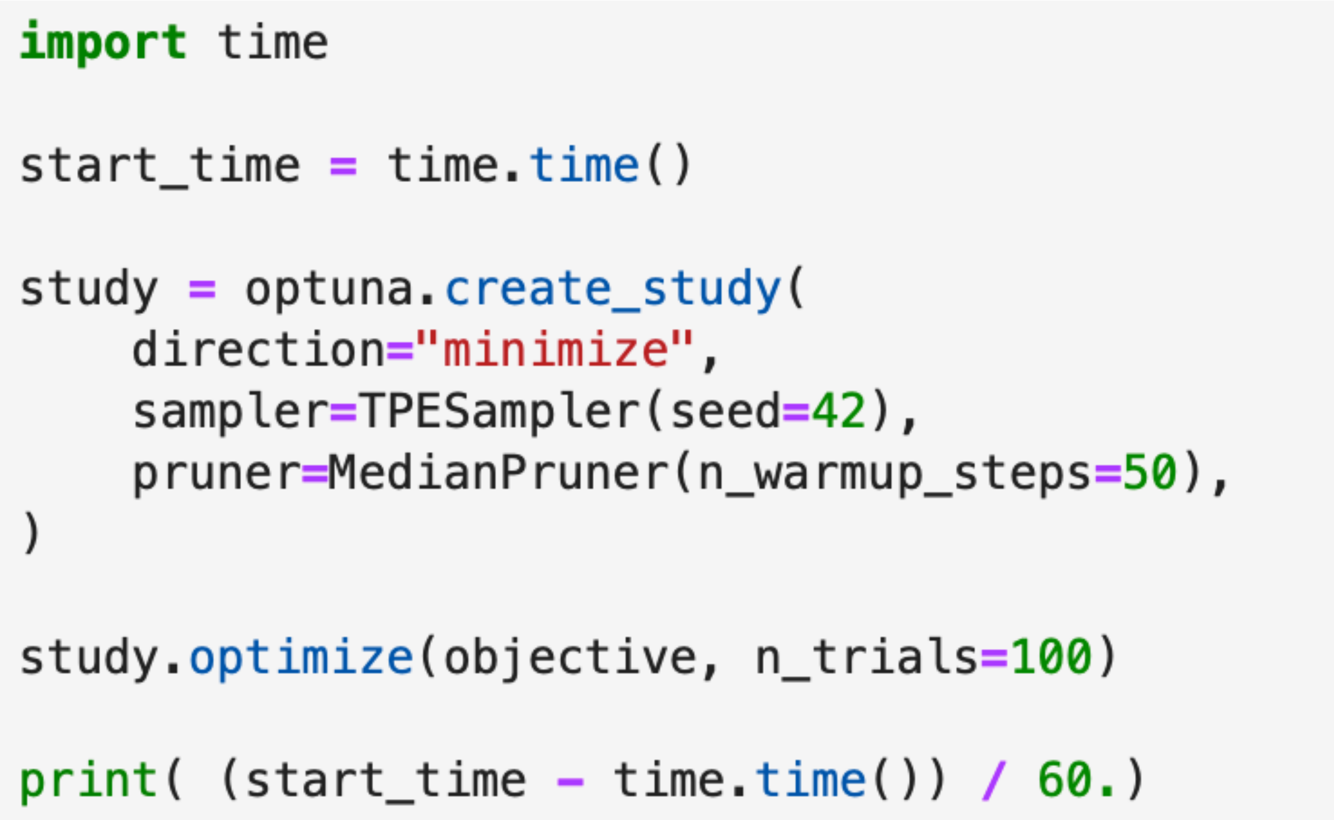
Appendix

Codes & descriptions

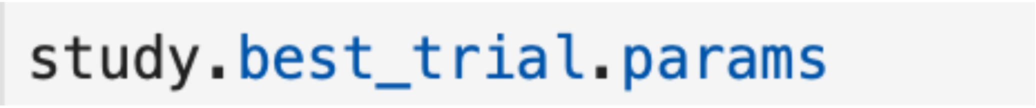
Optuna optimizer



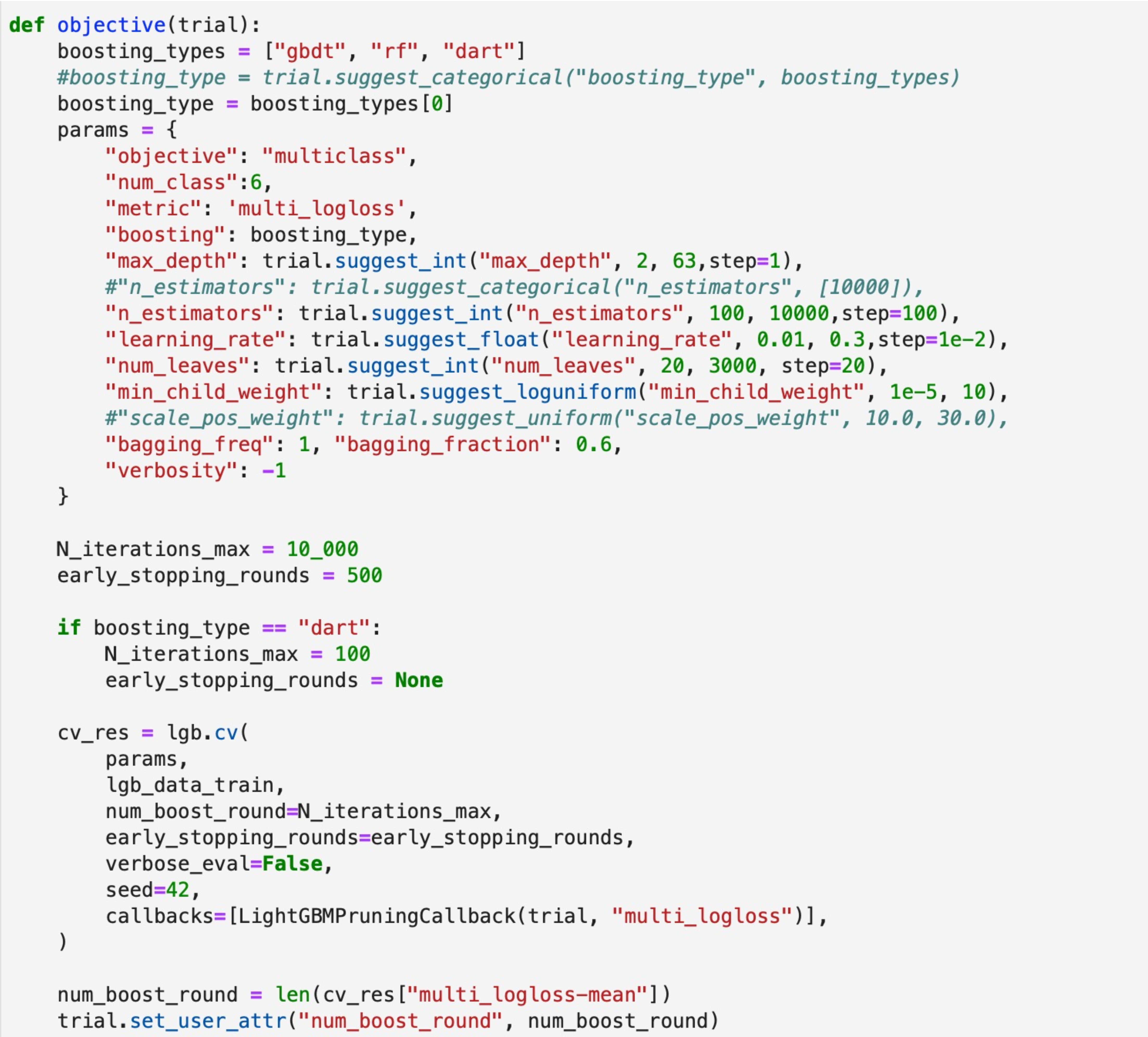
```
import optuna
from optuna.samplers import TPESampler
from optuna.integration import LightGBMPruningCallback
from optuna.pruners import MedianPruner
from sklearn.metrics import log_loss
from sklearn.model_selection import StratifiedKFold
import lightgbm as lgb

lgb_data_train = lgb.Dataset(X_train_smenn, label=y_train_smenn)
```



```
import time

start_time = time.time()

study = optuna.create_study(
    direction="minimize",
    sampler=TPESampler(seed=42),
    pruner=MedianPruner(n_warmup_steps=50),
)

study.optimize(objective, n_trials=100)

print( (start_time - time.time()) / 60.)
```



```
study.best_trial.params
```

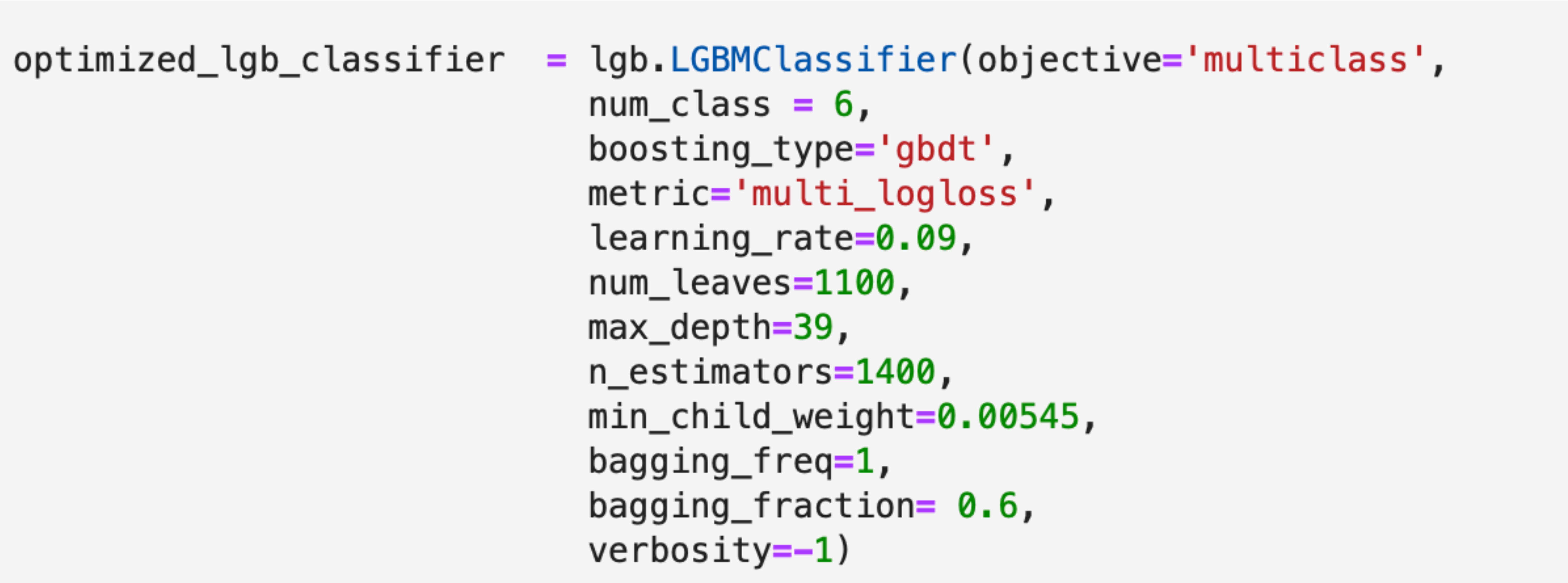


```
def objective(trial):
    boosting_types = ["gbdt", "rf", "dart"]
    #boosting_type = trial.suggest_categorical("boosting_type", boosting_types)
    boosting_type = boosting_types[0]
    params = {
        "objective": "multiclass",
        "num_class":6,
        "metric": 'multi_logloss',
        "boosting": boosting_type,
        "max_depth": trial.suggest_int("max_depth", 2, 63,step=1),
        #"n_estimators": trial.suggest_categorical("n_estimators", [10000]),
        "n_estimators": trial.suggest_int("n_estimators", 100, 10000,step=100),
        "learning_rate": trial.suggest_float("learning_rate", 0.01, 0.3,step=1e-2),
        "num_leaves": trial.suggest_int("num_leaves", 20, 3000, step=20),
        "min_child_weight": trial.suggest_loguniform("min_child_weight", 1e-5, 10),
        #"scale_pos_weight": trial.suggest_uniform("scale_pos_weight", 10.0, 30.0),
        "bagging_freq": 1, "bagging_fraction": 0.6,
        "verbosity": -1
    }

    N_iterations_max = 10_000
    early_stopping_rounds = 500

    if boosting_type == "dart":
        N_iterations_max = 100
        early_stopping_rounds = None

    cv_res = lgb.cv(
        params,
        lgb_data_train,
        num_boost_round=N_iterations_max,
        early_stopping_rounds=early_stopping_rounds,
        verbose_eval=False,
        seed=42,
        callbacks=[LightGBMPruningCallback(trial, "multi_logloss")],
    )

    num_boost_round = len(cv_res["multi_logloss-mean"])
    trial.set_user_attr("num_boost_round", num_boost_round)
```

Appendix

Codes & descriptions

Hyperparameter tuning with Optuna

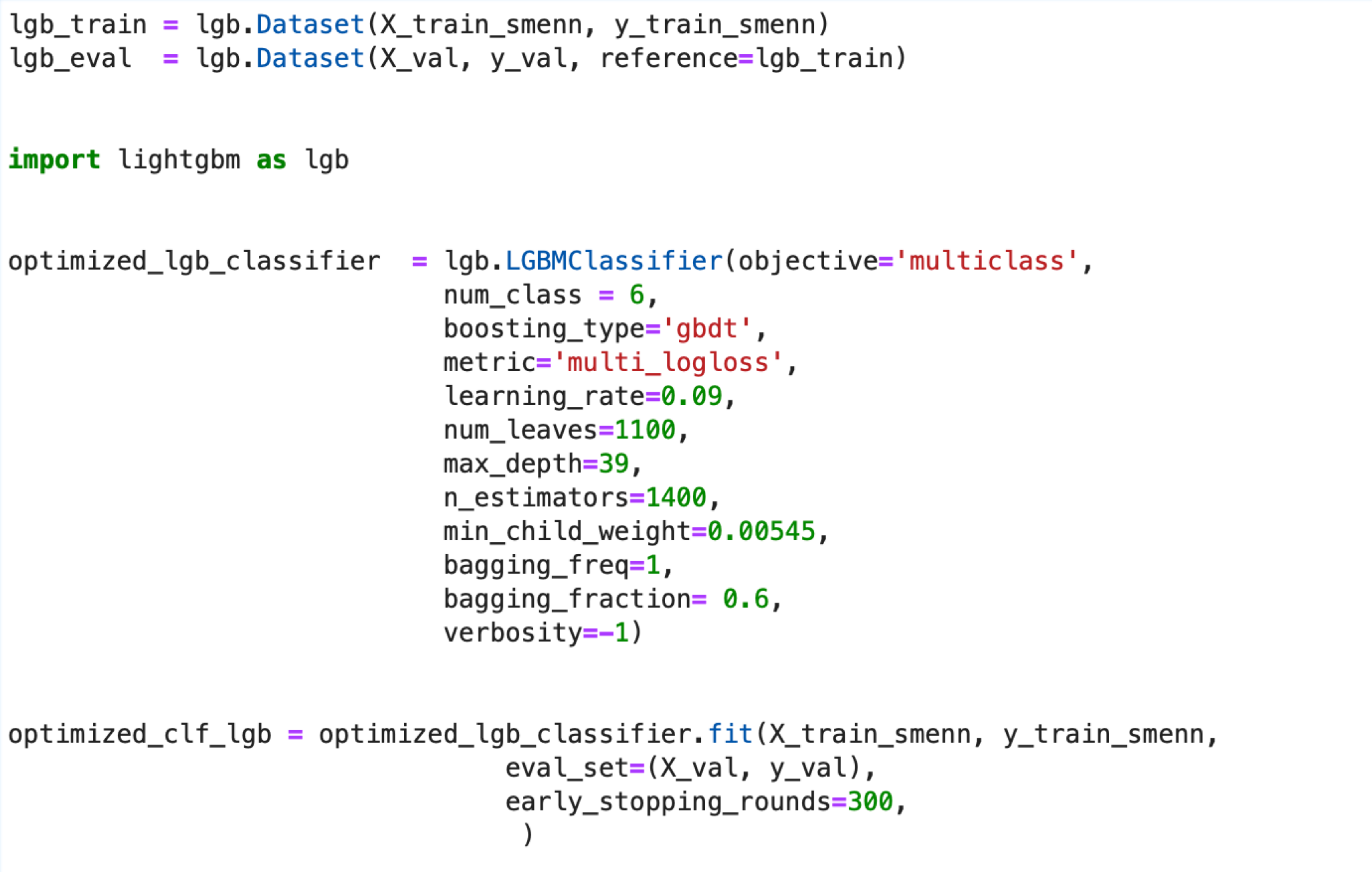


```
optimized_lgb_classifier  = lgb.LGBMClassifier(objective='multiclass',
                            num_class = 6,
                            boosting_type='gbdt',
                            metric='multi_logloss',
                            learning_rate=0.09,
                            num_leaves=1100,
                            max_depth=39,
                            n_estimators=1400,
                            min_child_weight=0.00545,
                            bagging_freq=1,
                            bagging_fraction= 0.6,
                            verbosity=-1)
```

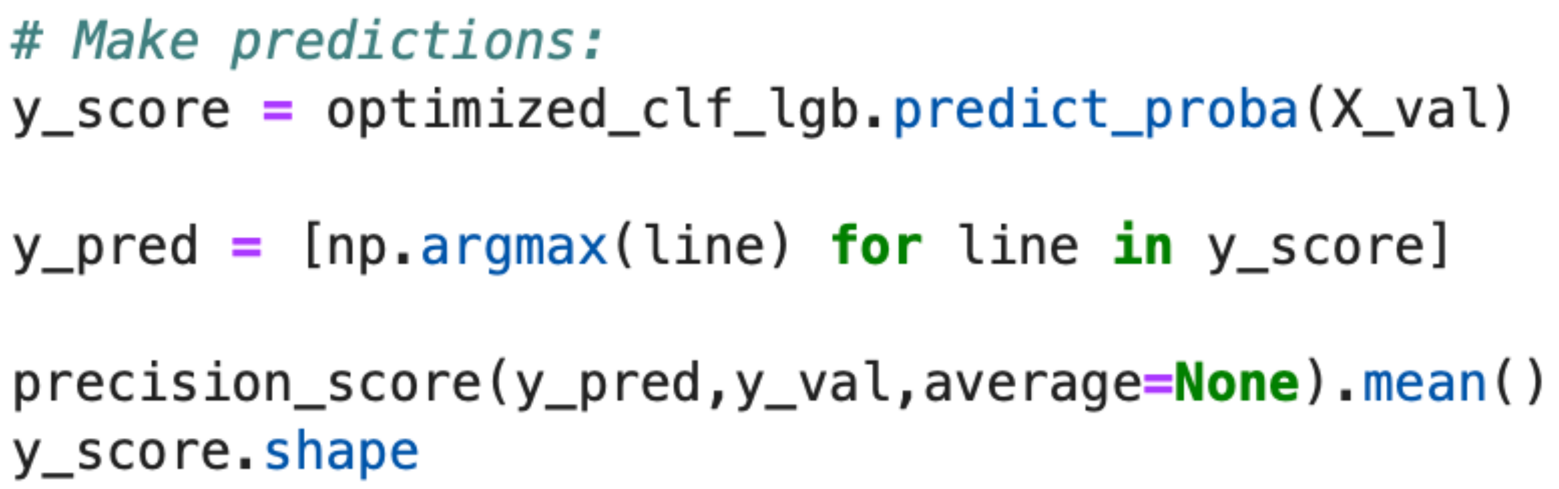
Appendix

Codes & descriptions

Training with LightGBM



```
lgb_train = lgb.Dataset(X_train_smenn, y_train_smenn)
lgb_eval  = lgb.Dataset(X_val, y_val, reference=lgb_train)


import lightgbm as lgb


optimized_lgb_classifier  = lgb.LGBMClassifier(objective='multiclass',
                            num_class = 6,
                            boosting_type='gbdt',
                            metric='multi_logloss',
                            learning_rate=0.09,
                            num_leaves=1100,
                            max_depth=39,
                            n_estimators=1400,
                            min_child_weight=0.00545,
                            bagging_freq=1,
                            bagging_fraction= 0.6,
                            verbosity=-1)


optimized_clf_lgb = optimized_lgb_classifier.fit(X_train_smenn, y_train_smenn,
                                eval_set=(X_val, y_val),
                                early_stopping_rounds=300,
                                 )
```



```
# Make predictions:
y_score = optimized_clf_lgb.predict_proba(X_val)

y_pred = [np.argmax(line) for line in y_score]

precision_score(y_pred,y_val,average=None).mean()
y_score.shape
```