Data Science Lab

Lab 5 solution

1 Introduction

In the following solution, we will go through the pipeline that achieved the baseline result you saw on the leaderboard. Therefore, careful readers will found several things that can be improved and others left apart for further discussions.

1.0.1 Prerequisites

We need to download several contents to complete the lab assignment.

[1]:
```
import numpy as np
import nltk
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/giuseppe/nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
```

[1]:
```
True
```

2 Newsgroup clustering

Let's start by loading the dataset.

[2]:
```
import os

in_dir = "T-newsgroups"
files = sorted(os.listdir(in_dir), key=int) # note how the part 'key=lambda x:␣
↪int(x)' is shortened here
files[:5]
```

[2]:
```
['0', '1', '2', '3', '4']
```

Be careful: `os.listdir` returns filenames in random order. To tackle that, we can just sort them using their integer cast as ordering criteria.

2.1 Data preparation step

Here we can make use of the `TfidfVectorizer` class from scikit-learn. Starting from the code snipped provided in the laboratory text, we have a custom `LemmaTokenizer` class that can be used in place of the TfidfVectorizer's standard tokenizer. In our tokenizer, we create tokens and lemmatize them through nltk `word_tokenize` and `WordNetLemmatizer` respectively.

[3]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk.tokenize import word_tokenize
from nltk.stem.wordnet import WordNetLemmatizer
from nltk.corpus import stopwords as sw
import re
import string

class LemmaTokenizer(object):
    def __init__(self):
        self.lemmatizer = WordNetLemmatizer()

    def __call__(self, document):
        lemmas = []
        re_digit = re.compile("[0-9]") # regular expression to filter digit␣
 ↪tokens

        for t in word_tokenize(document):
            t = t.strip()
            lemma = self.lemmatizer.lemmatize(t)

            # remove tokens with only punctuation chars and digits
            if lemma not in string.punctuation and \
               len(lemma) > 3 and \
               len(lemma) < 16 and \
               not re_digit.match(lemma):
                lemmas.append(lemma)

        return lemmas
```

Let's apply now our custom tokenizer to generate the TFIDF representation. We can let the TfidfVectorizer read the documents for us, using the `input` parameter to its constructor. We should only care about passing to fit_transform the correct path to our files.

[4]:
```
stopwords = sw.words('english') + ["'d", "'ll", "'re", "'s", "'ve", 'could',␣
 ↪'doe', 'ha', 'might', 'must', "n't", 'need', 'sha', 'wa', 'wo', 'would']
tokenizer = LemmaTokenizer()
vectorizer = TfidfVectorizer(input='filename', tokenizer=tokenizer,␣
 ↪stop_words=stopwords)
```

```
filepath = [os.path.join(in_dir, f) for f in files] # here we need the correct␣
↪path
X_tfidf = vectorizer.fit_transform(filepath)
X_tfidf
```

[4]:
```
<4000x40684 sparse matrix of type '<class 'numpy.float64'>'
        with 357446 stored elements in Compressed Sparse Row format>
```

Note that we extended the list of stopwords with further non significative words.

2.1.1 Principal Component Analysis

As you can see, we are dealing with about 40k different features. These are all the distinct words found in our corpus (i.e. the collection of our news). We can transform and reduce our input feature space through the Principal Component Analysis (PCA) (see Appendix on the laboratory text).

Let's focus on PCA applied by means of the Singular Value Decomposition (SVD). Each component extracted by the decomposition will express a given amount of the variance of our data. In the best case, all the variance is expressed by a low number of new features. As suggested, the `TruncatedSVD` class let us decide how many components we want to retain in the new space. Additionally, it provides you the variance those components encode with some inner attributes.

[5]:
```
from sklearn.decomposition import TruncatedSVD
```

Let's try with a low number of features, e.g. 30, and use them to cluster our points. Please remember that the SVD algebric method transforms the initial space. Hence, you will not end up with the 30 words the best separate your data. Instead, you will have new components (i.e. projections of your points) in a new 30-dimensional space.

The `fit_transform` method performs the decomposition and maps the input data to the new vector space.

[6]:
```
svd = TruncatedSVD(n_components=30, random_state=42)
X_svd = svd.fit_transform(X_tfidf)

print(f"Total variance explained: {np.sum(svd.explained_variance_ratio_):.2f}")
```

```
Total variance explained: 0.07
```

The 7% is quite low for real applications. We will investigate it in Section 2.4.

2.2 Clustering and wordclouds

Now that we have an initial vector representation, we can use it to cluster our documents. We will focus on the K-means algorithm with K starting from 2 and increasing. Next, we can choose the K value that better helps us to spot topics in the final clusters. Doing so, we are impersonating a domain expert that analyzes and evaluates the final subdivision. Remember that, in our context, one clear topic per cluster is the desired outcome.

One simple approach to describe the clusters would exploit wordcloud images. These "clouds" include the most important words in collection of texts for a certain metric. The more a word is significant the bigger it appears in the picture. Fortunately, we already have a "significance" metric: the TFIDF weight of each word. Let's exploit its very meaning and choose it as the "importance" metric for the wordcloud.

[7]:
```
from sklearn.cluster import KMeans
from wordcloud import WordCloud
import matplotlib.pyplot as plt
%matplotlib inline
```

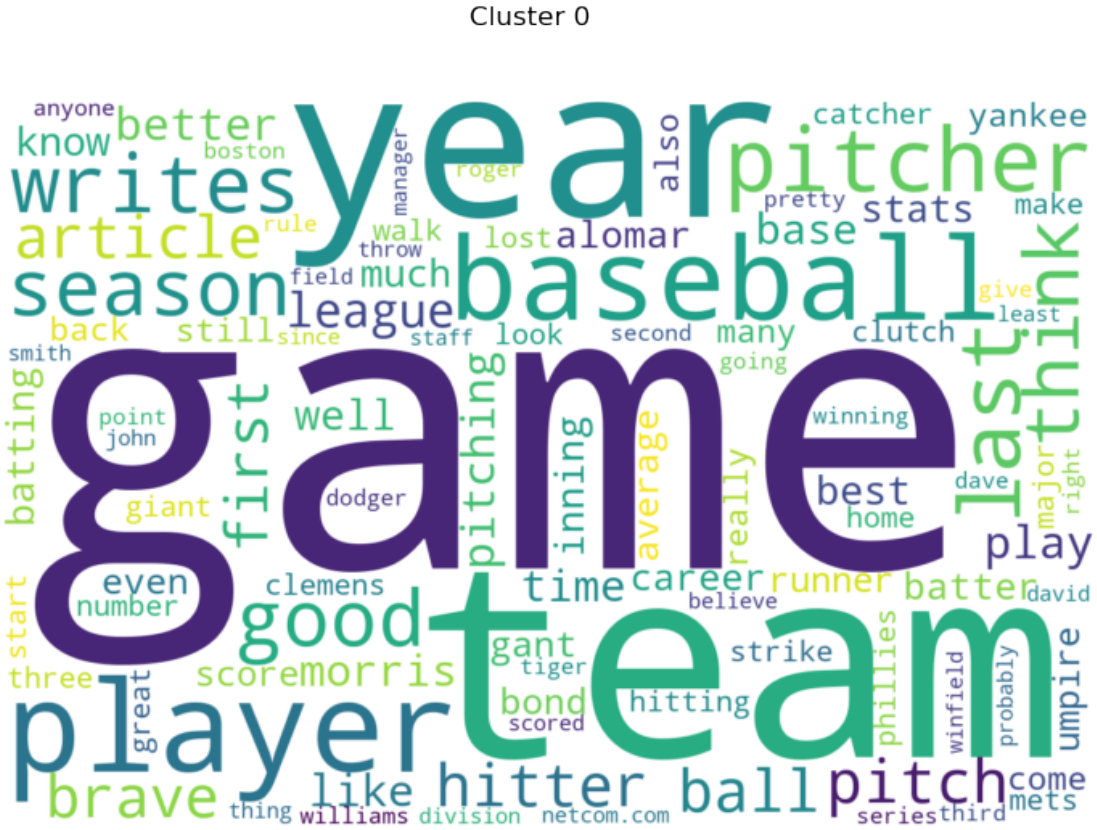
Let's define a function that clusters our data with K-means and generates one wordcloud per cluster. As you can see in the following code, we will use only the `top_count` words with the highest TFIDF value.

[8]:
```
def generate_wordclouds(X, X_tfidf, k, word_positions):
    """Cluster X with K-means with the specified k, and generate one wordcloud␣
↪per cluster.

    :input X: numpy.array or numpy.matrix to cluster
    :input X_tfidf: sparse matrix with TFIDF values
    :input k: the k to be used in K-means
    :input word_positions: dictionary with pairs as word index (in vocabulary)␣
↪-> word
    :return cluster_ids: set with the clusters ids
    """

    model = KMeans(n_clusters=k, random_state=42, n_jobs=-1)
    y_pred = model.fit_predict(X)
    cluster_ids = set(y_pred)
    top_count = 100

    for cluster_id in cluster_ids:

        # compute the total tfidf for each term in the cluster
        tfidf = X_tfidf[y_pred == cluster_id]
        tfidf_sum = np.sum(tfidf, axis=0) # numpy.matrix
        tfidf_sum = np.asarray(tfidf_sum).reshape(-1) # numpy.array of shape␣
↪(1, X.shape[1])
        top_indices = tfidf_sum.argsort()[-top_count:]

        term_weights = {word_positions[idx]: tfidf_sum[idx] for idx in␣
↪top_indices}
        wc = WordCloud(width=1200, height=800, background_color="white")
        wordcloud = wc.generate_from_frequencies(term_weights)

        fig, ax = plt.subplots(figsize=(10, 6), dpi=100)
        ax.imshow(wordcloud, interpolation='bilinear')
```

```
        ax.axis("off")
        fig.suptitle(f"Cluster {cluster_id}")

    return cluster_ids
```

2.2.1 K = 2

[9]:
```
word_positions = {v: k for k, v in vectorizer.vocabulary_.items()}
cluster_ids = generate_wordclouds(X_svd, X_tfidf, 2, word_positions)
```



Cluster 1

It can easily be seen that Cluster 0 is characterized by words related to baseball. We can assume that there exists at least a group of news talking about baseball/sport. On the contrary, Cluster 1 looks way less promising. Many among the heaviest words have a low significance (e.g. article, writes, thing, like, think). Results can surely improve increasing K.

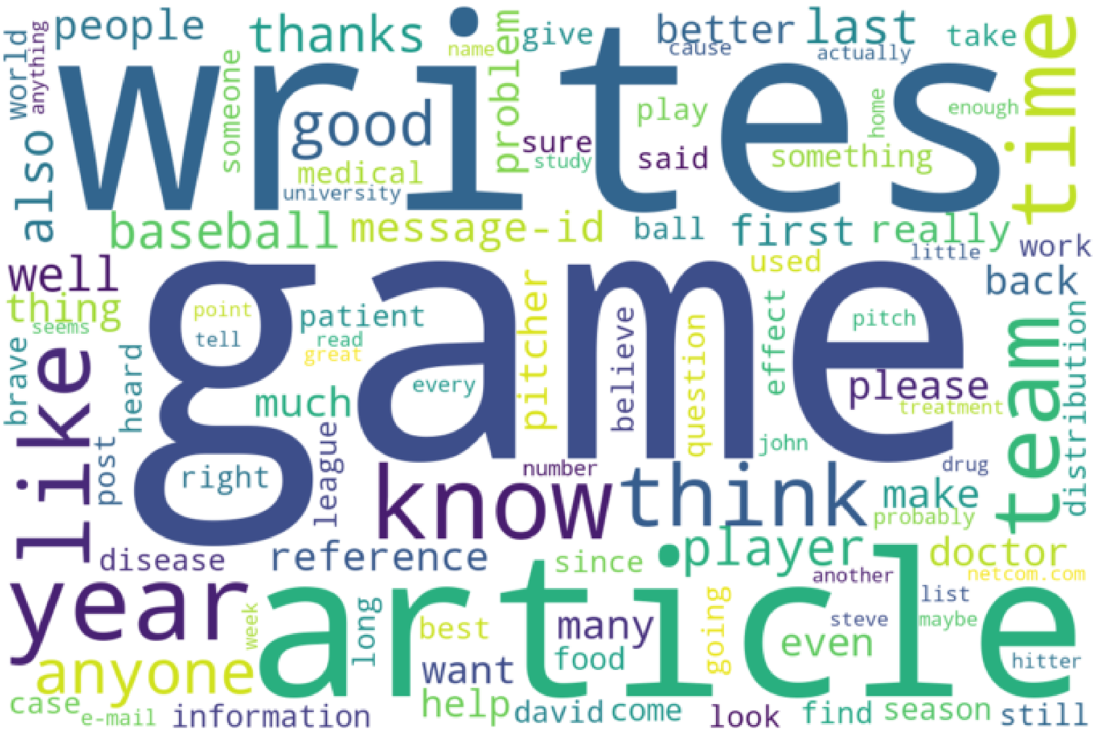
2.2.2 K = 3

[10]:
```
cluster_ids = generate_wordclouds(X_svd, X_tfidf, 3, word_positions)
```

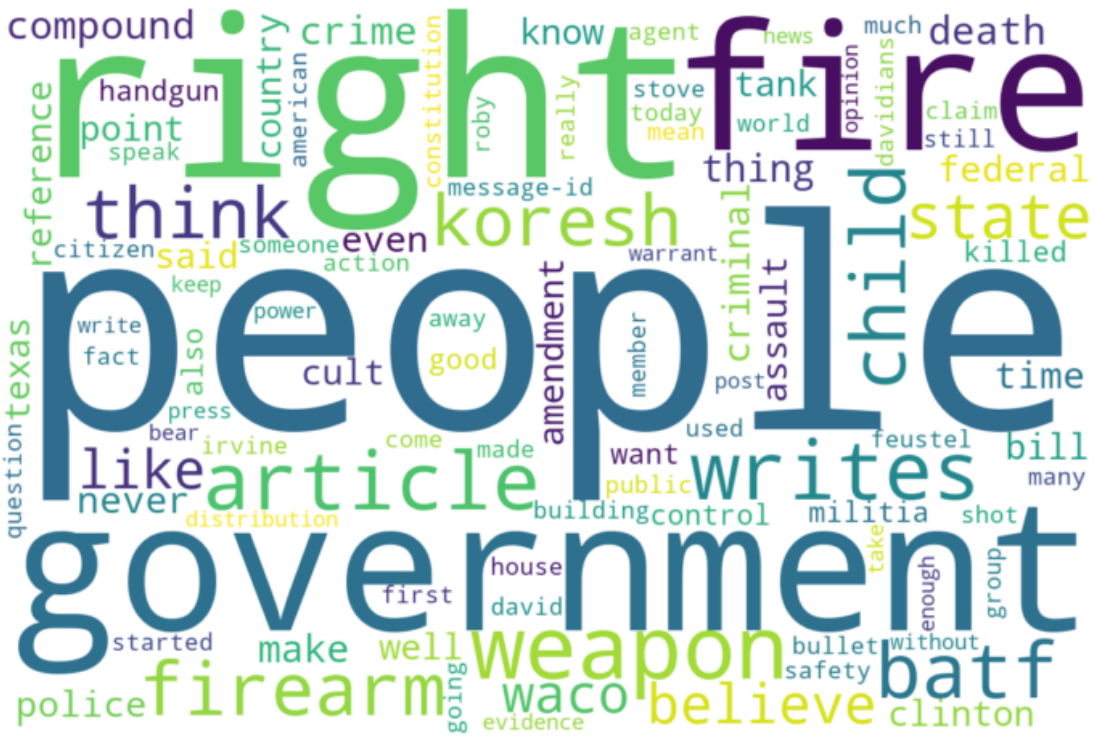
Cluster 0


space
shuttle
henry
launch
moon
orbit
lunar
utzoo
reference
news
technology
billion
development
high
problem
year
zoology
vehicle
cost
data
prize
design
also
array
even
project
world
station
much
thing
flight
around
russian
power
back
large
message-id
people
article
light
work
pluto
surface
support
better
astronomy
object
good
probe
money
fuel
program
payload
zoo.toronto.edu
nasa
toronto
first
spacecraft
propulsion
idea
option
software
spencer
available
telescope
science
part
system
make
enough
something
billboard
time
baalke
orbital
planet
like
mission
think
date
rocket
nsmca
commercial
organization
titan
know
question
long
image
well
research
solar
using
writes
satellite
earth
mass
want
going
temperature

Cluster 1


writes
game
article
know
think
team
time
like
year
anyone
good
thanks
people
last
better
also
baseball
message-id
first
really
well
player
doctor
reference
many
even
please
make
back
much
pitcher
problem
patient
thing
right
want
help
information
disease
food
best
since
play
said
something
medical
sure
give
take
world
someone
question
believe
effect
distribution
league
heard
post
brave
long
going
ball
used
work
case
e-mail
david
come
look
find
season
still
anything
name
cause
actually
enough
home
study
university
little
point
seems
tell
read
great
every
pitch
treatment
john
drug
number
probably
another
netcom.com
list
steve
maybe
hitter
week

Cluster 2



It looks like a new topic has popped out. Cluster 0 is now related to space stuff (e.g. space, launch, NASA, shuttle). Also, Cluster 2 now seems to highlight words related to law, government, and justice in general. Note that you cannot expect any fixed order in clusters' Ids, hence Cluster * is likely going to change "identity" from different runs.

There are at least two reasons that should lead you to increase K. On one hand, we lost the topic identified in Cluster 0 with K=2 (baseball/sport) and that we assumed to find in our final result. On the other hand, Cluster 1 has still no well-defined identity.

2.2.3 K = 4

[11]:
```
cluster_ids = generate_wordclouds(X_svd, X_tfidf, 4, word_positions)
```

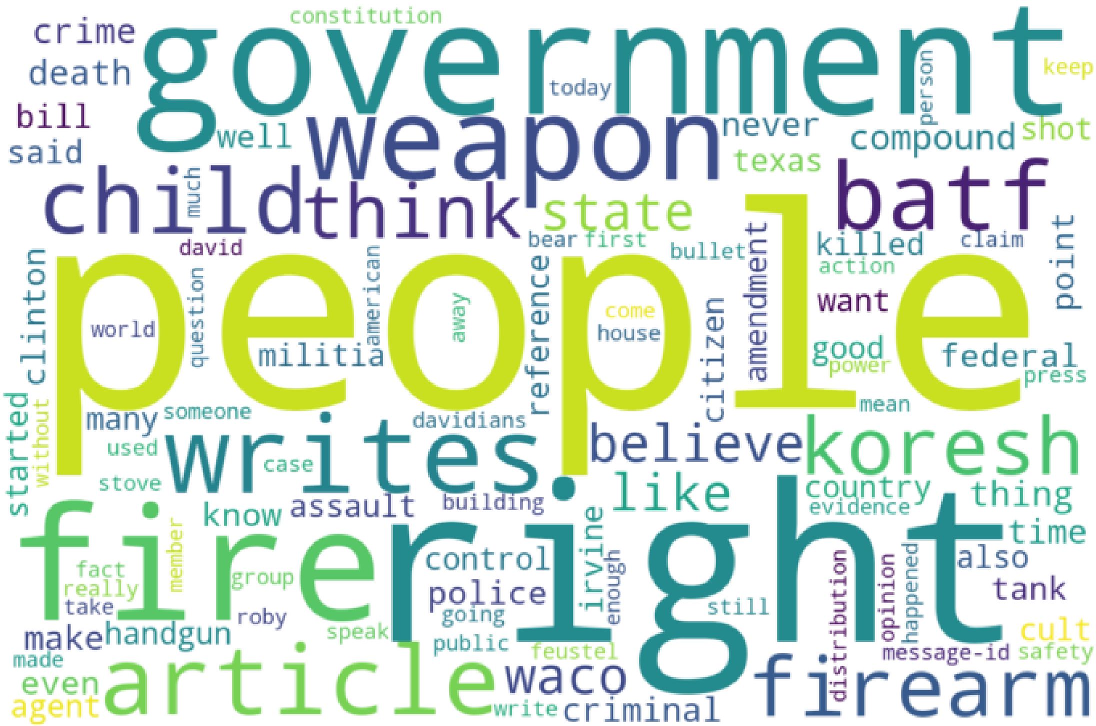
Cluster 0

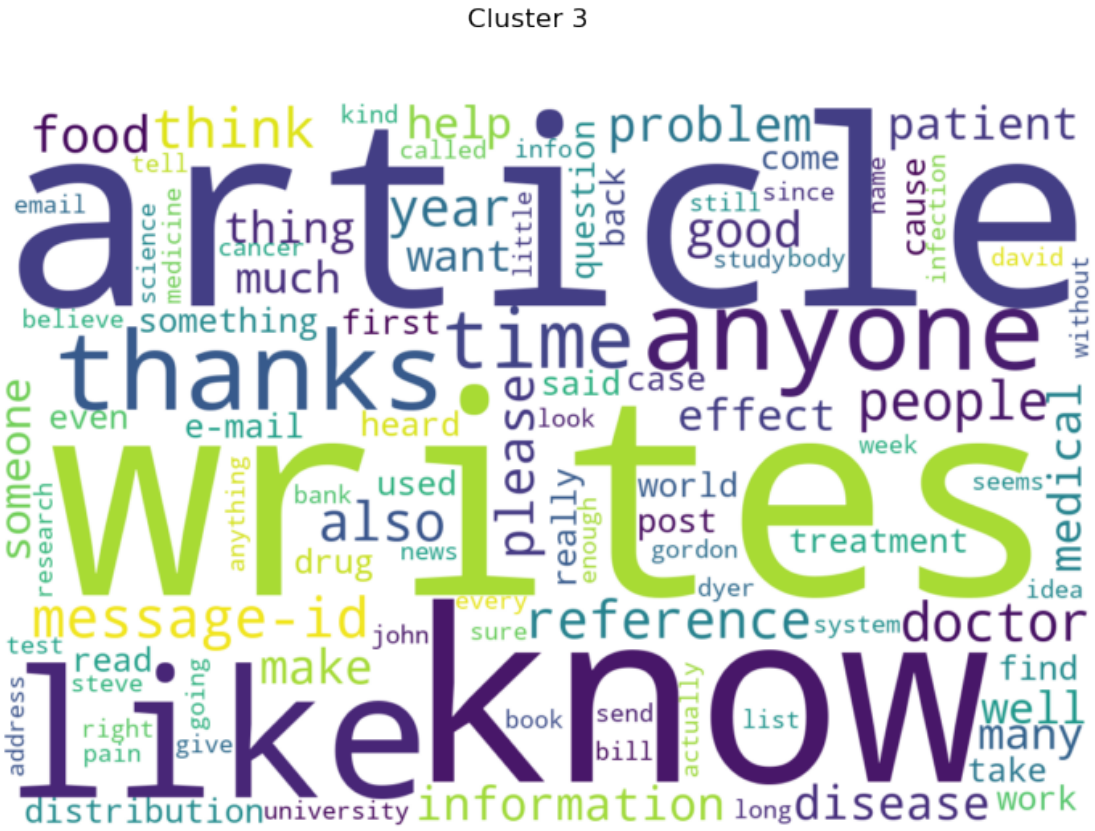

space
henry
shuttle
mission
launch
moon
orbit
zoo.toronto.edu
system
lunar
earth
program
satellite
station
nasa
solar
cost
article
writes
data
project
science
spencer
thing
like
work
flight
technology
year
much
also
idea
spacecraft
rocketsoftware
baalke
first
billion
know
world
part
light
nsmca
mass
think
array
utzoo
people
problem
time
zoology
development
design
vehicle
message-id
stsci.edu
toronto
enough
reference
make
prize
back
option
around
good
high
image
large
want
date
payload
element
money
propulsion
power
planet
jacked
orbital
temperature
news
going
using
fuel
adam
pluto
commercial
long
probe
titan
better
well
available
probably
telescope
russian
something
billboard
base
organization
allen

Cluster 1


least
alomar
bond
career
better
also
second
field
catcher
division
david
well
best
still
player
pitch
look
third
winning
tiger
winfield
dodger
play
umpire
point
home
baseball
think
ball
great
clutch
score
know
giant
record
game
season
good
batting
netcom.com
phillies
make
throw
probably
morris
article
really
back
brave
strike
average
right
gant
clemens
inning
going
give
manager
base
many
baerga
league
writes
first
ever
pitching
roger
offense
smith
year
team
williams
stats
defensive
three
thing
since
lost
like
time
hitting
yankee
number
mets
boston
scored
major
pitcher
last
much
even
hitter
staff
rule
runner
believe
batter
walk
start
series

Cluster 2


constitution
crime
government
keep
today
person
death
bill
weapon
never
compound
shot
said
well
texas
child
think
batf
state
much
david
bear
first
bullet
killed
claim
action
point
people
american
clinton
question
away
come
reference
amendment
citizen
want
world
house
good
power
federal
militia
press
mean
started
without
someone
many
davidians
believe
koresh
used
writes
case
stove
like
country
evidence
thing
assault
building
know
time
member
fire
right
irvine
enough
distribution
opinion
happened
also
control
fact
really
group
tank
police
take
going
still
roby
make
handgun
cult
speak
public
feustel
message-id
safety
made
even
waco
article
firearm
agent
write
criminal



In this case, we can spot a topic for three clusters out of four. Texts within Cluster 0 deal with space while Cluster 1 contains sport-related news. In Cluster 2 things get blurred. However, it seems again close to law and public justice in relation to guns and weapons. Finally, Cluster 3 stands as the dirty cluster we have already encountered before.

Maybe another cluster exists and K=5 would have split Cluster 2 in two. Also, many of the significant words in Cluster 3 are not relevant to the task (e.g. know, like, writes, article, thanks, anyone) and the overall quality is poor. At this point, a solution could be to filter them out. For now, let's increase K to 5.

2.2.4 K = 5

[12]:
```
cluster_ids = generate_wordclouds(X_svd, X_tfidf, 5, word_positions)
```

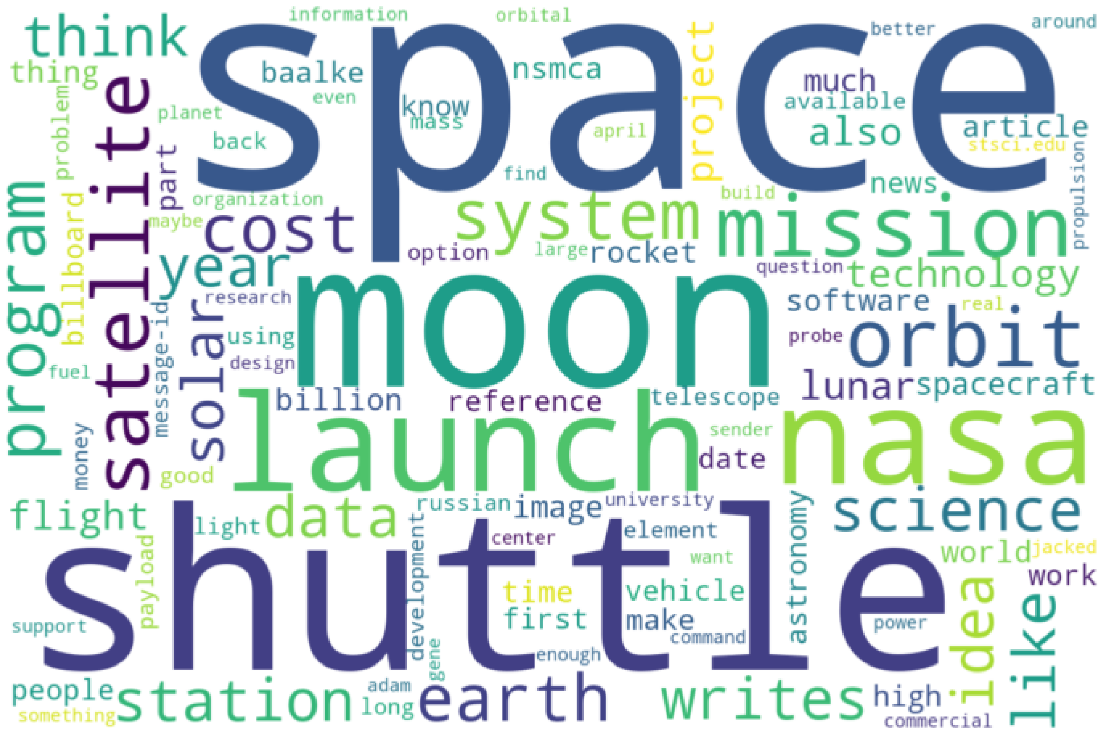
Cluster 0


really
question
bullet
american
davidians
opinion
militia
assault
government
writes
firearm
state
compound
david
roby
want
still
many
bear
people
believe
feustel
amendment
started
point
happened
handgun
well
clinton
take
today
away
agent
going
keep
good
used
said
building
action
life
constitution
texas
press
also
know
article
batf
shot
public
think
like
even
federal
child
right
come
evidence
someone
without
tank
time
citizen
reference
waco
police
fact
person
group
made
thing
first
safety
never
make
mean
likely
claim
cult
country
killed
place
criminal
much
wrong
warrant
fire
weapon
control
house
member
world
enough
bill
news
koresh
death
crime
message-id
case

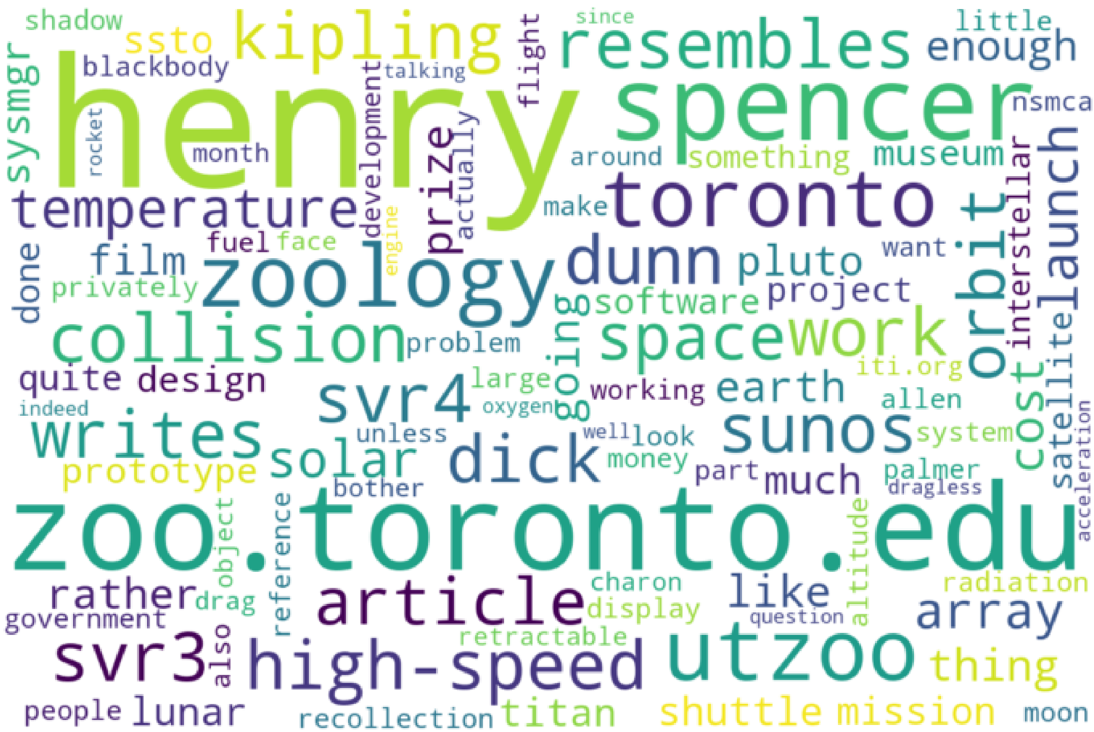
Cluster 1

system
heard
distribution
something
come
year
know
thanks
world
going
well
effect
also
without
probably
cause
reference
think
steve
disease
group
much
article
even
help
week
cancer
long
still
work
said
really
post
good
make
seems
info
want
science
give
sure
case
bank
kind
first
anyone
news
test
soon
e-mail
study
doctor
like
please
medicine
people
read
book
bill
idea
medical
patient
question
called
david
university
list
since
email
body
writes
infection
dyer
find
treatment
believe
take
tell
many
look
anything
enough
research
thing
used
back
name
john
little
actually
gordon
food
time
drug
message-id
right
problem
someone
pain
information

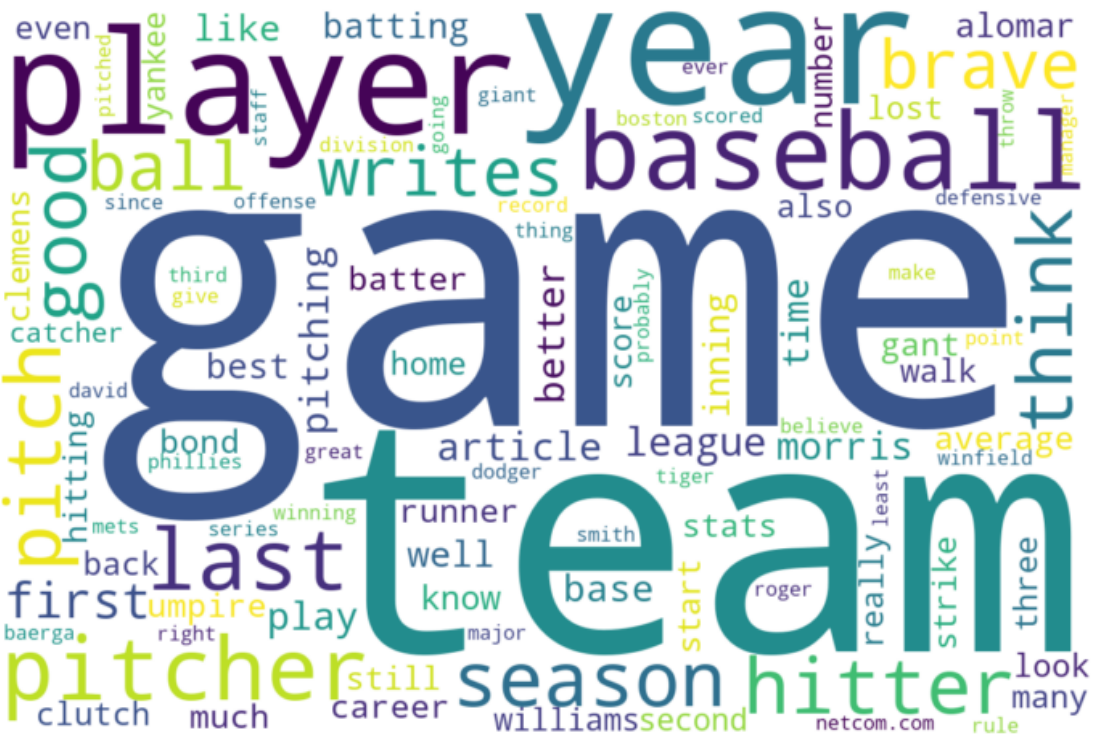
Cluster 2


space
moon
shuttle
launch
nasa
mission
system
orbit
satellite
program
think
cost
year
solar
data
science
station
earth
writes
like
idea
technology
lunar
spacecraft
also
project
article
much
baalke
nsmca
thing
know
mass
back
news
software
billion
reference
telescope
date
flight
image
russian
element
world
work
time
vehicle
first
make
people
high
astronomy
development
information
orbital
better
around
available
planet
even
april
stsci.edu
find
build
organization
option
large
rocket
question
propulsion
research
real
probe
using
design
fuel
message-id
billboard
problem
part
maybe
sender
money
good
university
light
center
want
jacked
payload
support
command
power
enough
adam
gene
long
something
commercial

Cluster 3


henry
spencer
zoo.toronto.edu
zoology
toronto
kipling
resembles
collision
space
work
dunn
high-speed
utzoo
svr3
svr4
article
dick
sunos
writes
temperature
orbit
launch
prize
solar
pluto
earth
array
like
rather
lunar
titan
shuttle
mission
thing
ssto
enough
museum
film
cost
much
prototype
design
quite
software
project
going
satellite
interstellar
something
sysmgr
blackbody
month
around
make
shadow
since
little
nsmca
talking
flight
development
actually
rocket
fuel
face
engine
privately
done
problem
large
oxygen
working
allen
iti.org
want
indeed
unless
well
look
system
money
part
palmer
dragless
acceleration
bother
object
reference
altitude
charon
display
radiation
question
drag
government
also
retractable
people
recollection
moon

Cluster 4



From a first look, only Cluster 2 (space) and Cluster 4 (baseball) describe a specific topic. Cluster 0 is nosier but still the "justice" topic is visible (e.g. people, right, government, fire).

Nevertheless, the overall clustering outcome seems worse to K = 4. Cluster 1 and Cluster 3 contain mainly noise. That could advise us that we are moving away from the right number of topics present in the original dataset. However, let's take it one step further and try with K = 6.

[13]:
```
cluster_ids = generate_wordclouds(X_svd, X_tfidf, 6, word_positions)
```

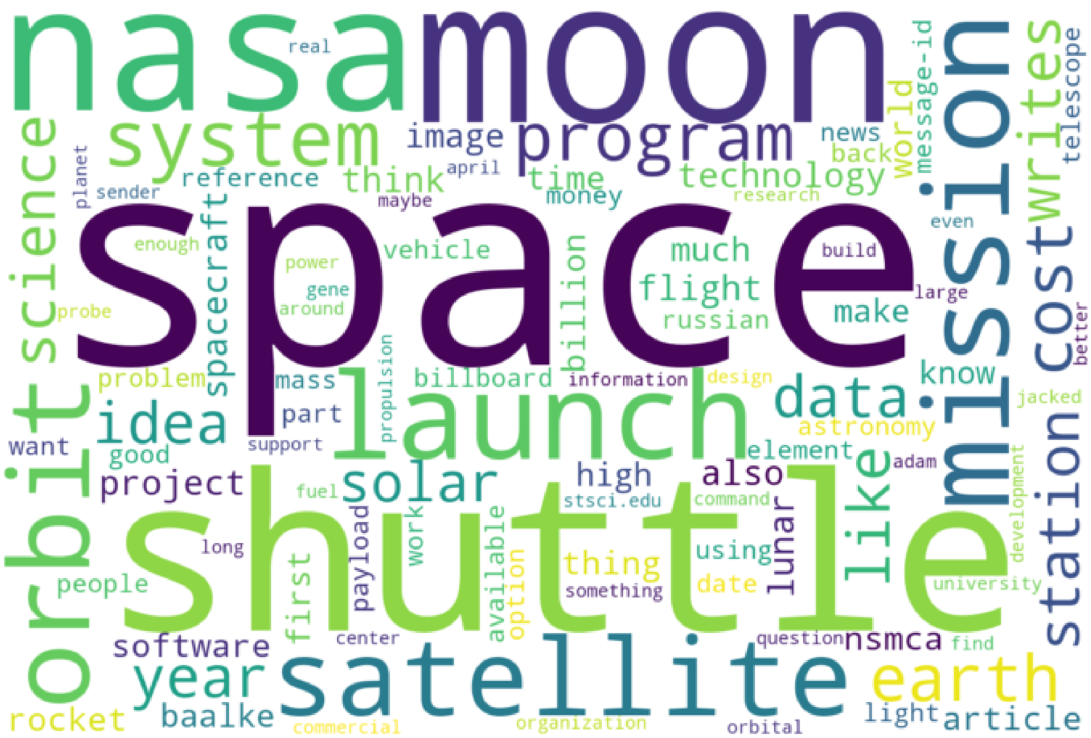
Cluster 0


warrant
happened
member
action
child
writes
crime
federal
firearm
david
case
control
time
thing
make
mean
opinion
public
used
without
take
right
state
koresh
clinton
away
never
said
reference
point
citizen
really
enough
started
question
even
waco
someone
tank
bill
american
compound
likely
government
claim
good
shot
evidence
handgun
assault
cult
fire
like
keep
think
believe
well
first
amendment
come
world
also
people
feustel
agent
country
militia
group
still
fact
news
life
bear
place
constitution
house
press
safety
person
message-id
bullet
police
going
killed
much
today
know
weaponbatf
wrong
article
many
death
want
criminal
made
building
davidians
texas
roby

Cluster 1


message-id
thanks
writes
problem
send
question
heard
believe
dyer
find
look
john
patient
little
anything
group
post
called
also
actually
even
book
friend
disease
please
pain
much
medical
address
long
steve
body
first
said
still
like
used
doctor
think
make
many
email
name
news
week
sure
read
enough
year
food
time
treatment
e-mail
bill
cause
drug
really
since
article
else
help
list
cancer
world
want
university
system
take
right
tell
good
thing
effect
come
case
idea
study
science
back
medicine
every
anyone
information
seems
info
infection
give
kind
know
work
going
someone
people
without
research
well
reference
distribution
something
david

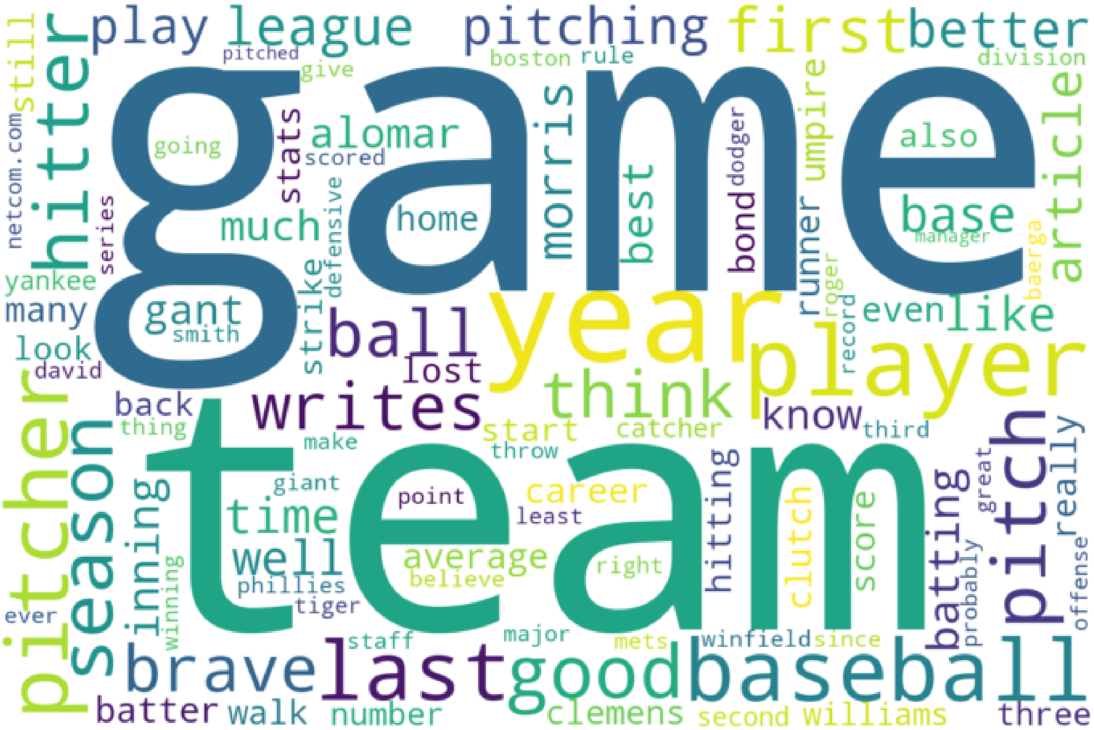
Cluster 2

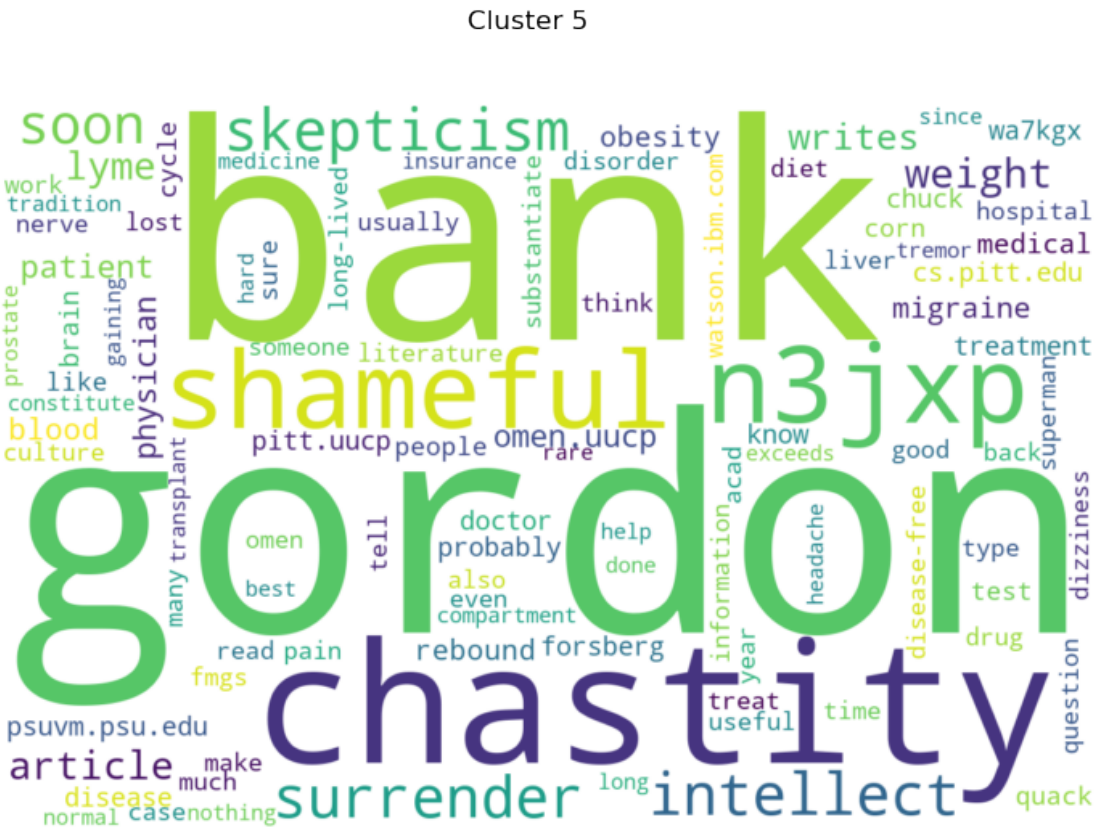

nasa
moon
space
shuttle
satellite
launch
mission
writes
cost
station
orbit
science
system
program
image
real
planet
sender
reference
think
april
time
money
maybe
technology
research
news
back
world
message-id
telescope
even
build
large
better
enough
power
gene
around
probe
vehicle
much
flight
russian
make
spacecraft
billion
idea
data
know
problem
mass
billboard
information
design
jacked
part
astronomy
want
good
support
propulsion
element
adam
like
project
solar
high
also
fuel
stsci.edu
command
development
long
work
payload
lunar
available
thing
using
people
option
something
date
university
first
software
center
question
nsmca
find
year
earth
rocket
baalke
light
article
commercial
organization
orbital

Cluster 3


zoo.toronto.edu
henry
spencer
utzoo
toronto
zoology
resembles
sunos
svr3
article
collision
temperature
launch
writes
orbit
dunn
dick
svr4high-speed
work
kipling
solar
space
array
mission
prototype
sysmgr
software
like
going
lunar
pluto
prize
film
cost
earth
ssto
design
titan
rather
much
thing
enough
shuttle
interstellar
museum
quite
done
project
something
satellite
talking
reference
recollection
also
radiation
oxygen
indeed
allen
bother
working
acceleration
charon
part
question
moon
privately
shadow
drag
atmosphere
around
blackbody
display
shielding
palmer
make
look
system
nsmca
month
dragless
rocket
money
engine
face
altitude
development
want
people
object
large
government
iti.org
problem
little
retractable
actually
since
fuel
well

Cluster 4


game
team
year
player
last
good
baseball
brave
pitcher
season
hitter
writes
think
ball
pitch
first
better
play
league
pitching
article
morris
alomar
time
well
inning
batting
gant
much
base
like
know
still
stats
home
best
bond
umpire
runner
also
even
many
look
back
lost
strike
career
average
hitting
clutch
score
really
start
catcher
batter
walk
number
clemens
second
williams
three
yankee
going
scored
pitched
give
boston
rule
division
netcom.com
series
defensive
dodger
manager
baerga
roger
record
smith
david
thing
make
throw
third
giant
point
least
great
probably
offense
winning
phillies
believe
right
ever
tiger
staff
major
mets
winfield
since



The outcome looks pretty much like the one with K = 5: we ended up with two well-defined clusters, one a bit blurred but still identifiable and noise for the rest of them. Since no improvement was achieved in the last two K values, we can decide to stop the analysis here and select K = 4 for the next steps.

2.3 Frequent itemset

Among our trials, the outcome from K = 4 seems the most promising one. To extend the study, we can now analyze the most frequent sets of words in each cluster separately, just as we did with wordclouds. We will use the `mlxtend` library.

[14]:
```
import pandas as pd
from mlxtend.frequent_patterns import fpgrowth
```

[15]:
```
model = KMeans(n_clusters=4, random_state=42, n_jobs=-1)
y_pred = model.fit_predict(X_svd)
cluster_ids = set(y_pred)
```

Given the dimensions in play, we prefer the FP-Growth algorithm over Apriori, since the former does not require to create the candidate sets explicitly. Note that we will use pandas here and,

specifically, its sparse data structures to reduce the memory penalty. You can read more about it here.

[16]:
```
min_support = 0.3
dist_words = sorted(v for k, v in word_positions.items()) # distinct words in␣
↪the vocabulary
for cluster_id in cluster_ids:
    print(f"FP-Growth results on Cluster {cluster_id} with min support␣
↪{min_support}")

    tfidf = X_tfidf[y_pred == cluster_id]
    tfidf[tfidf > 0] = 1 # encoded as binary "presence/absence" representation␣
↪as required by mlxtend
    df = pd.DataFrame.sparse.from_spmatrix(tfidf, columns=dist_words) # df is a␣
↪pandas sparse dataframe

    fset = fpgrowth(df, min_support=min_support, use_colnames=True).
↪sort_values(by='support', ascending=False)
    print(fset, '\n')
```

```
FP-Growth results on Cluster 0 with min support 0.3
     support           itemsets
1   0.615234            (space)
2   0.599609           (writes)
3   0.546875          (article)
10  0.494141  (writes, article)
4   0.378906             (like)
5   0.349609             (also)
9   0.322266    (writes, space)
8   0.320312             (much)
7   0.306641             (year)
0   0.302734           (system)
6   0.300781            (think)

FP-Growth results on Cluster 1 with min support 0.3
     support                 itemsets
0   0.672504                (writes)
1   0.604203                  (game)
2   0.597198                (article)
16  0.563923        (writes, article)
12  0.502627                  (year)
3   0.432574                  (last)
4   0.430823                  (team)
5   0.390543                 (think)
15  0.383538           (game, writes)
6   0.378284                  (like)
13  0.376532                  (good)
7   0.367776                  (time)
```

```
8   0.366025               (baseball)
20  0.355517           (year, writes)
17  0.350263          (game, article)
9   0.343257                  (first)
10  0.339755                 (player)
18  0.327496  (game, writes, article)
19  0.316988             (last, year)
21  0.309982          (year, article)
14  0.304729                   (know)
11  0.304729                   (well)

FP-Growth results on Cluster 2 with min support 0.3
     support                    itemsets
0   0.728129                   (writes)
1   0.682369                  (article)
16  0.609690          (writes, article)
2   0.557201                   (people)
14  0.418573                    (right)
7   0.414536                    (think)
17  0.410498           (people, writes)
18  0.380888          (people, article)
8   0.379542                     (like)
11  0.376851                (government)
9   0.360700                      (time)
13  0.345895                      (make)
10  0.340511                     (thing)
3   0.336474                      (even)
19  0.332436  (people, writes, article)
4   0.325707                      (know)
5   0.317631                   (believe)
20  0.316285            (think, writes)
15  0.312248                      (also)
22  0.310902            (writes, right)
6   0.309556                (reference)
21  0.305518           (think, article)
12  0.301480                      (well)

FP-Growth results on Cluster 3 with min support 0.3
    support           itemsets
0  0.518399           (writes)
1  0.469641          (article)
2  0.418583  (writes, article)
```

Here are reported word sets with support at least above 30% sorted by support. As you can see, compared to a wordcloud visualization, cluster topics are harder to identify. Nonetheless, with our prior knowledge coming from wordclouds, we can spot some key words like: * “space” in Cluster 0 * “team”, “baseball” and “game” in Cluster 1 * “government”, “people” in Cluster 2

2.4 Further comments

Every previous result was generated with a transformed dataset by means of the Truncated SVD. However, we skipped over the discussion on the number of components. Actually, our new representation expresses only 7% of the explained variance, which is an extremely low number in almost every real-world case.

With this in mind, we can enhance the truncated representation and then explore the results with the best configuration so far (K = 4). The choice of the number of components which span the new vector space is typically a matter of the amount of explained variance desired and varies from application to application. To sufficiently distantiate from previous results, let's see how clustering behaves when we retain 80% of the explained variance.

Let's see if 2000 components are enough to our goal.

[17]:
```
svd = TruncatedSVD(n_components=2000, random_state=42)
X_svd = svd.fit_transform(X_tfidf)
```

We can now compute the cumulative sum of the explained variance ratio, per component, and check whether we cross the 80% or not.

[18]:
```
cum_variance = np.cumsum(svd.explained_variance_ratio_)
idx = np.argmax(cum_variance > .8)
idx
```

[18]:
```
1622
```

Hence, we should transform our data into a new 1622-dimensional space to retain at least 80% of the explained variance. Let's do it and inspect the wordclouds obtained from it.

[19]:
```
svd = TruncatedSVD(n_components=1622, random_state=42)
X_svd = svd.fit_transform(X_tfidf)
_ = generate_wordclouds(X_svd, X_tfidf, 4, word_positions)
```

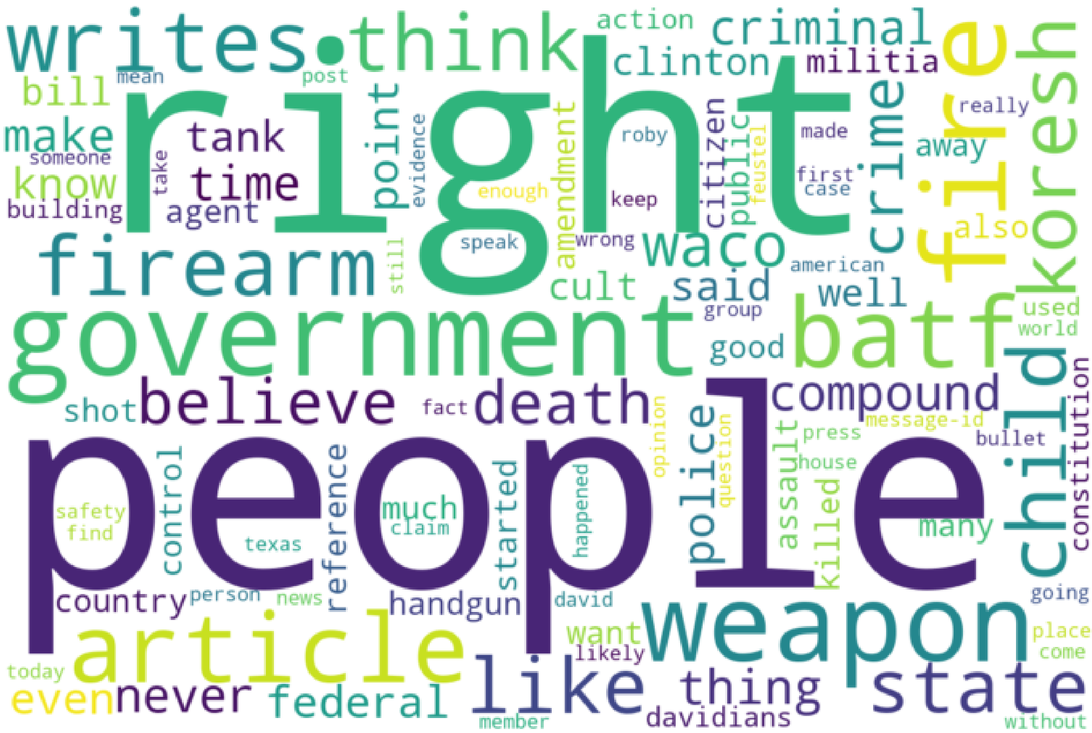
Cluster 0

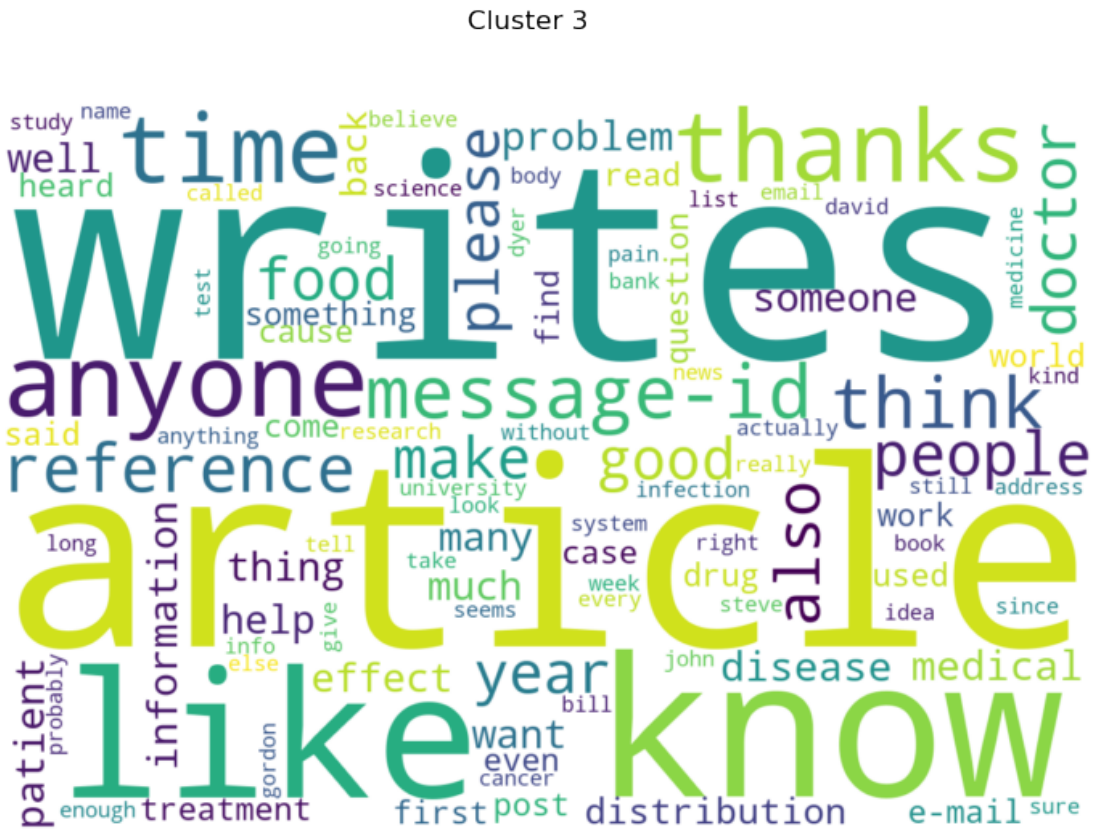

space
shuttle
henry
orbit
launch
mission
moon
nasa
satellite
system
zoo.toronto.edu
earth
writes
article
solar
project
program
cost
lunar
work
software
also
like
station
science
spencer
flight
much
data
billion
spacecraft
question
year
thing
idea
baalke
date
large
image
payload
probe
rocket
something
power
problem
prize
going
object
billboard
research
make
vehicle
using
first
russian
know
want
high
good
around
pluto
development
message-id
well
enough
nsmca
people
planet
even
light
long
back
toronto
array
part
design
propulsion
available
commercial
telescope
astronomy
organization
think
world
option
orbital
mass
technology
news
reference
better
money
temperature
time
fuel
zoology
allen
utzoo
titan
stsci.edu

Cluster 1

home
third
since
make
career
ever
walk
gant
batter
alomar
first
last
team
player
even
runner
throw
offense
right
well
give
winfield
catcher
much
play
score
defensive
strike
still
clutch
number
scored
believe
baerga
brave
writes
hitter
ball
boston
series
least
left
great
dodger
netcom.com
game
season
baseball
rule
tiger
point
base
yankee
mets
going
best
umpire
hitting
clemens
three
batting
stats
like
david
bond
major
year
start
many
thing
league
inning
good
really
division
lost
pitched
back
second
average
better
also.
look
know
time
staff
roger
winning
pitch
think
pitcher
probably
dave
giant
article
pitching
morris
williams
smith
phillies

Cluster 2


writes
think
right
action
criminal
fire
koresh
clinton
militia
mean
post
bill
make
tank
really
someone
roby
made
away
know
time
evidence
take
citizen
public
feustel
crime
first
case
building
agent
point
enough
keep
also
firearm
speak
wrong
waco
amendment
american
still
cult
said
well
government
batf
child
group
used
world
good
believe
death
compound
shot
fact
opinion
press
message-id
people
police
bullet
question
house
constitution
control
reference
started
assault
killed
safety
much
claim
happened
many
find
texas
country
person
news
handgun
david
going
weapon
article
want
place
likely
come
today
even
never
federal
like
thing
state
davidians
member
without



The outcome is pretty much similar to the one achieved with the first SVD transformation. We can still identify three well-defined clusters, plus a noisier one. Even if the explained variance was significantly increased, this result tells us that the final descriptive capability of our model is fairly robust to the retained explained variance.

Indeed, further improvements should rather be searched in different steps of the pipeline, such as the TFIDF generation (e.g. the choice of the stopwords, the use of thresholds for the maximum and minimum document frequency) or the clustering algorithm.