Laboratorio 2: NLP para Ciberseguridad

Camilo A. Martínez, Daniel Rambaut

Gathering

Dado el momento crucial de las elecciones presidenciales de 2022 en Colombia, se propone trabajar sobre información que tenga que ver con el candidato Gustavo Petro, ya que este fue amenazado por un grupo paramilitar y su vida corre riesgo.

- **Motivo:** Existen diversos grupos que NO muestran afinidad con el candidato mencionado. Dentro de estos grupos encontramos grupos políticos con una tendencia política opuesta, grupos paramilitares, grupos empresariales, empresarios, periodistas, entre otros. Algunas de estas organizaciones están en total desacuerdo, por diversas razones, con una eventual llegada a la presidencia de Petro, por lo que se ven campañas de desprestigio contra este candidato. Además, existen rumores de planes para asesinarlo.
- **Oportunidad:** El candidato Petro puede tener vulnerabilidades en su esquema de seguridad, personas infiltradas en su campaña, infraestructura cibernética con vulnerabilidades, entre otras.
- **Medio:** Distintos tipos de personas pueden ser instrumentalizadas para cometer delitos cibernéticos (promoción del odio, desprestigio, hackeo, manipulación de la información), o incluso ataques físicos ya sea a personas o lugares o espacios de campaña.

Para la búsqueda de información se usó la herramienta TAGS para recopilar tweets durante un intervalo de 7 días. Este intervalo está limitado por la aplicación.

Los parámetros de búsqueda usados fueron:

- **Enter term:** petro
- **Period:** -7
- **Follower count filter:** 10
- **Number of tweets:** 18000
- **Type:** search/tweets

Con estos argumentos se logró recopilar 17244 tweets, durante las fechas de 06/05/2022 23:06:27 y 07/05/2022 00:59:59.

Se usó el término de búsqueda *petro* para recopilar suficientes tweets. Búsquedas con otro tipo de términos arrojaban pocos tweets. Como la búsqueda se hace con este sencillo y general término, esta puede generar spam, así que establecemos un número mínimo de seguidores que una persona debe tener para ser incluida en el archivo. En este caso usamos de manera empírica 10.

Lectura de datos

El primer paso es leer los datos. Para esto leemos particularmente la hoja de Excel ”Archive” de nuestro documentos “TAGS_petro.xlsx”.

```
filepath = 'TAGS_petro.xlsx'
data = pd.read_excel(filepath,sheet_name="Archive") # Importing the database (tweets)
```

Python

```
data.head() # Returns the first rows of the database
```

Python

	id_str	from_user	text	created_at	time	geo_coordinates	user_lang	in_reply_to
0	1522727911144534016	elanticrocs	@BarbasTattoo Yo no me amargó ; no ando en esa...	Fri May 06 23:59:59 +0000 2022	2022-05-07 00:59:59	NaN	NaN	
1	1522727907768025089	gduquej	RT @AlvaroUribeVel: La violencia asesina, dest...	Fri May 06 23:59:58 +0000 2022	2022-05-07 00:59:58	NaN	NaN	
2	1522727907101220867	Artemis909090	@natiibedoya No sabe si no hablar de Petro hab...	Fri May 06 23:59:58 +0000 2022	2022-05-07 00:59:58	NaN	NaN	
3	1522727906073583623	Jahpal3	RT @jarizabaletaf: Analicen los últimos trinos...	Fri May 06 23:59:58 +0000 2022	2022-05-07 00:59:58	NaN	NaN	
4	1522727906014814210	javipatino1	RT @Danielbricen: 4. Gustavo Petro no es capaz...	Fri May 06 23:59:58 +0000 2022	2022-05-07 00:59:58	NaN	NaN	

```
# converting to list the column "text" where it contains the tweets
cols = ['from_user', 'text']
text = data[cols].values.tolist()
```

Python

1. Preprocesamiento

Aquí definimos y usamos funciones que hacen la limpieza de los datos.

a. Remover URL links, @mention, #hashtags

```python
def strip_links(text): # function to remove/strip URL links
    link_regular_expression = re.compile('((https?):((//)|(\\\\))+([\w\d:#@%/;$()~_?\+-=\\\.&](#!)?)*)', r
    links = re.findall(link_regular_expression, text)
    for link in links:
        text = text.replace(link[0], ', ')
    return text
```

Python

```python
def strip_all_entities(text): # function to remove/strip mentions, hashtags, characters from some users
    entity_prefixes = ['@','#','\\','_']
    for separator in  string.punctuation:
        if separator not in entity_prefixes :
            text = text.replace(separator,' ')
    words = []
    for word in text.split():
        word = word.strip()
        if word:
            if word[0] not in entity_prefixes:
                words.append(word)
    return ' '.join(words)
```

Python

source: https://stackoverflow.com/questions/8376691/how-to-remove-hashtag-user-link-of-a-tweet-using-regular-expression

```python
def strip_all(list_text): # function that removes all URL links, @mention, #hashtags
    list_stripped = []
    for t in list_text:
        word = strip_all_entities(strip_links(t))
        list_stripped.append(word)
    return(list_stripped)
```

Python

```python
tweets = strip_all([text[i][1] for i in range(len(text))])
text = [[users[i],tweets[i]] for i in range(len(users))]
text
```

Python

```python
tweets = strip_all([text[i][1] for i in range(len(text))])
text = [[users[i],tweets[i]] for i in range(len(users))]
text
```

Python

```
Output exceeds the size limit. Open the full output data in a text editor
[['elanticrocs',
  'Yo no me amargó no ando en esas dinámicas salgo converso con amigos y vecinos y trato de conseguir votos
para Gustavo Petro presidente en primera vuelta y Francia Marquez vicepresidenta'],
 ['gduquej',
  'RT La violencia asesina destruye la familia con el microtráfico y exige votar por Petro Vuelve y juega'],
 ['Artemis909090',
  'No sabe si no hablar de Petro hable de las propuestas de Rico y de los beneficios bde Duque para el
país'],
 ['Jahpal3',
  'RT Analicen los últimos trinos de Petro un ganador no tendría esa actitud Está desquiciado porque sabe
que se aproxima su tercera derrota'],
 ['javipatino1',
  'RT 4 Gustavo Petro no es capaz de decir que Hugo Chávez fue un dictador'],
 ['Stephy3pa',
  'RT Hoy Nuevamente es recibido en pista del Aeropuerto de Cúcuta Gustavo Petro por el excomandante de la
FARC Rubén Zamora frente 33 Quienes se atribuyeron el atentado en Bogotá donde mueren los dos niños días
atrás RT'],
 ['sergio_robledo1', 'RT Lo dijo Petro hace más de una semana…'],
```

b. Convertir tweets a minúsculas.

```python
def lower_case(list_text): # Convert lowercase tweets for language processing
    list_lower_case = []
    for i in list_text:
        word = i.strip()
        new_word = word.lower()
        list_lower_case.append(new_word)
    return(list_lower_case)
```

Python

```python
tweets = lower_case([text[i][1] for i in range(len(text))])
text = [[users[i],tweets[i]] for i in range(len(users))]
text
```

Python

```
Output exceeds the size limit. Open the full output data in a text editor
[['elanticrocs',
  'yo no me amargó no ando en esas dinámicas salgo converso con amigos y vecinos y trato de conseguir votos
para gustavo petro presidente en primera vuelta y francia marquez vicepresidenta'],
 ['gduquej',
  'rt la violencia asesina destruye la familia con el microtráfico y exige votar por petro vuelve y juega'],
 ['Artemis909090',
  'no sabe si no hablar de petro hable de las propuestas de rico y de los beneficios bde duque para el
país'],
 ['Jahpal3',
  'rt analicen los últimos trinos de petro un ganador no tendría esa actitud está desquiciado porque sabe
que se aproxima su tercera derrota'],
 ['javipatino1',
  'rt 4 gustavo petro no es capaz de decir que hugo chávez fue un dictador'],
 ['Stephy3pa',
  'rt hoy nuevamente es recibido en pista del aeropuerto de cúcuta gustavo petro por el excomandante de la
farc rubén zamora frente 33 quienes se atribuyeron el atentado en bogotá donde mueren los dos niños días
atrás rt'],
 ['sergio_robledo1', 'rt lo dijo petro hace más de una semana…'],
```

c. Remplazar emoticones por sus nombres.

```python
def replace_name_emoji(list_text): # function that replaces emoticons by their names
    list_name_emoji = []
    for l in list_text:
        name_emoji = emoji.demojize(l, delimiters=(' ', ' '))
        list_name_emoji.append(name_emoji)
    return(list_name_emoji)
```

Python

```python
tweets = replace_name_emoji([text[i][1] for i in range(len(text))])
tweets = [tweets[i].replace("_"," ") for i in range(len(tweets))]
text = [[users[i],tweets[i]] for i in range(len(users))]
text
```

Python

```
Output exceeds the size limit. Open the full output data in a text editor
[['elanticrocs',
  'yo no me amargó no ando en esas dinámicas salgo converso con amigos y vecinos y trato de conseguir votos
para gustavo petro presidente en primera vuelta y francia marquez vicepresidenta'],
 ['gduquej',
  'rt la violencia asesina destruye la familia con el microtráfico y exige votar por petro vuelve y juega'],
 ['Artemis909090',
  'no sabe si no hablar de petro hable de las propuestas de rico y de los beneficios bde duque para el
país'],
 ['Jahpal3',
  'rt analicen los últimos trinos de petro un ganador no tendría esa actitud está desquiciado porque sabe
que se aproxima su tercera derrota'],
 ['javipatino1',
  'rt 4 gustavo petro no es capaz de decir que hugo chávez fue un dictador'],
 ['Stephy3pa',
  'rt hoy nuevamente es recibido en pista del aeropuerto de cúcuta gustavo petro por el excomandante de la
farc rubén zamora frente 33 quienes se atribuyeron el atentado en bogotá donde mueren los dos niños días
atrás rt'],
 ['sergio_robledo1', 'rt lo dijo petro hace más de una semana…'],
```

d. Traducción de los tweets a inglés. Para esto usamos un modelo de traducción basado en transformers: Helsinki-NLP/opus-mt-es-en

```python
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM

tokenizer = AutoTokenizer.from_pretrained("Helsinki-NLP/opus-mt-es-en")
model = AutoModelForSeq2SeqLM.from_pretrained("Helsinki-NLP/opus-mt-es-en").to(device)
```

Python

```python
tweets = [text[i][1] for i in range(len(text))]
tweets
```

Python

```python
test_tweets = []

character_count = 0
translated_tweets = 0

for f in tweets:
    batch = tokenizer([f], return_tensors="pt").to(device)
    generated_ids = model.generate(**batch)
    tweet_translated = tokenizer.batch_decode(generated_ids, skip_special_tokens=True)[0]

    test_tweets.append(tweet_translated)

    translated_tweets += 1
    character_count += len(f)
    pp = (translated_tweets/len(tweets))*100
    if translated_tweets%100 == 0:
        print(round(pp,2), '%')
```

Python

```python
print("Total characters translated: {0}".format(character_count))
print("Translated Tweets: {0}".format(translated_tweets))
```

Python

```
Total characters translated: 2681767
Translated Tweets: 17244
```

```python
text = [[users[i],test_tweets[i]] for i in range(len(users))]
text
```

Python

```
Output exceeds the size limit. Open the full output data in a text editor
[['elanticrocs',
  "I didn't tie myself up not going in those dynamics I go out with friends and neighbors and try to get votes for petro-like president in first round and france marquez vice president"],
 ['gduquej',
  'rt murderous violence destroys the family with microtrafficking and demands to vote for petro comes back and plays'],
 ['Artemis909090',
  "he doesn't know if not to talk about petro talk about the proposals of rich and the benefits bde duke for the country"],
 ['Jahpal3',
  "rt analyze the last petro trills a winner wouldn't have that attitude is deranged because he knows his third defeat is approaching"],
 ['javipatino1',
  'rt 4 petro gustavo is not able to say that hugo chávez was a dictator'],
 ['Stephy3pa',
  'rt is again received on the runway of the airport of cucuta gustavo petro by the former commander of the farc rubén zamora front 33 who were attributed the attack in Bogota where the two children die days ago rt'],
 ['sergio_robledo1', 'Rt said petro more than a week ago...'],
```

e. Guardamos los tweets traducidos y luego eliminamos líneas que quedaron en blanco.

```python
# exporting translated tweets
base = pd.DataFrame(text, columns=["users","tweets"])
base.to_excel('test.xlsx')
```

Python

```python
df = pd.read_excel("test.xlsx")
df.to_string(index=False)
nan_value = float("nan")

df.replace("", nan_value, inplace=True)

df.dropna(subset = ["tweets"], inplace=True)
df.reset_index(drop=True, inplace=True)

df.head()
```

Python

	Unnamed: 0	users	tweets
0	0	elanticrocs	I didn't tie myself up not going in those dyna...
1	1	gduquej	rt murderous violence destroys the family with...
2	2	Artemis909090	he doesn't know if not to talk about petro tal...
3	3	Jahpal3	rt analyze the last petro trills a winner woul...
4	4	javipatino1	rt 4 petro gustavo is not able to say that hug...

f. Guardamos lo datos ya preprocesados y traducidos.

```python
def write_excel(tab,name):
    writer = pd.ExcelWriter(name+'.xlsx', engine='xlsxwriter')
    # Convert the dataframe to an XlsxWriter Excel object.
    tab.to_excel(writer, sheet_name='Sheet1',index=False)
    # Close the Pandas Excel writer and output the Excel file.
    writer.save()
```

Python

```python
cols = ['users', 'tweets']
df_eng = df[cols]
df_eng.head()
```

Python

	users	tweets
0	elanticrocs	I didn't tie myself up not going in those dyna...
1	gduquej	rt murderous violence destroys the family with...
2	Artemis909090	he doesn't know if not to talk about petro tal...
3	Jahpal3	rt analyze the last petro trills a winner woul...
4	javipatino1	rt 4 petro gustavo is not able to say that hug...

```python
write_excel(df_eng, 'users_tweets_eng')
```

Python

2. Matriz de similaridad

Para realizar la matriz de similaridad primero debemos hacer un embedding de los tweets, palabra a palabra. Para esto usaremos el embedding *GoogleNews-vectors-negative300.bin,* mediante la librería *gensim*.

```python
import gensim

#wv_embedding is the embedding loaded
wv_embeddings = gensim.models.KeyedVectors.load_word2vec_format('GoogleNews-vectors-negative300.bin', bina
type(wv_embeddings )
```

✓ 21.1s Python

```
gensim.models.keyedvectors.KeyedVectors
```

From word to text embedding

```python
#This function converts a question in a vector
import numpy as np
def tweets_to_vec(tweet, embeddings, dim=300):

    result = np.zeros(dim) #300 dimensional vector for phrase vector
    cnt = 0
    words = tweet.split()
    for word in words: #All word vectors composing the phrase are summed
        if word in embeddings:
            result += np.array(embeddings[word])
            cnt += 1
    if cnt != 0: #This would happen if no word was found in the embedding
        result /= cnt
    return result
```

✓ 0.1s Python

```python
tweets2vec = []
for tweet in test_tweets:
    tweet = tweet.strip()
    answer = tweets_to_vec(tweet, wv_embeddings)
    tweets2vec.append(answer)
```

✓ 9.4s Python

```python
tweets2vec
```

✓ 9.4s Python

```
Output exceeds the size limit. Open the full output data in a text editor
[array([ 4.81341327e-02, -1.19990596e-02,  7.08098235e-02,  9.98456037e-02,
        -3.41887297e-02, -5.79065394e-02,  7.01565213e-03, -1.11234312e-01,
         5.37018953e-02,  3.41506534e-02, -4.91813377e-02, -1.15347403e-01,
        -5.34622758e-02,  1.03194625e-02, -9.30677343e-02,  7.35767506e-02,
         1.10052038e-01,  9.04134115e-02,  4.13547092e-02, -6.14657932e-02,
        -3.56716580e-02,  6.53844763e-02,  7.21039949e-02, -3.72924805e-02,
         4.05262135e-02, -3.42124656e-02, -1.18876704e-01, -7.43046513e-03,
        -1.41104239e-02, -1.96442781e-02,  1.02231061e-02,  7.55570023e-02,
        -5.41336625e-02, -3.53144893e-02,  3.34020544e-02,  4.02238634e-02,
         1.14321532e-02,  3.55586299e-03,  4.41662824e-02,  9.97879593e-02,
         1.17838542e-01, -8.65455910e-02,  1.17684823e-01,  4.12326389e-03,
        -5.55010195e-02,  3.26775445e-02,  7.59096499e-03, -4.23154478e-02,
         2.54313151e-04,  6.92824611e-02, -2.51792625e-02,  9.10780165e-02,
         2.00240524e-02,  8.02838361e-03, -3.47968207e-02,  1.83438902e-02,
        -5.62450268e-02, -2.09147135e-02,  5.24167661e-02, -1.05484574e-01,
        -3.95372179e-03,  5.90424714e-02, -9.79116934e-02, -6.16319444e-02,
        -4.49309172e-02, -5.69435402e-03, -3.35196036e-02,  1.01105867e-01,
         3.11098452e-02,  1.09185113e-01,  1.20208740e-01,  3.13878942e-02,
         6.31713867e-02,  2.39919027e-02, -1.74189815e-01, -7.18270761e-02,
         2.21840187e-02,  8.00577799e-02,  7.69359447e-02,  5.86841725e-02,
         7.68590856e-05, -3.89042607e-02,  4.29992676e-02,  2.65525535e-02,
         4.40809462e-04, -3.35874204e-02, -1.02906404e-01,  8.72780129e-02,
        -1.64043285e-02,  4.77125380e-02,  5.84016023e-02,  4.17435258e-02,
        -6.85040509e-02, -7.02898944e-02, -5.58449074e-02, -6.88188341e-02,
         2.32543945e-02, -7.14789497e-03, -1.32943613e-02, -1.37601782e-02,
...
```

Luego calculamos la matriz de similaridad por coseno

```
from sklearn.metrics.pairwise import cosine_similarity
```
✓ 0.4s Python

```
cosines = cosine_similarity(tweets2vec)
```
✓ 6.5s Python

```
cosines
```
✓ 0.1s Python

```
array([[1.        , 0.57276428, 0.70623251, ..., 0.70967679, 0.59884393,
        0.63976405],
       [0.57276428, 1.        , 0.64726427, ..., 0.6313542 , 0.72565841,
        0.66534617],
       [0.70623251, 0.64726427, 1.        , ..., 0.69820544, 0.729907  ,
        0.72265548],
       ...,
       [0.70967679, 0.6313542 , 0.69820544, ..., 1.        , 0.66561522,
        0.67592446],
       [0.59884393, 0.72565841, 0.729907  , ..., 0.66561522, 1.        ,
        0.74194331],
       [0.63976405, 0.66534617, 0.72265548, ..., 0.67592446, 0.74194331,
        1.        ]])
```

```
cosines.shape
```
✓ 0.1s Python

```
(17242, 17242)
```

3. Clusterización

Para encontrar posibles clusters empleando el algoritmo de Kmeans, en especial el MiniBatchKMeans ya que es un poco más rápido.

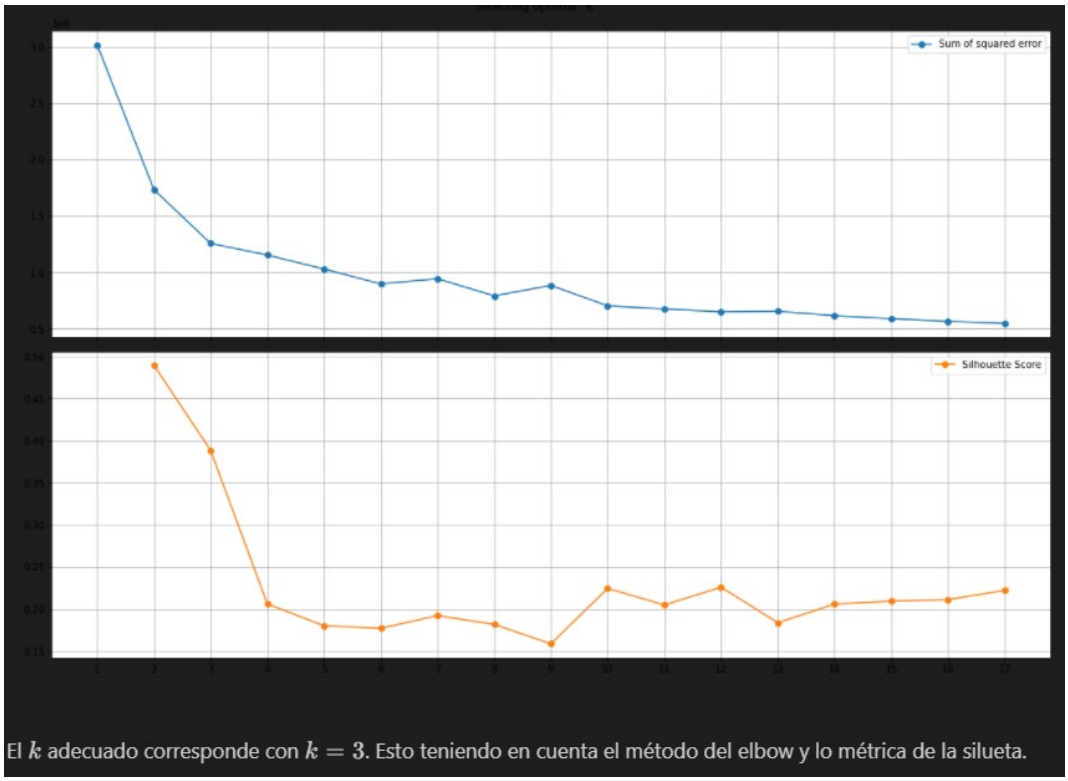
Probamos con diferentes valores para luego elegir el k optimo empleando el criterio del elbow y teniendo en cuanta también la métrica de la silueta.

Selección del numero de clusters k

```python
from sklearn.cluster import MiniBatchKMeans
from sklearn import metrics

search_range = range(1, 18)
report = {}
for k in search_range:
    print('k =', k)
    temp_dict = {}
    kmeans = MiniBatchKMeans(n_clusters=k,
                             max_iter=500,
                             random_state=42,
                             batch_size=2048).fit(cosines)
    inertia = kmeans.inertia_
    temp_dict['Sum of squared error'] = inertia
    try:
        cluster = kmeans.predict(cosines)
        ss = metrics.silhouette_score(cosines, cluster)
        temp_dict['Silhouette Score'] = ss
        report[k] = temp_dict
    except:
        report[k] = temp_dict
```

Python



El k adecuado corresponde con $k = 3$. Esto teniendo en cuenta el método del elbow y lo métrica de la silueta.

Ya encontrado el valor de k óptimo. Definimos un modelo de kmeans para k = 3, ajustado a nuestros datos.

Clustering con KMeans

```
from sklearn.cluster import KMeans

k = 3
kmeans = KMeans(n_clusters=k,
                max_iter=500,
                random_state=42).fit(cosines)
kmeans.labels_
```

✓ 33.9s Python

```
array([2, 0, 2, ..., 2, 2, 2])
```

Luego usamos PCA sobre nuestros datos, usando 2 componentes principales, para graficarlos.

```
import numpy as np
from sklearn.decomposition import PCA

pca = PCA(n_components=2)
pca.fit(cosines)
X = pca.transform(cosines)
```

✓ 13.1s Python

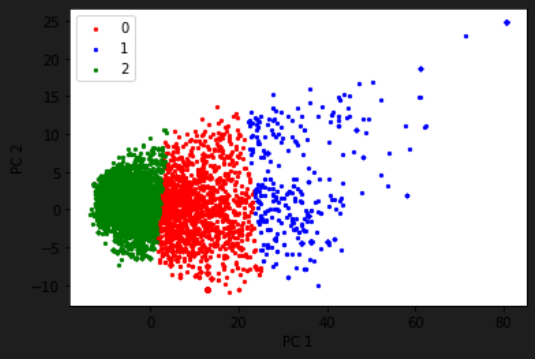
```
np.unique(kmeans.labels_)
```

✓ 0.2s Python

```
array([0, 1, 2])
```

```
cdict = {0: 'red', 1: 'blue', 2: 'green'}
for g in np.unique(kmeans.labels_):
    ix = np.where(kmeans.labels_ == g)
    plt.scatter(X[ix,0], X[ix,1], c = cdict[g], label = g, s = 5)
plt.xlabel('PC 1')
plt.ylabel('PC 2')
plt.legend()
plt.show()
```

✓ 1.5s Python



Seleccionamos *n* muestras aleatorias para cada cluster con el fin de conocer cuál es la tendencia de opinión de cada uno.

```python
import random
test_tweets_nparray = np.array(test_tweets)
rand_sample = {0:[], 1:[], 2:[]}

# Seleccionamos n muestras aleatorias de cada cluster
n = 5
for g in np.unique(kmeans.labels_):
    ix = np.where(kmeans.labels_ == g)
    ix = list(ix[0])
    ix_rand = random.sample(ix, n)
    rand_sample[g].append(test_tweets_nparray[ix_rand])
```

Python

```python
rand_sample
```

✓ 0.6s Python

```
{0: [array(['the government plan that they have ordered petro thinking face anxious face with sweet face
vomiting nauseated face',
        'rt the carvajal chicken wants to talk to colombia about petro and state secrets',
        'petro all right at home',
        'rt person q vote for petro in 1st or 2nd round is complicit d corrupt criminals terrorists narco
kidnappers murderers rapists',
        "That they don't like it I understand and it's good but that they like it don't fuck that it's hard
and pure mamerism of people who lack ethical values and moral principles"],
       dtype='<U1045')],
 1: [array(['Ha ha ha ha ha ha now petro is paramilitary ha ha ha ha what to read for god',
        'rt teachings of petro and saints',
        'petro eu só comprar cuando ela cai mais de 10 em um dia thumbs up handshake',
        'rt Senator Meisel responds to petro',
        'rt alias otoniel 128 petro processes 591 processes calculate'],
       dtype='<U1045')],
 2: [array(['Rt Zapeiro very machito to meddle with petro but very cowardly to face the clan of the gulf
that man no good',
        'Although there was a lot of pettyness on the part of political sectors opposed to their former
adherents petrism, they also changed their minds as soon as they knew the characteristics of their
leadership.',
        'rt they go marbelle and fernanda fernanda full if it wins petro it is necessary to take advantage
of such promotion',
        "rt can show them petro stealing banks and still their fans are going to say that the other one
stole more and that's how to argue with kids are fools who don't care about the truth they want revenge and
be robbed by others that if they're then seeing who to blame and who to ask",
        'rt everything should have in your account of 5890000 followers I follow all the voters of and I
expect the followed back smiling face with open hands we are world power of life 1 give like to the net
heart 2 da repeat button 4 comment petro pdte memo 3 follow me and'],
       dtype='<U1045')]}
```

```
rand_sample
✓ 0.9s                                                                    Python
{0: [array(['rt 2 the Vegan registrar admitted that “savoury petro incited his electoral witnesses to
pressure voting jurors to misfill the boxes of the forms and 14 on March 13”',
        'rt that little girl looks like the petro daughter',
        'rt murderous violence destroys the family with microtrafficking and demands to vote for petro
comes back and plays',
        'rt “murderous violence destroys the family with microtrafficking and demands to vote for petro”
alvaro uribe',
        'My great-grandmother saying that petro is going to fix the country face with spiral eyes woozy
face melting face yawning face is like severe shock of emotions djdjsjsjs'],
       dtype='<U1045')],
 1: [array(['Liar petro', 'petro', "petro's daughter is re pretty",
        'rt my candidate is petro-like', 'Rt 3 mandarins in 2 thousand.'],
       dtype='<U1045')],
 2: [array(['rt the expenses of the presidential campaigns petro 6 200 million and fific 398 million have
been announced being the lowest of all the candidates so that we realize what is the historical joint paying
buses of people to fill squares and the first line to sabotage',
        "I'm sure most Colombians already have the frequent images of the despicable drunk and possibly
drugged petro. Let's ask ourselves that is the future president. We deserve the answer, not for God's
sake.",
        'rt petro sends his sarnosos and to the cage of mad women to make me intelligence and to publish
fallacies and infamies do not chicken me “petrocelli” you are a cynical fuck',
        "Rt over here I'll leave you to think gilbert tobon the new idol of petrism on petro",
        'Rt Zapeiro very machito to meddle with petro but very cowardly to face the clan of the gulf that
man no good'],
       dtype='<U1045')]}
```

Podemos observar según nuestro criterio que el cluster más violento corresponde al cluster 0. Esto se puede observar al ver las muestras aleatorias de tweets de cada cluster.