# Detección de tópicos por tweet en threads

## Importacion de librerias y definicion de funciones

Se importan las librerias necesarias y se definen las funciones para tokenizar, lemmatizar y para preparar el texto

En la tokenizacion se eliminan los hashtags y los usuarios citados

In [68]:

```
import os
```

In [69]:

```
import spacy
spacy.load('en')

from spacy.lang.en import English
parser = English()

def tokenize(text):
    lda_tokens = []
    tokens = parser(text)
    for token in tokens:
        if token.orth_.isspace():
            continue
        elif token.like_url:
            continue
        elif token.orth_.startswith('#'):
            continue
        elif token.orth_.startswith('@'):
            continue
        else:
            lda_tokens.append(token.lower_)
    return lda_tokens

import nltk

nltk.download('wordnet')
from nltk.corpus import wordnet as wn
def get_lemma(word):
    lemma = wn.morphy(word)
    if lemma is None:
        return word
    else:
        return lemma

from nltk.stem.wordnet import WordNetLemmatizer
def get_lemma2(word):
    return WordNetLemmatizer().lemmatize(word)

nltk.download('stopwords')
en_stop = set(nltk.corpus.stopwords.words('english'))

def prepare_text_for_lda(text):
    tokens = tokenize(text)
    tokens = [token for token in tokens if len(token) > 3]
    tokens = [token for token in tokens if token not in en_stop]
    tokens = [get_lemma(token) for token in tokens]
    return tokens
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\carlo\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\carlo\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

# Importacion de archivos

Se cargan los archivos csv y se agrupan los tweets por threads, para luego crear un diccionario de tweets por cad

In [70]:

```
import random
import pandas as pd
```

In [71]:

```
csv1 = pd.read_csv('five_ten.csv', encoding='iso-8859-1')
csv1_grouped_by_thread = csv1.groupby(['thread_number'])
threads1 = {}
documentos1 = []

csv2 = pd.read_csv('ten_fifteen.csv', encoding='iso-8859-1')
csv2_grouped_by_thread = csv2.groupby(['thread_number'])
threads2 = {}
documentos2 = []

csv3 = pd.read_csv('fifteen_twenty.csv', encoding='iso-8859-1')
csv3_grouped_by_thread = csv3.groupby(['thread_number'])
threads3 = {}
documentos3 = []

csv4 = pd.read_csv('twenty_twentyfive.csv', encoding='iso-8859-1')
csv4_grouped_by_thread = csv4.groupby(['thread_number'])
threads4 = {}
documentos4 = []

csv5 = pd.read_csv('twentyfive_thirty.csv', encoding='iso-8859-1')
csv5_grouped_by_thread = csv5.groupby(['thread_number'])
threads5 = {}
documentos5 = []
```

## Creación de diccionario de tweets por threads

Se agruparán los tweets de cada hilo en un diccionario para cada archivo.

In [72]:

```
for thread, data in dict(list(csv1_grouped_by_thread)).items():
    threads1[thread] = list(data['text'])



for thread, data in dict(list(csv2_grouped_by_thread)).items():
    threads2[thread] = list(data['text'])



for thread, data in dict(list(csv3_grouped_by_thread)).items():
    threads3[thread] = list(data['text'])



for thread, data in dict(list(csv4_grouped_by_thread)).items():
    threads4[thread] = list(data['text'])



for thread, data in dict(list(csv5_grouped_by_thread)).items():
    threads5[thread] = list(data['text'])
```

In [95]:

```
threads1
```

Out[95]:

```
{'Thread 1': 'Extraordinary evidence at Treasury committee from Jon Thompson, CEO of HMRC on
e Brexiter favourite Max Fac - would cost business between Â£17 and Â£20bn a year\r\r\n\r\r\
t.co/0MwIcwre4t) \n How does he arrive at the figure\r\r\n\r\r\n200m export consignments at
O (https://t.co/KxnkU2QiVO) \n Theresa May\'s New Customs Partnership is much cheaper for bu
H (https://t.co/0LcsJHah0H) \n Mr Thompson said he did not expect the EU to reciprocate over
9c3uhhnZGX (https://t.co/9c3uhhnZGX) \n Both would not be ready by 2021. Max Fac needs 3 yea
s://t.co/luLzgUsiR4 (https://t.co/luLzgUsiR4) \n "We think we can manage the risk - we think
orâ?¦ https://t.co/Ti1nbbjfpU', (https://t.co/Ti1nbbjfpU',)
 'Thread 10': 'Here are some real ideas to fix the #immigration system.\r\r\n\r\r\n1. Get ri
s://t.co/DzShYf2Kn1) \n 5. Create a path to a green card for E-2 investors. Include any chil
 (https://t.co/jZ8DoSvRxk) \n 9. Put money and resources into fraud detection.\r\r\n10. Expa
udrK (https://t.co/nzebluudrK) \n 15. Expand ESL instruction.\r\r\n16. Create state Offices
co/twEacdaoIS (https://t.co/twEacdaoIS) \n 21. Survey world immigration laws and sample the
s://t.co/0dFy5bTCa0 (https://t.co/0dFy5bTCa0) \n 24. Vote out elected officials with close t
s://t.co/eLNeyaLIRH (https://t.co/eLNeyaLIRH) \n So much can be done. Yet we\'re stuck betwe
s://t.co/0mfAc6mUsf (https://t.co/0mfAc6mUsf) \n And then whining that "illegal immigration
Oâ?¦ https://t.co/cUUOKHEwv3', (https://t.co/cUUOKHEwv3',)
```

# LDA para cada thread de cada CSV

Se definen la cantidad de topicos a detectar, en conjunto con la cantidad de palabras que se mostraran al imprimi

La detección de tópicos se realizará a cada thread de todos los archivos CSV, por lo que se considerará cada twe

In [74]:

```
import gensim
from gensim import corpora
NUM_TOPICS = 5
NUM_WORDS = 5
import pickle
```

## CSV five_ten

In [75]:

```
THIS_FOLDER = os.getcwd()
threads_leer = threads1
carpeta_guardar = "tpcsv1"

#Poblar text_data

for hilos in threads_leer:
    camino = os.path.join(THIS_FOLDER, carpeta_guardar)
    text_data = []
    documentos = []
    dictionary = []
    corpus = []
    print(hilos)
    documentos = threads_leer[hilos]

    #print(documentos)

    for line in documentos:
        #print(line)
        tokens = prepare_text_for_lda(line)
        if random.random() > .009:
            #print(tokens)
            text_data.append(tokens)

    #print(text_data)
    NDIC = camino+"\\"+hilos+"_t_dictionary1.gensim"
    NMOD = camino+"\\"+hilos+"_t_model1.gensim"
    NCOR = camino+"\\"+hilos+"_t_corpus1.pkl"
    dictionary = corpora.Dictionary(text_data)
    corpus = [dictionary.doc2bow(text) for text in text_data]
    pickle.dump(corpus, open(NCOR, 'wb'))
    dictionary.save(NDIC)

    ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics = NUM_TOPICS, id2word=dict
    ldamodel.save(NMOD)
    topics = ldamodel.print_topics(num_words=NUM_WORDS)
    for topic in topics:
        print(topic)
```

```
Thread 1
(0, '0.068*"thompson" + 0.068*"customs" + 0.046*"partnership" + 0.046*"say" + 0.025*"ready"'
(1, '0.056*"cost" + 0.056*"partnership" + 0.056*"customs" + 0.056*"almost" + 0.056*"business
(2, '0.059*"cost" + 0.059*"would" + 0.059*"business" + 0.059*"almost" + 0.059*"jusâ"')
(3, '0.113*"think" + 0.062*"say" + 0.062*"sure" + 0.062*"backdoorâ" + 0.062*"sound"')
(4, '0.019*"cost" + 0.019*"say" + 0.019*"partnership" + 0.019*"customs" + 0.019*"would"')
Thread 10
(0, '0.052*"money" + 0.052*"immigration" + 0.052*"things" + 0.052*"expensive" + 0.052*"spend
(1, '0.048*"white" + 0.048*"fairâ" + 0.048*"nativist" + 0.048*"fearmongering" + 0.048*"offic
(2, '0.035*"expand" + 0.035*"worker" + 0.035*"guest" + 0.035*"detection" + 0.035*"fraud"')
(3, '0.040*"real" + 0.040*"currentâ" + 0.040*"daca" + 0.040*"gutting" + 0.040*"much"')
(4, '0.059*"immigration" + 0.059*"create" + 0.032*"model" + 0.032*"give" + 0.032*"world"')
Thread 100
(0, '0.066*"trump" + 0.036*"ways" + 0.036*"billionaire" + 0.036*"everyone" + 0.036*"panoply"
(1, '0.087*"american" + 0.087*"every" + 0.087*"know" + 0.048*"happen" + 0.048*"want"')
(2, '0.082*"data" + 0.056*"happen" + 0.056*"breach" + 0.031*"deny" + 0.031*"votersâ"')
(3, '0.054*"mercer" + 0.054*"fuel" + 0.054*"resultâ" + 0.054*"assiduous" + 0.054*"genius"')
(4, '0.050*"reveal" + 0.050*"pundit" + 0.050*"reason" + 0.050*"assert" + 0.050*"information"
```

## CSV ten_fifteen

In [76]:

```
THIS_FOLDER = os.getcwd()
threads_leer = threads2
carpeta_guardar = "tpcsv2"

#Poblar text_data

for hilos in threads_leer:
    camino = os.path.join(THIS_FOLDER, carpeta_guardar)
    text_data = []
    documentos = []
    dictionary = []
    corpus = []
    print(hilos)
    documentos = threads_leer[hilos]

    #print(documentos)

    for line in documentos:
        #print(line)
        tokens = prepare_text_for_lda(line)
        if random.random() > .009:
            #print(tokens)
            text_data.append(tokens)

    #print(text_data)
    NDIC = camino+"\\"+hilos+"_t_dictionary1.gensim"
    NMOD = camino+"\\"+hilos+"_t_model1.gensim"
    NCOR = camino+"\\"+hilos+"_t_corpus1.pkl"
    dictionary = corpora.Dictionary(text_data)
    corpus = [dictionary.doc2bow(text) for text in text_data]
    pickle.dump(corpus, open(NCOR, 'wb'))
    dictionary.save(NDIC)

    ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics = NUM_TOPICS, id2word=dict
    ldamodel.save(NMOD)
    topics = ldamodel.print_topics(num_words=NUM_WORDS)
    for topic in topics:
        print(topic)
```

```
Thread 1
(0, '0.077*"labour" + 0.029*"want" + 0.029*"conditionsâ" + 0.029*"cheap" + 0.029*"agency"')
(1, '0.068*"outside" + 0.068*"corbyn" + 0.046*"single" + 0.025*"future" + 0.025*"trade"')
(2, '0.062*"corbyn" + 0.033*"customs" + 0.033*"union" + 0.033*"single" + 0.033*"ensure"')
(3, '0.011*"labour" + 0.011*"corbyn" + 0.011*"means" + 0.011*"rule" + 0.011*"market"')
(4, '0.065*"corbyn" + 0.045*"rule" + 0.045*"â??we" + 0.045*"accept" + 0.024*"economic"')
Thread 10
(0, '0.071*"trump" + 0.054*"mueller" + 0.037*"probe" + 0.037*"witness" + 0.037*"russianð??·ð
(1, '0.039*"teamð??ºð??¸mueller" + 0.039*"source" + 0.039*"question" + 0.039*"focus" + 0.039
(2, '0.052*"trump" + 0.029*"campaign" + 0.029*"prospective" + 0.029*"even" + 0.029*"though"'
(3, '0.049*"miss" + 0.049*"universe" + 0.049*"moscow" + 0.049*"trump" + 0.034*"mueller"')
(4, '0.043*"trump" + 0.024*"close" + 0.024*"pal" + 0.024*"testify" + 0.024*"congress"')
Thread 11
(0, '0.012*"drop" + 0.012*"establishment" + 0.012*"advisorâ" + 0.012*"original" + 0.012*"sho
(1, '0.065*"anon" + 0.035*"confirm" + 0.035*"board" + 0.035*"research" + 0.035*"earlier"')
(2, '0.033*"tweet" + 0.033*"clown" + 0.033*"like" + 0.033*"maybe" + 0.033*"tropical"')
(3, '0.064*"post" + 0.044*"agency" + 0.044*"warning" + 0.024*"assume" + 0.024*"role"')
(4, '0.045*"post" + 0.045*"warning" + 0.024*"election" + 0.024*"/ga/" + 0.024*"fraud"')
```

## CSV fifteen_twenty

In [77]:

```
THIS_FOLDER = os.getcwd()
threads_leer = threads3
carpeta_guardar = "tpcsv3"

#Poblar text_data

for hilos in threads_leer:
    camino = os.path.join(THIS_FOLDER, carpeta_guardar)
    text_data = []
    documentos = []
    dictionary = []
    corpus = []
    print(hilos)
    documentos = threads_leer[hilos]

    #print(documentos)

    for line in documentos:
        #print(line)
        tokens = prepare_text_for_lda(line)
        if random.random() > .009:
            #print(tokens)
            text_data.append(tokens)

    #print(text_data)
    NDIC = camino+"\\"+hilos+"_t_dictionary1.gensim"
    NMOD = camino+"\\"+hilos+"_t_model1.gensim"
    NCOR = camino+"\\"+hilos+"_t_corpus1.pkl"
    dictionary = corpora.Dictionary(text_data)
    corpus = [dictionary.doc2bow(text) for text in text_data]
    pickle.dump(corpus, open(NCOR, 'wb'))
    dictionary.save(NDIC)

    ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics = NUM_TOPICS, id2word=dict
    ldamodel.save(NMOD)
    topics = ldamodel.print_topics(num_words=NUM_WORDS)
    for topic in topics:
        print(topic)
```

```
Thread 1
(0, '0.051*"meeting" + 0.051*"...." + 0.051*"news" + 0.028*"whistleblower" + 0.028*"true"')
(1, '0.072*"excite" + 0.027*"june" + 0.027*"publâ" + 0.027*"lie" + 0.027*"printing"')
(2, '0.060*"document" + 0.033*"working" + 0.033*"still" + 0.033*"continue" + 0.033*"verifica
(3, '0.036*"receive" + 0.036*"inside" + 0.036*"source" + 0.036*"network" + 0.036*"arrive"')
(4, '0.090*"news" + 0.049*"broadcaster" + 0.049*"slant" + 0.049*"distort" + 0.049*"intention
Thread 10
(0, '0.033*"trump" + 0.018*"enquãªte" + 0.018*"exigeant" + 0.018*"franchi" + 0.018*"rubicon"
(1, '0.027*"bien" + 0.027*"mois" + 0.027*"justice" + 0.027*"ministã¨re" + 0.027*"avaient"')
(2, '0.048*"trump" + 0.032*"pour" + 0.025*"breaking" + 0.017*"mueller" + 0.017*"badaboum"')
(3, '0.033*"trump" + 0.033*"cette" + 0.018*"thread" + 0.018*"2017" + 0.018*"situation"')
(4, '0.038*"dans" + 0.026*"c\'est" + 0.026*"lation" + 0.026*"trump" + 0.014*"kaboom"')
Thread 11
(0, '0.037*"well" + 0.037*"already" + 0.020*"posing" + 0.020*"tying" + 0.020*"proof"')
(1, '0.036*"black" + 0.025*"trump" + 0.025*"work" + 0.025*"stein" + 0.025*"vote"')
(2, '0.032*"putin" + 0.032*"russian" + 0.032*"state" + 0.032*"could" + 0.017*"know"')
(3, '0.052*"russian" + 0.040*"vote" + 0.027*"target" + 0.027*"troll" + 0.027*"american"')
(4, '0.041*"disinformation" + 0.041*"campaign" + 0.041*"hybridwar" + 0.023*"putin" + 0.023*"
```

## CSV twenty_twentyfive

In [78]:

```
THIS_FOLDER = os.getcwd()
threads_leer = threads4
carpeta_guardar = "tpcsv4"

#Poblar text_data

for hilos in threads_leer:
    camino = os.path.join(THIS_FOLDER, carpeta_guardar)
    text_data = []
    documentos = []
    dictionary = []
    corpus = []
    print(hilos)
    documentos = threads_leer[hilos]

    #print(documentos)

    for line in documentos:
        #print(line)
        tokens = prepare_text_for_lda(line)
        if random.random() > .009:
            #print(tokens)
            text_data.append(tokens)

    #print(text_data)
    NDIC = camino+"\\"+hilos+"_t_dictionary1.gensim"
    NMOD = camino+"\\"+hilos+"_t_model1.gensim"
    NCOR = camino+"\\"+hilos+"_t_corpus1.pkl"
    dictionary = corpora.Dictionary(text_data)
    corpus = [dictionary.doc2bow(text) for text in text_data]
    pickle.dump(corpus, open(NCOR, 'wb'))
    dictionary.save(NDIC)

    ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics = NUM_TOPICS, id2word=dict
    ldamodel.save(NMOD)
    topics = ldamodel.print_topics(num_words=NUM_WORDS)
    for topic in topics:
        print(topic)
```

```
(4, '0.029*"like" + 0.029*"leave" + 0.029*"good" + 0.029*"policy" + 0.029*"idea"')
Thread 10
(0, '0.065*"incumbent" + 0.045*"congressrun2018" + 0.025*"democratic" + 0.024*"davis" + 0.02
(1, '0.030*"contribute" + 0.030*"300,000" + 0.030*"east" + 0.030*"bloomington" + 0.030*"carb
(2, '0.051*"congressrun2018" + 0.035*"2018" + 0.035*"illinois" + 0.035*"midterms2018" + 0.03
(3, '0.072*"congressrun2018" + 0.044*"incumbent" + 0.031*"daniel" + 0.017*"hultgren" + 0.017
(4, '0.087*"congressrun2018" + 0.070*"incumbent" + 0.020*"john" + 0.020*"deter" + 0.020*"il1
Thread 100
(0, '0.052*"national" + 0.050*"anthem" + 0.049*"song" + 0.039*"banner" + 0.027*"wilson"')
(1, '0.031*"like" + 0.031*"song" + 0.017*"slavery" + 0.017*"probably" + 0.017*"sign"')
(2, '0.033*"song" + 0.030*"america" + 0.030*"beautiful" + 0.030*"pretty" + 0.018*"anthem"')
(3, '0.035*"military" + 0.035*"song" + 0.024*"would" + 0.024*"play" + 0.024*"scott"')
(4, '0.051*"song" + 0.028*"like" + 0.028*"popular" + 0.028*"country" + 0.028*"daughter"')
Thread 11
(0, '0.020*"friend" + 0.020*"democraticparty" + 0.020*"simple" + 0.011*"muslim" + 0.011*"ext
(1, '0.018*"dems" + 0.018*"inscrutable" + 0.018*"softball" + 0.018*"stand" + 0.018*"lob"')
(2, '0.018*"know" + 0.018*"theregime" + 0.018*"dreamer" + 0.018*"daca" + 0.018*"white"')
(3, '0.021*"unique" + 0.021*"must" + 0.021*"bash" + 0.021*"dictatorial" + 0.021*"suffer"')
```

## CSV twentyfive_thirty

In [79]:

```
THIS_FOLDER = os.getcwd()
threads_leer = threads5
carpeta_guardar = "tpcsv5"

#Poblar text_data

for hilos in threads_leer:
    camino = os.path.join(THIS_FOLDER, carpeta_guardar)
    text_data = []
    documentos = []
    dictionary = []
    corpus = []
    print(hilos)
    documentos = threads_leer[hilos]

    #print(documentos)

    for line in documentos:
        #print(line)
        tokens = prepare_text_for_lda(line)
        if random.random() > .009:
            #print(tokens)
            text_data.append(tokens)

    #print(text_data)
    NDIC = camino+"\\"+hilos+"_t_dictionary1.gensim"
    NMOD = camino+"\\"+hilos+"_t_model1.gensim"
    NCOR = camino+"\\"+hilos+"_t_corpus1.pkl"
    dictionary = corpora.Dictionary(text_data)
    corpus = [dictionary.doc2bow(text) for text in text_data]
    pickle.dump(corpus, open(NCOR, 'wb'))
    dictionary.save(NDIC)

    ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics = NUM_TOPICS, id2word=dict
    ldamodel.save(NMOD)
    topics = ldamodel.print_topics(num_words=NUM_WORDS)
    for topic in topics:
        print(topic)
```

```
Thread 1
(0, '0.028*"paraguay" + 0.028*"commando" + 0.028*"time" + 0.015*"brazil" + 0.015*"say"')
(1, '0.020*"attack" + 0.020*"gaza" + 0.020*"2017" + 0.020*"palace" + 0.020*"total"')
(2, '0.019*"force" + 0.019*"weapon" + 0.019*"allies" + 0.019*"kill" + 0.019*"nuclear"')
(3, '0.041*"tunnel" + 0.022*"carter" + 0.022*"jimmy" + 0.022*"north" + 0.012*"escape"')
(4, '0.041*"north" + 0.033*"korean" + 0.025*"rafaat" + 0.017*"toumani" + 0.017*"korea"')
Thread 10
(0, '0.052*"lucky" + 0.036*"naik" + 0.036*"reveal" + 0.036*"conversation" + 0.036*"phone"')
(1, '0.040*"swedish" + 0.040*"lucky" + 0.031*"model" + 0.021*"country" + 0.021*"case"')
(2, '0.034*"atala" + 0.034*"lucky" + 0.023*"druglord" + 0.023*"drug" + 0.023*"yasin"')
(3, '0.045*"indian" + 0.031*"case" + 0.031*"agent" + 0.031*"allege" + 0.031*"lucky"')
(4, '0.051*"israeli" + 0.041*"police" + 0.041*"lucky" + 0.021*"drug" + 0.021*"2010"')
Thread 100
(0, '0.036*"still" + 0.019*"concern" + 0.019*"woman" + 0.019*"voter" + 0.019*"silence"')
(1, '0.047*"activism" + 0.047*"transgender" + 0.029*"right" + 0.029*"debate" + 0.029*"articl
(2, '0.027*"identify" + 0.027*"offensive" + 0.027*"uncomfortable" + 0.027*"angry" + 0.027*"p
(3, '0.053*"transgender" + 0.028*"transform" + 0.028*"activism" + 0.015*"every" + 0.015*"ven
(4, '0.038*"woman" + 0.021*"indeed" + 0.021*"language" + 0.021*"child" + 0.021*"campaign"')
```

# Deteccion de topicos por threads

Al contrario del apartado anterior, se buscarán tópicos en el archivo completo, por lo que se considerará cada thre
por saltos de linea "\n".

In [80]:

```
string = " \n "

for thread, data in dict(list(csv1_grouped_by_thread)).items():
    threads1[thread] = string.join(list(data['text']))
Tthreads1 = list(threads1.values())

for thread, data in dict(list(csv2_grouped_by_thread)).items():
    threads2[thread] = string.join(list(data['text']))
Tthreads2 = list(threads2.values())

for thread, data in dict(list(csv3_grouped_by_thread)).items():
    threads3[thread] = string.join(list(data['text']))
Tthreads3 = list(threads3.values())

for thread, data in dict(list(csv4_grouped_by_thread)).items():
    threads4[thread] = string.join(list(data['text']))
Tthreads4 = list(threads4.values())

for thread, data in dict(list(csv5_grouped_by_thread)).items():
    threads5[thread] = string.join(list(data['text']))
Tthreads5 = list(threads5.values())
```

In [81]:

```
Tthreads1
```

Out[81]:

```
['Extraordinary evidence at Treasury committee from Jon Thompson, CEO of HMRC on customs and
avourite Max Fac - would cost business between Â£17 and Â£20bn a year\r\r\n\r\r\n- that\'s a
e4t) \n How does he arrive at the figure\r\r\n\r\r\n200m export consignments at an average c
t.co/KxnkU2QiVO) \n Theresa May\'s New Customs Partnership is much cheaper for business (alm
t.co/0LcsJHah0H) \n Mr Thompson said he did not expect the EU to reciprocate over the custom
 (https://t.co/9c3uhhnZGX) \n Both would not be ready by 2021. Max Fac needs 3 years. Custom
uLzgUsiR4 (https://t.co/luLzgUsiR4) \n "We think we can manage the risk - we think we can" h
s://t.co/Ti1nbbjfpU', (https://t.co/Ti1nbbjfpU',)
 'Here are some real ideas to fix the #immigration system.\r\r\n\r\r\n1. Get rid of the 3 an
Yf2Kn1) \n 5. Create a path to a green card for E-2 investors. Include any children brought
o/jZ8DoSvRxk) \n 9. Put money and resources into fraud detection.\r\r\n10. Expand guest work
s://t.co/nzebluudrK) \n 15. Expand ESL instruction.\r\r\n16. Create state Offices of New Ame
oIS (https://t.co/twEacdaoIS) \n 21. Survey world immigration laws and sample the best to cr
y5bTCa0 (https://t.co/0dFy5bTCa0) \n 24. Vote out elected officials with close ties to nativ
yaLIRH (https://t.co/eLNeyaLIRH) \n So much can be done. Yet we\'re stuck between a border w
c6mUsf (https://t.co/0mfAc6mUsf) \n And then whining that "illegal immigration is too expens
t.co/cUUOKHEwv3', (https://t.co/cUUOKHEwv3',)
```

In [82]:

```
from gensim import corpora
import gensim
NUM_TOPICS = 20
NUM_WORDS = 10
import pickle
```

## CSV five_ten

In [83]:

```
THIS_FOLDER = os.getcwd()
threads_leer = Tthreads1
carpeta_guardar = "Ttpcsv1"

#Poblar text_data


camino = os.path.join(THIS_FOLDER, carpeta_guardar)
text_data = []
documentos = []
dictionary = []
corpus = []
documentos = threads_leer

#print(documentos)

for line in documentos:
    #print(line)
    tokens = prepare_text_for_lda(line)
    if random.random() > .009:
        #print(tokens)
        text_data.append(tokens)

#print(text_data)
NDIC = camino+"\\t_dictionary1.gensim"
NMOD = camino+"\\t_model1.gensim"
NCOR = camino+"\\t_corpus1.pkl"
dictionary = corpora.Dictionary(text_data)
corpus = [dictionary.doc2bow(text) for text in text_data]
pickle.dump(corpus, open(NCOR, 'wb'))
dictionary.save(NDIC)

ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics = NUM_TOPICS, id2word=dictiona
ldamodel.save(NMOD)
topics = ldamodel.print_topics(num_words=NUM_WORDS)
for topic in topics:
    print(topic)
```

```
(0, '0.013*"investigation" + 0.011*"qanon" + 0.011*"security" + 0.011*"report" + 0.010*"comm
s" + 0.010*"unite"')
(1, '0.012*"smart" + 0.010*"samba" + 0.010*"kevin" + 0.010*"reynolds" + 0.010*"also" + 0.007
(2, '0.015*"indictment" + 0.015*"lard" + 0.013*"seal" + 0.011*"followthewhiterabbit" + 0.009
sethememo" + 0.007*"well"')
(3, '0.011*"woman" + 0.008*"tech" + 0.008*"kobebryant" + 0.008*"mccabe" + 0.008*"like" + 0.0
(4, '0.014*"peer" + 0.012*"trump" + 0.012*"make" + 0.012*"mkultra" + 0.009*"campaign" + 0.00
(5, '0.015*"north" + 0.015*"dakota" + 0.010*"vote" + 0.010*"northdakota" + 0.008*"candidate"
ke"')
(6, '0.016*"anons" + 0.010*"arrest" + 0.010*"believe" + 0.010*"expose" + 0.010*"post" + 0.01
(7, '0.014*"schiff" + 0.009*"trump" + 0.007*"american" + 0.007*"obama" + 0.007*"admin" + 0.0
(8, '0.023*"startup" + 0.015*"investor" + 0.012*"vous" + 0.009*"facebook" + 0.009*"pitch" +
(9, '0.016*"cambridge" + 0.014*"analytica" + 0.009*"papadopoulos" + 0.009*"julian" + 0.007*"
ytica" + 0.007*"government"')
(10, '0.092*"qanon" + 0.055*"part" + 0.026*"greatawakening" + 0.023*"thestorm" + 0.023*"inte
ring" + 0.007*"say"')
(11, '0.026*"democrat" + 0.016*"know" + 0.014*"arrest" + 0.013*"deny" + 0.006*"friend" + 0.0
(12, '0.022*"qposts" + 0.022*"2018" + 0.019*"qanon" + 0.019*"offline" + 0.019*"archive" + 0.
```

## CSV Ten_fifteen

In [84]:

```
THIS_FOLDER = os.getcwd()
threads_leer = Tthreads2
carpeta_guardar = "Ttpcsv2"

#Poblar text_data


camino = os.path.join(THIS_FOLDER, carpeta_guardar)
text_data = []
documentos = []
dictionary = []
corpus = []
documentos = threads_leer

#print(documentos)

for line in documentos:
    #print(line)
    tokens = prepare_text_for_lda(line)
    if random.random() > .009:
        #print(tokens)
        text_data.append(tokens)

#print(text_data)
NDIC = camino+"\\t_dictionary1.gensim"
NMOD = camino+"\\t_model1.gensim"
NCOR = camino+"\\t_corpus1.pkl"
dictionary = corpora.Dictionary(text_data)
corpus = [dictionary.doc2bow(text) for text in text_data]
pickle.dump(corpus, open(NCOR, 'wb'))
dictionary.save(NDIC)

ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics = NUM_TOPICS, id2word=dictiona
ldamodel.save(NMOD)
topics = ldamodel.print_topics(num_words=NUM_WORDS)
for topic in topics:
    print(topic)
```

```
(0, '0.021*"grassleymemo" + 0.015*"page" + 0.015*"declassify" + 0.012*"portion" + 0.009*"ope
*"people"')
(1, '0.011*"grassley" + 0.010*"cmte" + 0.008*"hastings" + 0.008*"know" + 0.007*"would" + 0.0
y"')
(2, '0.012*"woman" + 0.009*"northkorea" + 0.008*"memo" + 0.008*"france" + 0.008*"take" + 0.0
(3, '0.018*"trump" + 0.015*"mueller" + 0.010*"putin" + 0.009*"corbyn" + 0.007*"trumpâ??s" +
o"')
(4, '0.020*"cult" + 0.011*"words" + 0.011*"child" + 0.009*"code" + 0.009*"signal" + 0.009*"f
(5, '0.008*"memo" + 0.007*"make" + 0.006*"election" + 0.006*"news" + 0.006*"company" + 0.006
(6, '0.010*"ridicule" + 0.010*"need" + 0.006*"post" + 0.006*"phase" + 0.006*"demoralize" + 0
(7, '0.013*"care" + 0.011*"people" + 0.009*"clan" + 0.009*"land" + 0.007*"libertarian" + 0.0
(8, '0.015*"mueller" + 0.013*"facebook" + 0.007*"clinton" + 0.006*"people" + 0.006*"like" +
elete"')
(9, '0.023*"part" + 0.021*"qanon" + 0.019*"fisamemo" + 0.007*"fisa" + 0.007*"post" + 0.006*"
(10, '0.009*"stone" + 0.008*"qanon" + 0.007*"trump" + 0.007*"time" + 0.007*"roger" + 0.006*"
(11, '0.032*"qanon" + 0.032*"greatawakening" + 0.031*"thestorm" + 0.013*"oigreport" + 0.012*
e" + 0.008*"chapter" + 0.006*"work"')
(12, '0.015*"flynn" + 0.014*"cohen" + 0.010*"con\'t" + 0.010*"stone" + 0.008*"post" + 0.008*
```

## CSV fifteen_twenty

In [85]:

```
THIS_FOLDER = os.getcwd()
threads_leer = Tthreads3
carpeta_guardar = "Ttpcsv3"

#Poblar text_data


camino = os.path.join(THIS_FOLDER, carpeta_guardar)
text_data = []
documentos = []
dictionary = []
corpus = []
documentos = threads_leer

#print(documentos)

for line in documentos:
    #print(line)
    tokens = prepare_text_for_lda(line)
    if random.random() > .009:
        #print(tokens)
        text_data.append(tokens)

#print(text_data)
NDIC = camino+"\\t_dictionary1.gensim"
NMOD = camino+"\\t_model1.gensim"
NCOR = camino+"\\t_corpus1.pkl"
dictionary = corpora.Dictionary(text_data)
corpus = [dictionary.doc2bow(text) for text in text_data]
pickle.dump(corpus, open(NCOR, 'wb'))
dictionary.save(NDIC)

ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics = NUM_TOPICS, id2word=dictiona
ldamodel.save(NMOD)
topics = ldamodel.print_topics(num_words=NUM_WORDS)
for topic in topics:
    print(topic)
```

```
(0, '0.012*"iran" + 0.007*"alex" + 0.006*"trump" + 0.005*"zwaan" + 0.005*"like" + 0.005*"tel
(1, '0.010*"trump" + 0.008*"claim" + 0.007*"hindu" + 0.006*"wray" + 0.005*"would" + 0.005*"t
(2, '0.011*"trump" + 0.010*"like" + 0.008*"qanon" + 0.007*"many" + 0.006*"tesla" + 0.006*"kn
(3, '0.009*"prince" + 0.009*"harry" + 0.007*"britain" + 0.007*"ù?ù?ù?ù" + 0.007*"ù?ø§ù" + 0.
ù"')
(4, '0.017*"trump" + 0.012*"obama" + 0.008*"rabe" + 0.008*"page" + 0.006*"leak" + 0.006*"bac
(5, '0.019*"qanon" + 0.019*"student" + 0.019*"fakenewsawards" + 0.019*"bible" + 0.008*"germa
e"')
(6, '0.010*"story" + 0.010*"make" + 0.008*"even" + 0.008*"would" + 0.008*"defeat" + 0.008*"h
(7, '0.010*"clinton" + 0.009*"primary" + 0.008*"hearts" + 0.008*"scar" + 0.006*"vote" + 0.00
(8, '0.012*"company" + 0.010*"startup" + 0.010*"private" + 0.009*"time" + 0.007*"make" + 0.0
â??t"')
(9, '0.015*"democratic" + 0.014*"candidate" + 0.012*"alabama" + 0.009*"congress" + 0.006*"pr
*"american"')
(10, '0.015*"company" + 0.009*"trump" + 0.008*"sheep" + 0.007*"comey" + 0.007*"activist" + 0
*"say"')
(11, '0.012*"value" + 0.010*"mars" + 0.010*"terraforming" + 0.008*"theme" + 0.006*"land" + 0
enley"')
```

## CSV twenty_twentyfive

In [86]:

```
THIS_FOLDER = os.getcwd()
threads_leer = Tthreads4
carpeta_guardar = "Ttpcsv4"

#Poblar text_data


camino = os.path.join(THIS_FOLDER, carpeta_guardar)
text_data = []
documentos = []
dictionary = []
corpus = []
documentos = threads_leer

#print(documentos)

for line in documentos:
    #print(line)
    tokens = prepare_text_for_lda(line)
    if random.random() > .009:
        #print(tokens)
        text_data.append(tokens)

#print(text_data)
NDIC = camino+"\\t_dictionary1.gensim"
NMOD = camino+"\\t_model1.gensim"
NCOR = camino+"\\t_corpus1.pkl"
dictionary = corpora.Dictionary(text_data)
corpus = [dictionary.doc2bow(text) for text in text_data]
pickle.dump(corpus, open(NCOR, 'wb'))
dictionary.save(NDIC)

ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics = NUM_TOPICS, id2word=dictiona
ldamodel.save(NMOD)
topics = ldamodel.print_topics(num_words=NUM_WORDS)
for topic in topics:
    print(topic)
```

```
(0, '0.008*"demand" + 0.008*"iran" + 0.006*"kurd" + 0.005*"trump" + 0.005*"take" + 0.004*"at
(1, '0.039*"democratic" + 0.035*"candidate" + 0.023*"vote" + 0.021*"georgia" + 0.020*"2018"
8*"investigation"')
(2, '0.019*"trump" + 0.011*"know" + 0.008*"people" + 0.007*"make" + 0.006*"manafort" + 0.006
(3, '0.019*"incumbent" + 0.016*"justice" + 0.015*"congressrun2018" + 0.014*"trump" + 0.012*"
*"fulldisclosure"')
(4, '0.015*"qanon" + 0.010*"trump" + 0.009*"fusion" + 0.009*"maga" + 0.008*"post" + 0.007*"w
(5, '0.017*"wethepeople" + 0.014*"libertyrising" + 0.014*"noconcon" + 0.012*"trump" + 0.011*
0.007*"stone"')
(6, '0.026*"qanon" + 0.024*"post" + 0.009*"qalert" + 0.009*"4/11/18" + 0.008*"devin" + 0.007
(7, '0.008*"trump" + 0.008*"know" + 0.006*"president" + 0.006*"2016" + 0.005*"spectrum" + 0.
(8, '0.010*"like" + 0.009*"system" + 0.008*"know" + 0.007*"profit" + 0.007*"vote" + 0.007*"m
(9, '0.019*"murderer" + 0.014*"government" + 0.011*"county" + 0.010*"broward" + 0.010*"menta
*"israel"')
(10, '0.014*"politico" + 0.008*"school" + 0.007*"time" + 0.006*"trump" + 0.006*"know" + 0.00
(11, '0.027*"criminal" + 0.024*"qanon" + 0.010*"tell" + 0.008*"know" + 0.006*"even" + 0.005*
(12, '0.000*"host" + 0.000*"know" + 0.000*"think" + 0.000*"server" + 0.000*"make" + 0.000*"f
(13, '0.009*"know" + 0.007*"twitter" + 0.007*"clinton" + 0.007*"shearer" + 0.007*"follow" +
```

## CSV twentyfive_thirty

In [87]:

```
THIS_FOLDER = os.getcwd()
threads_leer = Tthreads5
carpeta_guardar = "Ttpcsv5"

#Poblar text_data


camino = os.path.join(THIS_FOLDER, carpeta_guardar)
text_data = []
documentos = []
dictionary = []
corpus = []
documentos = threads_leer

#print(documentos)

for line in documentos:
    #print(line)
    tokens = prepare_text_for_lda(line)
    if random.random() > .009:
        #print(tokens)
        text_data.append(tokens)

#print(text_data)
NDIC = camino+"\\t_dictionary1.gensim"
NMOD = camino+"\\t_model1.gensim"
NCOR = camino+"\\t_corpus1.pkl"
dictionary = corpora.Dictionary(text_data)
corpus = [dictionary.doc2bow(text) for text in text_data]
pickle.dump(corpus, open(NCOR, 'wb'))
dictionary.save(NDIC)

ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics = NUM_TOPICS, id2word=dictiona
ldamodel.save(NMOD)
topics = ldamodel.print_topics(num_words=NUM_WORDS)
for topic in topics:
    print(topic)
```

```
(0, '0.020*"system" + 0.010*"head" + 0.009*"canal" + 0.009*"balance" + 0.007*"vestibular" +
(1, '0.018*"page" + 0.007*"carter" + 0.007*"stigmergy" + 0.006*"2016" + 0.006*"c\x92est" + 0
(2, '0.026*"trump" + 0.014*"bombshell" + 0.010*"page" + 0.010*"qanon" + 0.009*"mueller" + 0.
(3, '0.013*"trump" + 0.006*"flynn" + 0.006*"obama" + 0.005*"russia" + 0.005*"know" + 0.004*"
(4, '0.022*"path" + 0.021*"openingday" + 0.021*"thingskidshavetaughtme" + 0.020*"qanon" + 0.
+ 0.007*"ecodescom" + 0.006*"woman"')
(5, '0.035*"contd" + 0.021*"mulehead" + 0.010*"flynn" + 0.006*"trump" + 0.006*"view" + 0.006
(6, '0.010*"iraq" + 0.009*"iran" + 0.008*"antum" + 0.008*"yang" + 0.007*"election" + 0.007*"
(7, '0.013*"trump" + 0.008*"facebook" + 0.007*"people" + 0.007*"time" + 0.006*"say" + 0.006*
(8, '0.016*"people" + 0.010*"white" + 0.008*"world" + 0.007*"chinese" + 0.006*"like" + 0.006
(9, '0.012*"basilisk" + 0.008*"people" + 0.008*"friendly" + 0.007*"also" + 0.007*"assume" +
(10, '0.037*"qanon" + 0.012*"internetbillofrights" + 0.011*"post" + 0.009*"stormy" + 0.008*"
ht"')
(11, '0.011*"york" + 0.010*"general" + 0.008*"washington" + 0.006*"officer" + 0.006*"terrori
b"')
(12, '0.046*"find" + 0.022*"vote" + 0.018*"poll" + 0.018*"info" + 0.008*"place" + 0.006*"dro
(13, '0.011*"atheist" + 0.010*"content" + 0.007*"belief" + 0.006*"people" + 0.006*"call" + 0
(14, '0.018*"food" + 0.011*"people" + 0.011*"know" + 0.010*"empire" + 0.010*"iran" + 0.005*"
```

# Megacorpus

Como tercera alternativa de análisis, se decide unir todos los threads que se tienen en un megacorpus, por lo que presentes en los aproximadamente 500 documentos entregados.

In [88]:

```
megatexto = Tthreads1+Tthreads2+Tthreads3+Tthreads4+Tthreads5
```

In [89]:

```
megatexto
```

```
ttps://t.co/wT76hT941U) \n 7) Congress appropriates, up to @POTUS to spend it, or not, as he
dBQrCo', (https://t.co/YoredBQrCo',)
 'Part 1: March 9ish. An excited #Qanon breaks down two major fronts: #Snowden, and #NorthKo
Part 2: #Qanon follows up on Trump\'s pit stop in Hawaii after his most recent Asia tour. Q
 3: #Qanon continues to taunt #Snowden...says can get him at any time, and seems to suggest
 4: #Qanon\'s huge #BOOM: North Korea folded to Trump. CIA strings controlling NK behind the
5: #Qanon\'s 2nd wave of posts March 9, continued thread below.\r\r\n\r\r\n#Q brings up Obam
j) \n Part 6: #Qanon continues on the #Snowden theme. What\'s the angle here? Q implies the
R) \n Part 7: #Qanon links Twitter CEO Jack Dorsey, #Snowden, and figures like Lynn de #Roth
E) \n Part 8: the 3rd phase of #Qanon posts March 9...\r\r\n\r\r\nFirst follows up on pendin
PIm25PZFla) \n Part 9: #Qanon returns to topic of #Snowden. Operation Snowden.\r\r\n\r\r\n#Q
s://t.co/YIs3I4Khu6) \n Just FYI, this was the tweet a lot of people on 8ch believe #Qanon w
ps://t.co/6YEOU8qEfN',)

 'REALITY CHECK: Pruitt has spent $3 MILLION on security and travel in just over a year. His
REALITY CHECK: Here\'s where you could stay in DC for the rent Pruitt was paying his fossil
 REALITY CHECK: While Pruitt is a dangerous threat to the health of our families, his sloppy
 REALITY CHECK: Trump is defending a guy who regularly defies him....#BootPruitt https://t.c
tps://t.co/JGA39bmuuj (https://t.co/JGA39bmuuj) \n REALITY CHECK: Trump is also defending th
```

In [90]:

```
from gensim import corpora
import gensim
NUM_TOPICS = 20
NUM_WORDS = 10
import pickle
```

In [96]:

```
THIS_FOLDER = os.getcwd()
threads_leer = megatexto
carpeta_guardar = "mega"

#Poblar text_data

camino = os.path.join(THIS_FOLDER, carpeta_guardar)
text_data = []
documentos = []
dictionary = []
corpus = []
documentos = threads_leer

#print(documentos)

for line in documentos:
    #print(line)
    tokens = prepare_text_for_lda(line)
    if random.random() > .009:
        #print(tokens)
        text_data.append(tokens)

print(text_data)
NDIC = camino+"\\t_dictionary1.gensim"
NMOD = camino+"\\t_model1.gensim"
NCOR = camino+"\\t_corpus1.pkl"
dictionary = corpora.Dictionary(text_data)
corpus = [dictionary.doc2bow(text) for text in text_data]
pickle.dump(corpus, open(NCOR, 'wb'))
dictionary.save(NDIC)

ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics = NUM_TOPICS, id2word=dictiona
ldamodel.save(NMOD)
topics = ldamodel.print_topics(num_words=NUM_WORDS)
for topic in topics:
    print(topic)
```

```
[['extraordinary', 'evidence', 'treasury', 'committee', 'thompson', 'hmrc', 'customs', 'brex
bn', 'year', 'almost', 'jusâ', 'arrive', 'figure', 'export', 'consignment', 'average', 'cost
heap', 'business', 'almost', 'zero', 'cost', 'seek', 'replicatâ', 'thompson', 'say', 'expect
'2021', 'need', 'years', 'customs', 'partnership', 'require', 'thompson', 'say', 'bordâ', 't
â'], ['real', 'idea', 'immigration', 'system', 'year', 'bars', 'replace', 'create', 'path',
ource', 'fraud', 'detection', 'expand', 'guest', 'worker', 'program', 'allow', 'people', 'ba
eal', 'visa', 'take', 'survey', 'world', 'immigration', 'laws', 'sample', 'best', 'create',
ficial', 'close', 'tie', 'nativist', 'white', 'nationalist', 'fearmongering', 'group', 'fair
rrentâ', 'whine', 'illegal', 'immigration', 'expensive', 'whatabouting', 'things', 'spend',
'basically', 'inconsequeâ', 'panoply', 'ways', 'trump', 'use', 'cachet', 'billionaire', 'sup
'want', 'know', 'happen', '2016', 'every', 'american', 'critical', 'thinker', 'know', 'theâ'
nton', 'reach', 'working', 'class', 'votersâ', 'happen', 'cambridgeanalyticafiles', 'accumul
'trump', 'winning', 'upset', 'ignore', 'information', 'reveal', 'assiduous', 'work', 'genius
wmccabe', 'stormydaniels', 'scandal', 'well', 'latest', 'call', 'trump', 'attornâ'], ['attri
'omnibus"â', 'another', 'porkulus', 'bill', 'like', 'give', 'obama', 'horrid', 'years', 'omn
titution', 'president', 'mustâ', 'must', 'realize', 'â??appropriationsâ', 'omnibus', 'bill',
ver', 'congress', 'didnâ??t', 'thing', 'coulâ', 'decide', 'tell', 'treasury', 'dept', 'â??sl
'spend', 'plea', 'suspect', 'obama', 'never'], ['part', 'march', '9ish', 'excite', 'qanon',
```

## Analisis de resultados de megacorpus

Luego de detectar los topicos, se clasificarán los threads de un archivo según los tópicos obtenidos.

In [92]:

```
for hilo in Tthreads1:
    hilito = prepare_text_for_lda(hilo)
    hilito_bow = dictionary.doc2bow(hilito)
    print(hilo)
    print(ldamodel.get_document_topics(hilito_bow))
```

```
Extraordinary evidence at Treasury committee from Jon Thompson, CEO of HMRC on customs and B
 The Brexiter favourite Max Fac - would cost business between Â£17 and Â£20bn a year

- that's almost 1% of GDP

- jusâ?¦ https://t.co/0MwIcwre4t (https://t.co/0MwIcwre4t)
 How does he arrive at the figure

200m export consignments at an average cost of Â£32.50 each = Â£6.5bn (times two beâ?¦ https
 Theresa May's New Customs Partnership is much cheaper for business (almost zero cost)  beca
 Mr Thompson said he did not expect the EU to reciprocate over the customs partnership.

What that means is UK collâ?¦ https://t.co/9c3uhhnZGX (https://t.co/9c3uhhnZGX)
 Both would not be ready by 2021. Max Fac needs 3 years. Customs Partnership requires 5, Mr

The bordâ?¦ https://t.co/luLzgUsiR4 (https://t.co/luLzgUsiR4)
 "We think we can manage the risk - we think we can" he said. He didn't sound so sure.
```

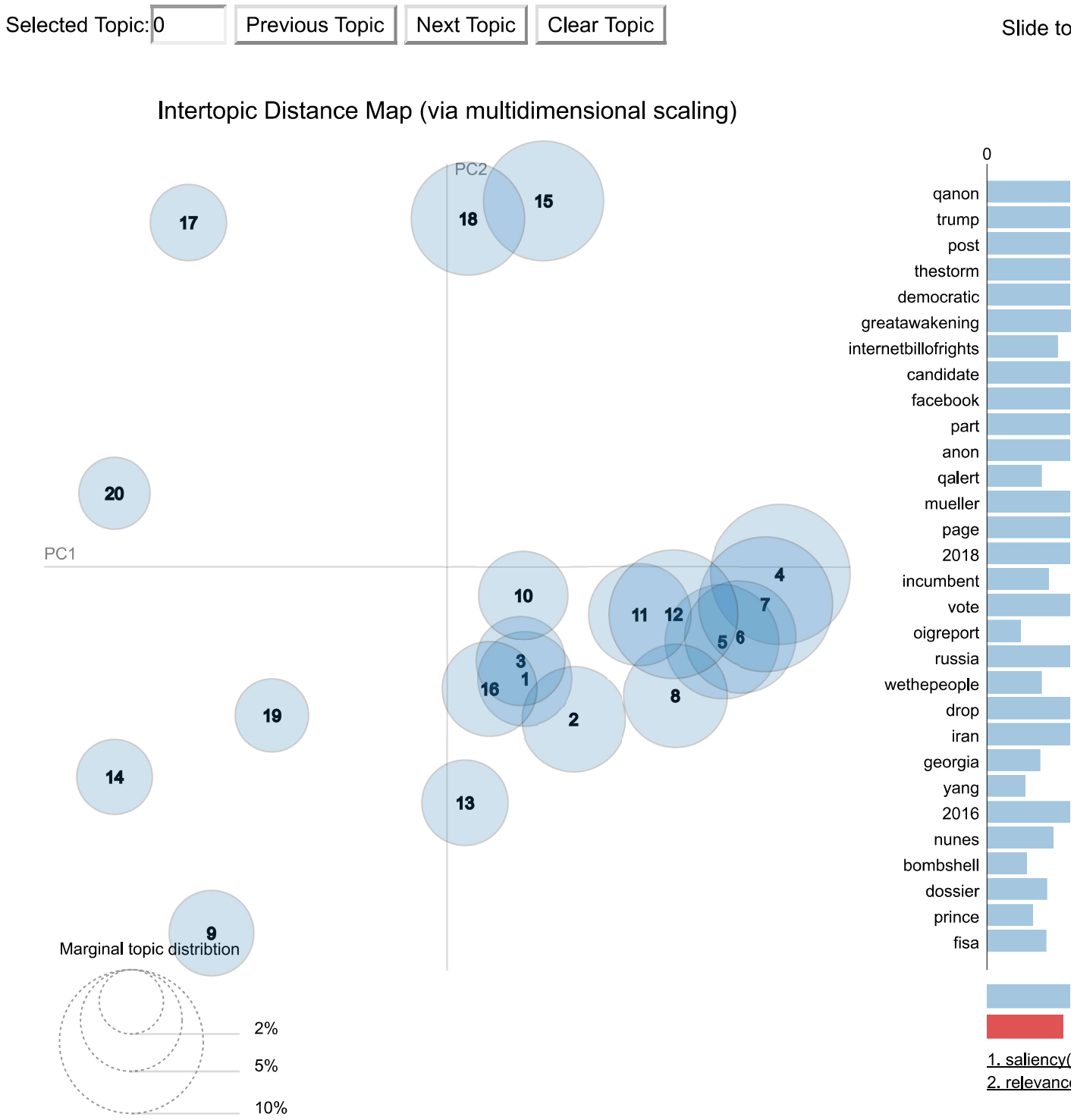
## Análisis de tópicos

Es posible analizar la relación entre los tópicos obtenidos a través de la librería pyLDAvis, la cual grafica la distanc

In [93]:

```
dictionary = gensim.corpora.Dictionary.load(NDIC)
corpus = pickle.load(open(NCOR, 'rb'))
lda = gensim.models.ldamodel.LdaModel.load(NMOD)
import pyLDAvis.gensim
lda_display = pyLDAvis.gensim.prepare(lda, corpus, dictionary, sort_topics=False)
pyLDAvis.display(lda_display)
```

Out[93]:



In [ ]:

```
```