Genona Torruella, Adrià Solé y Iago Nus

Report

Section 1 - Data collection

Link to data.json: https://drive.google.com/file/d/1o55-EmZG0GdrLIF2a7_u91_xVvw3vvFR/view?usp=sharing

- Here the information about
- the data collection, like the aggregated statistics of the collected data (number of tweets, number of users, etc..)

Hemos obtenido un total de:

```
Tot Retweets
125480
Unique Tweets
87497
Unique Users
158014
```

- the keywords used for the data collection

Las keywords que hemos utilizado han sido relacionadas con el COVID-19. Podemos verlas a continuación:

```
TRACKING_KEYWORDS = ["coronavirus", "covid", "#COVID", "#coronavirus", "#COVID19",
                     "disease","pandemic","vaccine","dead","deaths","#WHO","Hospitals",
                      "doctors","nurses","quarantine","symptoms","fauci"]
```

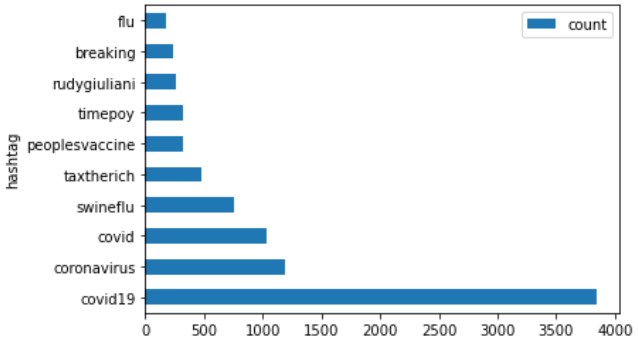
- the **approximate** time needed to collect the data.

Aproximadamente tardo 1 hora 11 minutos y 28 segundos.

Section 2 - Search Engine

- Without considering the GitHub repo:
  - Description of the pre-processing strategy

Primente hemos extraído los hashtags para ver que nuestras palabras claves estuviesen relacionadas con los tweets y las intenciones de nuestras keywords. Hemos obtenido que los hashtags más comunes están relacionados con el Covid como era de esperar. Añado una imagen:



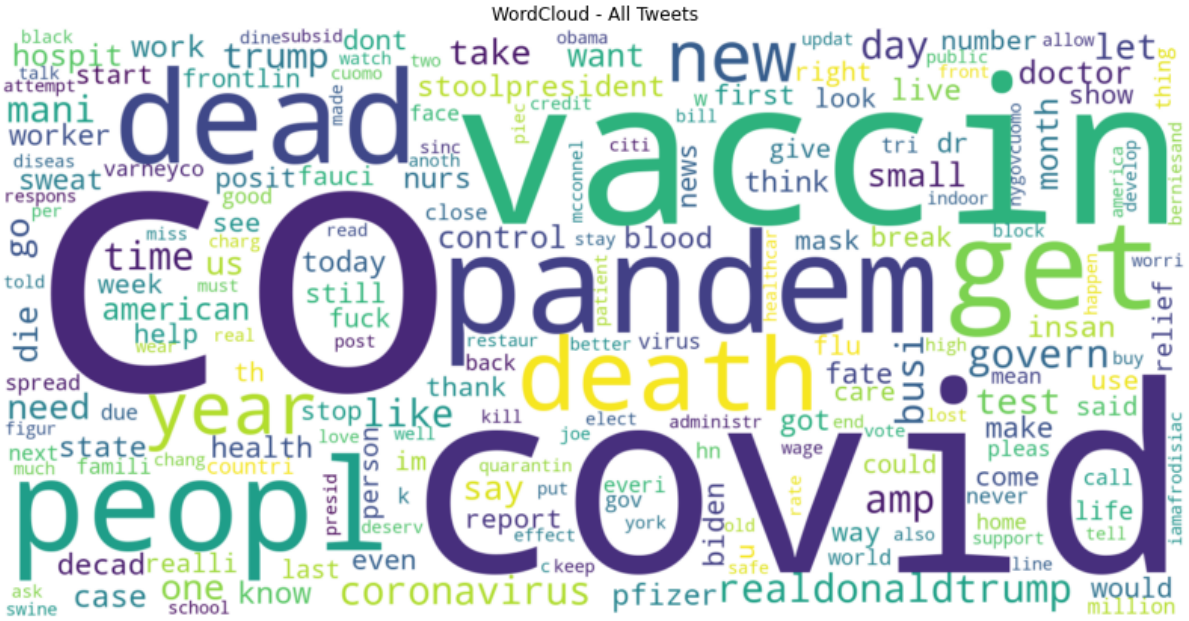
A continuación hemos limpiado nuestros tweets pasandolos a minúsculas, eliminando los caracteres especiales, extrayendo los emojis, eliminando más de un espacio entre las palabras…. Añadimos un ejemplo del cuerpo de un tweet de cómo lo leemos y como acaba después de limpiarlo:

```
array([['@HouseGOP @RepLizCheney This #UnitedStates Representative cares nothing for Americans Struggling with #covid19. Hea… h
ttps://t.co/l2u07hVh4H',
        'housegop replizcheney this unitedstates representative cares nothing for americans struggling with covid hea https t c
o luhvhh']],
      dtype=object)
```

El siguiente paso ha sido tokenizar el texto y de los tokens que obtenemos los filtramos por si no son stopwords y por sus stemmings. También añadimos dentro de los stopwords las palabras de rt y de https ya que no tienen significado y eras de las palabras más fracuentes.

Después de realizar este proceso podemos decir que filtrado el texto.

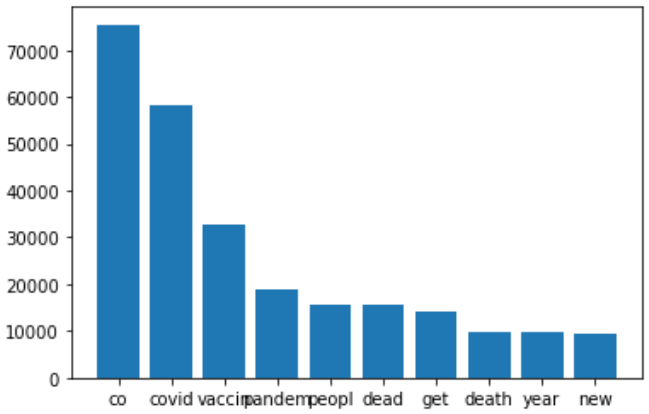
- WordCloud generated by the whole corpus (or at least a relevant sample).



- Only the words considered for the inverted index have to be included.

Las palabras que hemos utilizado para el inverted index son aquellas que ya hemos filtrado como en el paso anterior. Dentro del inverted index no encontraremos stopwords y palabras que han pasado el “stemming”. Como vemos en el wordcloud, todas las palabras que tienen más frecuencia tienen un significado o son el lexema de la palabra.

- Bar plot of the 10 most frequent words (same set of the WordCloud)



Podemos ver que las palabras más frecuentes son co y covid que se refieren ambas al covid. En este caso co es una abreviatura. Sigue de vaccine, pandemy, people, dead… . Vemos que hay palabras que solo tienen el lexema porque las hemos filtrado con anterioridad.

- Description and example to describe your score to rank the documents

La función que hemos usado es la que usamos en un seminario que hicimos. Exactamente lo que hace la función es hacer un ranking de los resultados de la query basado en los pesos de tf-idf. La función está comentada en el notebook pero básicamente calcula un score para cada documento, en este caso tweet a partir del tf-idf de la consulta que hayamos puesto como input. El resultado es:

##
Query number: 1.
Query: covid
##

Sample of 5 results out of 56820 for the seached query:

tweet= covid 😭 - Username: pinchelindseyy - Date: Fri Dec 11 18:23:50 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0

tweet= Because of covid - Username: ThatsJDP - Date: Fri Dec 11 18:23:14 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0

tweet= no i do not have covid - Username: mangogurl - Date: Fri Dec 11 18:19:37 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0

tweet= what about covid????? - Username: wajidraja55 - Date: Fri Dec 11 18:18:48 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0

tweet= Covid 19 - Username: Antwone1600 - Date: Fri Dec 11 18:15:03 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0

Vemos que si la query es “covid”, nos devuelve los tweets que más se parezcan. Vemos que son tweets muy cortos donde la principal palabra es covid.

- **Screenshot** to compare the difference between the classical ranking and the one by your score (screenshot of the output generated by the search-engine)

Classical Ranking

```
Query: covid
########################################

Sample of 20 results out of 32923 for the seached query:

tweet= covid-1907.. - Username: biroleginn - Date: Sun Dec 13 11:08:40 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0

tweet= Covid - Username: _NicoleeeM - Date: Sun Dec 13 11:07:51 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0

tweet= Do I have covid?? - Username: jenna_1249 - Date: Sun Dec 13 11:01:03 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets:
0

tweet= But covid... :( - Username: nasucchan - Date: Sun Dec 13 10:59:05 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0

tweet= Covid-1907 - Username: sergennyolcu - Date: Sun Dec 13 10:56:51 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0
```

Similarity ranking

```
Query: covid
########################################

Sample of 20 results out of 10 for the seached query:

tweet= covid - Username: alijahh3 - Date: Fri Dec 11 17:48:20 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0 - Similarit
y: [1.]

tweet= 😔 Covid - Username: Blonde_tweeting - Date: Fri Dec 11 17:17:07 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0 -
Similarity: [1.]

tweet= Covid * - Username: Mooseloversuza1 - Date: Fri Dec 11 18:09:51 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0 -
Similarity: [1.]

tweet= Covid - Username: ilteri46009471 - Date: Fri Dec 11 18:09:50 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0 - Sim
ilarity: [1.]

tweet= COVID-19 - Username: nammmy_ - Date: Fri Dec 11 17:38:35 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0 - Similar
ity: [1.]
```

Section 3 - RQs

1st RQ - Output Analysis

- **How to choose the queries?**

Seleccionamos las queries buscando dentro del dicoFreq palabras que tengan una frecuencia alta y que estén relacionadas con el COVID-19. A continuación veremos algunos ejemplos (son los lexemas):

'covid': 58209,
'vaccin': 32746,
'pandem': 18704

- **Subsection RQ1**
  - List of 10 selected queries

```
queries = ['covid', 'vaccine', 'pandemic', 'people','death', 'coronavirus', 'test', 'doctor', 'quarantine', 'mask']
```

  - Answer to **Question d**

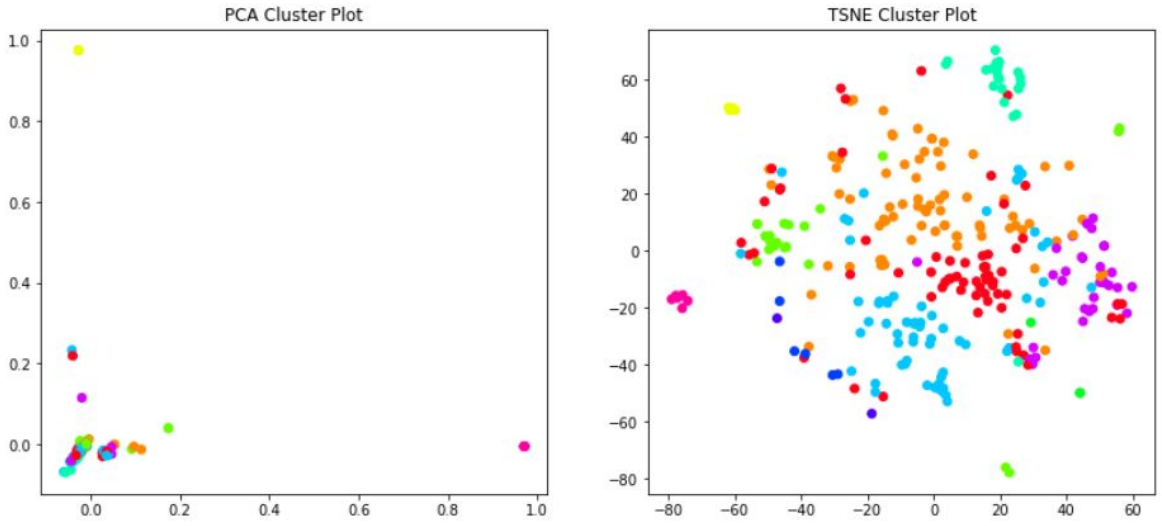
d) Can you imagine a better representation than word2vec? Justify your answer.
(HINT - what about Doc2vec? Sentence2vec? Which are the pros and cons?)

Una mejor representación que el word2vec sería aprovechar la información de cada palabra del tweet para generar un único vector que lo represente: la media de las palabras que forman el tweet. De esta forma tenemos algo que se asimila al Doc2Vec.

Un índice por documentos es más sencillo de mantener, la varianza entre los documentos siempre será menor a la que se da entre palabras, y por lo tanto es más fácil de mantener (updates). Creemos que un índice que contenga cada tweet será más eficiente para hacer la query, ya que contiene información de contexto.

  - Output given by t-sne

Hemos seleccionado 12 clústers. Vemos como se pueden distinguir diferentes tipos de clústeres por color. Lo más claros podrían ser el verde, naranja, azul cielo o rosa.



  - Answer to question RQ 1A - reasoning and plots may help.

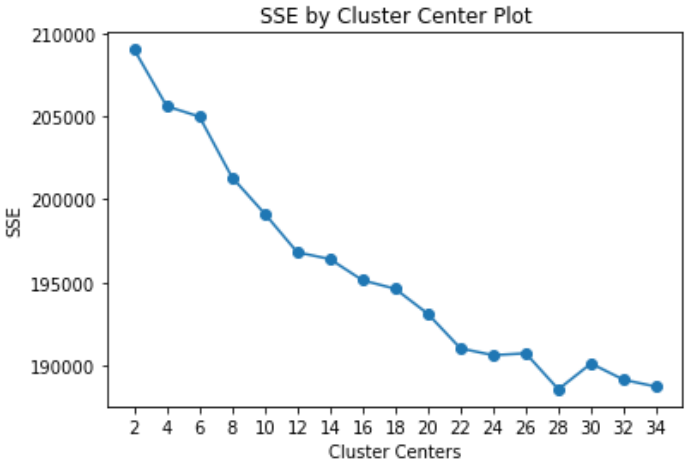
RQ 1A - Are you able to detect some subgroups within your tweets representation? Are you able to perform some clustering over the tweets and detect some topics within the conversation? How do you choose the best possible number of clusters?

Hemos dividido en 12 clusters ya que en el plot del SSE vemos que en 12 clusters es donde se produce el “codo”, donde el ritmo de bajada se estabiliza.

Los temas de los clusters se dividen en:

- Vacunas: obtenemos tweets con nombres de farmacéuticas, vacunas, compras de vacunas, etc.
- Gobierno USA: Sobre Biden, Trump, Obama y la administración del gobierno.
- Restricciones en ciudades: sobre restaurantes, ciudades, ‘indoor’ y Nueva York.
- Trabajadores en primera línea: tweets con palabras relacionadas con los oficios de sanidad, ‘frontline’, ‘nurses’, ‘doctors’, ‘workers’, etc. Sobretodo tweets haciendo hincapié en la importancia de su trabajo.
- Indignación respecto al coronavirus durante este año: Podemos verlo en tweets con menciones al tiempo, la virus y a insultos. Palabras como “F*ck”, “Year”, “Coronavirus”, “time” o “people”.

Como vemos en el gráfico, el efecto ‘elbow’ se produce en el cluster 12-14.

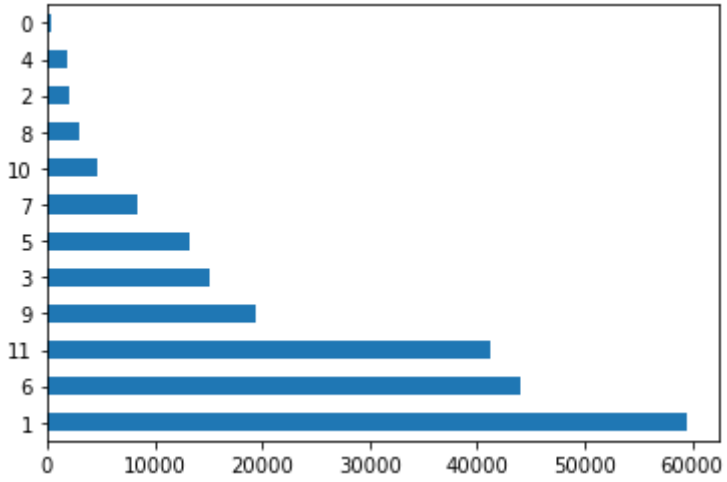


- Answer to question RQ 1B - a table for each cluster may help.

RQ 1B - What are the 5 most relevant keywords in the tweet that are part of each cluster? To what extent these keywords characterize/separate well the clusters?

CLUSTER	KEYWORDS	CLUSTER	KEYWORDS
0	friday,story,high,price,laid	**6**	tested,died,relief,positive,covid
1	time,fuck,like,year,coronavirus	**7**	new,covid,cases,day,deaths
2	forgot,workers,doctors,nurses,frontline	**8**	restaurants,city,new,york,indoor,dining
3	thank,workers,government,wages,subsidized	**9**	didn,buy,covid,pfizer,vaccine
4	joe,administration,biden,charge,obama	**10**	small,businesses,control,sweat,fate
5	realdonaldtrump,game,ass,alive,dead	**11**	time,just,people,coronavirus,pandemic

Vemos el número de términos que tenemos por cada clúster. El más numeroso es el clúster 1.



Subsection RQ2:

Queries escogidas:

```
queries = ['covid', 'vaccine', 'pandemic', 'people','death', 'coronavirus', 'test', 'doctor', 'quarantine', 'mask']
```

Using the same clusters defined in the previous question:

a. **Define the best possible number of clusters and assign the cluster labels to each document;**

   Hemos asignado 12 clústeres.

b. **Define a *diversity score* which the aim is to DIVERSIFY the final output. This score is assigned to the list of returned documents for the input query.**

   Con una input query, calculamos los top n documentos con mayor relevancia (usamos el ranking_ir, con word2vec). Una vez listados, calculamos el diversity score y seleccionamos un top n.

   Para calcular el diversity score hemos usado el lexical diversity, utilizado para calcular el ratio de palabras únicas diferentes. Es decir, un ratio mayor de palabras únicas significa mayor riqueza léxica.

```
# Lexical diversity
def lexical_diversity(text):
    return len(set(text)) / len(text)
```

c. **Now, defining a method to diversify the output through the diversity score defined above, try to generate a still relevant but more diverse final top-k list of documents.**

   Como hemos dicho en las respuesta anterior, primero calculamos los top n documentos con mayor relevancia y a continuación la diversity. Al no hacer un corte con los primeros n documentos por relevancia nos puede ocurrir que un documento no relevante tenga mucha diversity. De esta forma podemos hacer que la lista del top-k sea más diversa.

RQ 2A - Test your new method on some queries, comparing the 2 outputs before and after the re-ranking and comment the results.

Por la query ‘vaccine usa’ con ranking ir:

1. tweet= Getting the vaccine in a bit , I’m a lil nervous 😬😂 - Username: _YaGirlJessy - Date: Fri Dec 18 15:19:55 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0 - Similarity: [0.44812763]
2. tweet= RT @pastorlocke: All you vaccine sheep need to wake up. This nurse in TN took the vaccine, then passed out in the vaccine press conference.… - Username: rememberrooster - Date: Fri Dec 18 15:22:00 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0 - Similarity: [0.43151283]
3. tweet= RT @pastorlocke: All you vaccine sheep need to wake up. This nurse in TN took the vaccine, then passed out in the vaccine press conference.… - Username: ea53aebea017466 - Date: Fri Dec 18 15:21:08 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0 - Similarity: [0.43151283]
4. tweet= RT @NicStar310: No vaccine for AIDS in 40 years.

   No vaccine for common cold…

   But in one year, a vaccine for COVID19…

   FOH.... - Username: DeShawn__B - Date: Mon Dec 07 17:20:32 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0 - Similarity: [0.42984453]

Con ranking por similaridad y diversity:

1. tweet= Newborns in 2021 - Username: LivLafLuvDrank - Date: Mon Dec 07 16:43:56 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0 - Similarity: [0.47424382] - Diversity: [0.81818182]
2. tweet= Getting the vaccine in a bit , I’m a lil nervous 😬😂 - Username: _YaGirlJessy - Date: Fri Dec 18 15:19:55 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0 - Similarity: [0.44812763] - Diversity: [0.36956522]
3. tweet= RT @catturd2: We need an anti-dumbass vaccine in the USA. - Username: larryfix2 - Date: Fri Dec 18 15:20:02 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0 - Similarity: [0.42869449] - Diversity: [0.30188679]
4. tweet= RT @catturd2: We need an anti-dumbass vaccine in the USA. - Username: adorno918 - Date: Fri Dec 18 15:18:18 +0000 2020 - Hashtags: [] - Likes: 0 - Retweets: 0 - Similarity: [0.42869449] - Diversity: [0.30188679]

Vemos que los resultados se parecen mucho, ambos tops son iguales pero hay alguna diferencia en las posiciones más bajas. Los documentos con diversity almenos contiene las dos palabras de la query.

RQ 2B - What about the coverage? Any difference between the two rankings (with AND without diversity score)?

Podemos ver que el primer documento devuelto por el sistema con diversity quizás no es el más adecuado para la query, pero los siguientes sí parecen ser relevantes y además contienen ambas palabras de la query (esto no ocurre en el sistema que sólo computa la similaridad).

Por lo tanto creemos que es más eficiente el searcher que usa diversity, ya que parece que devuelve tweets más adecuados y relevantes.

Subsection RQ3:

- **Subsection RQ3**
  - Summary statistics of the retweet graph, train and test.

```
In [31]: 1 df_graph.describe()
```

Out[31]:

	RT	Post
count	37	37
unique	36	37
top	MollyJongFast	TimeskipRose
freq	2	1

```
In [28]: 1 test_sample.describe()
```

Out[28]:

	RT	Post
count	8	8
unique	8	8
top	eye_steal_memes	Alex_Turner_81
freq	1	1

```
In [29]: 1 train_sample.describe()
```

Out[29]:

	RT	Post
count	29	29
unique	29	29
top	KevinMKruse	TimeskipRose
freq	1	1

RQ 3A - Which is the best algorithm among the 4 selected in terms of accuracy?

- Como el algoritmo de ALS necesita un training creemos que será el más adecuado al saber cómo está repartido el grafo y sus puentes.
- El pagerank será mejor usarlo cuando queramos ver que nodos podríamos recomendar a otro nodo.
- Y Adamic-Adar se usa para buscar qué nodos están relacionados con los dos nodos estudiados en ese momento. Para ver las relaciones comunes entre ellos.