Análisis de redes Sociales – Práctica

Exploración de análisis de sentimiento en Twitter

Máster en Business Analytics y Big Data

Asignatura: Análisis de redes sociales
Módulo: Tecnologías de Big Data/Gestión de Datos
Alumno: Alberto Marino, albertomarino@campusciff.net
Profesor: David Martin-Corral

1. Introducción

a. Contextualización

Actualmente no se tiene certeza de la opinión madrileña de la gestión de Manuela Carmena a cargo del ayuntamiento de Madrid. los medios de comunicación son parciales, por lo que un buen indicador puede ser la información de la gente de a pie o también de los medios en una red social como twitter.

b. Objetivo y alcance

El objetivo de la práctica es sacar conclusiones de la opinión de la gente que se manifiesta vía twitter, identificando aquellos que más mencionan a #Carmena o #Ahora Madrid.

La idea es identificar qué medios o personas realizan más menciones, positivas o negativas, y medir de alguna manera su radio de influencia.

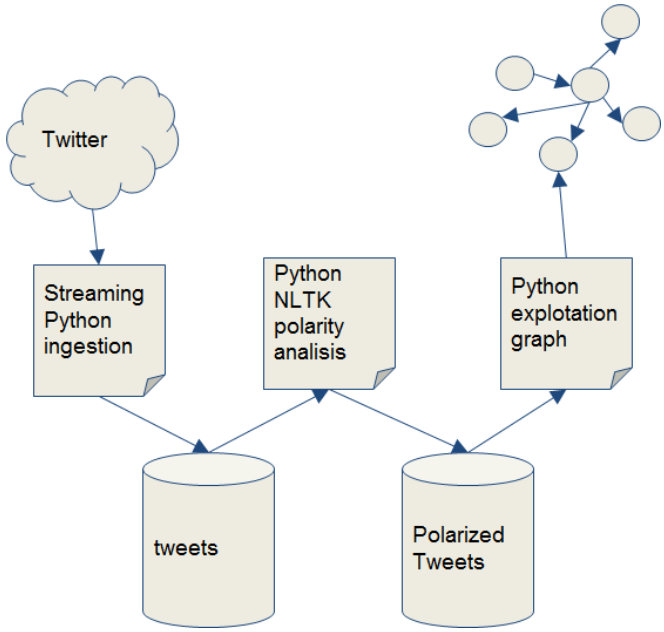
Identificar aquellas personas o medios y su tendencia así como influencia, es parte de las preguntas a contestar en esta práctica.

2. Práctica propuesta

a. Extracción de la información

La extracción proviene de twitter, mediante un conector que ha estado ingestando tweets con las menciones dadas en el contexto. Debido a que el conector tiene un timeout configurado y que solo pretende ser una muestra de ejemplo, el dataset proviene de un filtro de dos días en diferentes franjas horarias.

El conjunto de información se basa en más de 7000 tweets analizados y clasificados. El modelo del grafo se explica en el punto e.



b. Preparación de la información

Tras la ingesta y almacenamiento temporal en fichero, los datos han sido pasados por un algoritmo de clasificación NPL Naive Bayes, donde se introduce la polaridad de los tweets. Dicho algoritmo fue entrenado mediante el corpus del TASS y da una tasa de acierto bastante elevada.

El DataFrame generado contiene todos los datos necesarios para analizar la información.

```
dftweets.head(2)
```

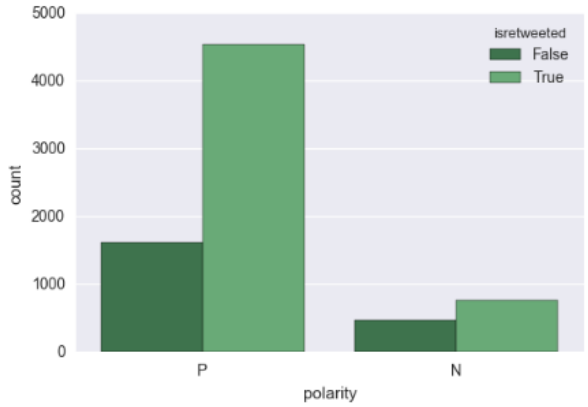
	userId	user_name	user_desc	user_url	user_followers	user_image
800355989086203905	7.990320e+17	GRUPO_METRO_MZA	GRUPO METRO MENDOZA.\nPRODUCTORA DE SEÑAL Y CO...	http://www.grupometromza.com	24.0	http://pbs.twimg
800355991057530880	2.326149e+09	pepelopezperaza	None	None	329.0	http://pbs.twimg

```
dftweets.head(2)
```

user_following	geo	coordinates	timestamp	...	isretweeted	autor_id	autor_name	autor_desc	autor_followers	autor_following	lang	lat
None	None	None	1479654712350	...	True	14436030.0	elmundoes	- Cuenta oficial del diario EL MUNDO - 📻 Sara...	2621501.0	None	es	nu
None	None	None	1479654712820	...	True	244101525.0	pmanglano	Ante todo, mucha libertad. Concejal del Ayunta...	16754.0	None	es	nu

Los primeros sondeos manifestaban una tendencia positiva, que se verá posteriormente reflejada en el grafo.

```
%matplotlib inline
import seaborn as sns, numpy as np

sns.countplot(x="polarity",hue="isretweeted", data=postdf, palette="Greens_d");
```



c. Generación del grafo en Neo4j

El primer intento fue volcar el grafo a Neo4j, lo que además sirvió para verificar que el API vía python es algo más fácil. Si bien, Neo4j es una NoSQL de grafos que tiene un shell (Cypher) de consulta muy potente, pero que el cálculo de las métricas de centralidad, betweenness, etc es más difícil de realizar que con Gephi.

Creacion de grafo dentro de Neo4j

```
def create_node_user(user,graph):
    try:
        user_node = Node("User", id=str(user.name), desc=user.desc, followers=user.followers, following=user.following)
        graph.merge(user_node)
    except Exception as e:
        print e.args,':','creating user node graph'

    return user_node

def create_relationship(tweet,graph):
    #publish relationship when is retwweted from
    #autor the original tweeter user
    #publisher is who publish the RT
    try:
        tweet_node = Node("Tweet", id=str(tweet.identifier), source=tweet.source, text=tweet.text, retweeted=str(tweet.retweeted)
                          polarity= tweet.polarity)
        graph.merge(tweet_node)
    except Exception as e:
        print e.args,':','creating tweet node graph'

    if (tweet.retweeted):
        pub = create_node_user(tweet.publisher,graph)
        graph.create(rel(tweet_node, "RT_BY", pub))
    else:
        au = create_node_user(tweet.autor,graph)
        graph.create(rel(tweet_node, "CREATED_BY", au))
```

d. Creación del grafo con Networkx de python

Una buena opción para calcular las distintas métricas de manera automática es Gephi, aunque la conexión con Neo no es sencilla, ya que el plugin necesario tiene dependencias y solo es compatible con las versiones 0.7.1. Aquí se utilizó la versión 0.9.1.

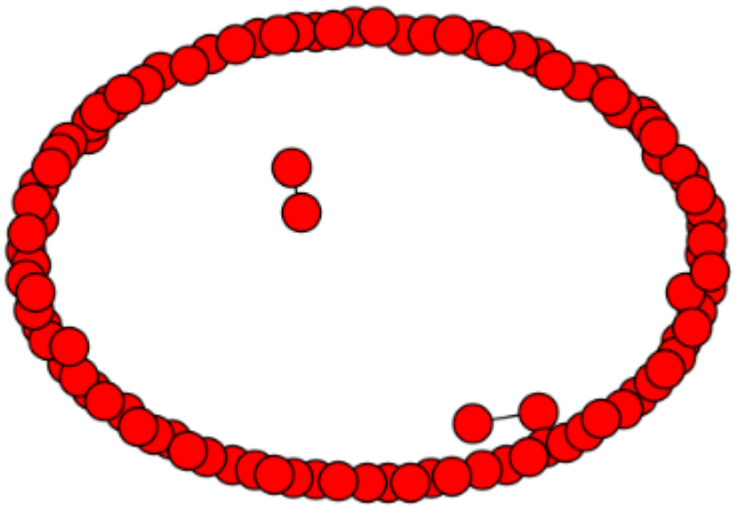
Por tanto, se generó de nuevo los métodos necesarios para crear el grafo vía Networkx, exportar a fichero .gml e importarlo en Gephi.

```
def create_node_user_x(user,g):
    try:
         g.add_node(enc(user.name), nodetype='user', followers=float(user.followers), following=float(user.following))
    except Exception as e:
        print 'ERROR:',e.args,':','create_node_user_x:creating user node graph'

    return user

def create_relationship_x(tweet,g):
    #publish relationship when is retwweted from
    #autor the original tweeter user
    #publisher is who publish the RT
    try:
        g.add_node(tweet.identifier,nodetype='tweet',source=enc(tweet.source),text=enc(tweet.text),retweeted=str(tweet.retweeted)
    except Exception as e:
        print 'ERROR:',e.args,':','create_relationship_x:creating tweet node graph'

    print 'INFO: teet retwweted:',tweet.retweeted
    if (tweet.retweeted == 'True' ):
        pub = create_node_user_x(tweet.publisher,g)
        g.add_edge(pub.name,tweet.identifier, edgetype='RT_BY')
        print 'Relation created:',pub.name,tweet.identifier,'RT_BY'
        #g.edge[pub.name][tweet.identifier]['type'] = 'RT_BY'
    else:
        #print 'Relation created:',au.name,tweet.identifier,'CREATED_BY'
        au = create_node_user_x(tweet.autor,g)
        g.add_edge(au.name,tweet.identifier,edgetype='CREATED_BY')
        print 'Relation created:',au.name,tweet.identifier,'CREATED_BY'
        #g.edge[au.name][tweet.identifier]['type'] = 'CREATED_BY'
```

Generacion de grafo desde DataFrame

```
g=nx.Graph()
dftweetslabeled = pd.read_csv('tweets_analyzed.csv', encoding='utf-8')
dftweetslabeled=dftweetslabeled.rename(columns = {'Unnamed: 0.1':'id'})
sub = dftweetslabeled[:50]
tw = 'null'
for index,row in sub.iterrows():
    pub = user_class(row['user_name'], row['user_desc'], row['user_followers'], row['user_following'])
    if (row['isretweeted']):
        autor = user_class(row['autor_name'], row['autor_desc'], row['autor_followers'], row['autor_following'])
    tw = tweet_class(row['id'], autor, row['tweet'], row['source'], row['isretweeted'], pub, row['polarity'])
    try:
        create_relationship_x(tw,g)
    except Exception as a:
        print 'ERROR:',a.args,':','creating node and relationship'

#nx.write_gexf(g, "tweeter_export.gexf")
nx.write_gml(g,"grafo_tweets_min.gml")
```

```
INFO: teet retwweted: True
Relation created: GRUPO_METRO_MZA 800355989086203905 RT_BY
INFO: teet retwweted: True
Relation created: pepelopezperaza 800355991057530880 RT_BY
INFO: teet retwweted: True
Relation created: Reiv4x 800355991892037632 RT_BY
INFO: teet retwweted: False
Relation created: subversivos_ 800355992147881988 CREATED_BY
INFO: teet retwweted: True
Relation created: angeljurado666 800355997126512640 RT_BY
INFO: teet retwweted: True
Relation created: PablodelaMac 800356004424622080 RT_BY
INFO: teet retwweted: True
Relation created: JuanmaSilvaLH 800356015325712385 RT_BY
INFO: teet retwweted: True
Relation created: RamonGismero 800356033025687552 RT_BY
INFO: teet retwweted: True
Relation created: niblick62 800356132388818945 RT_BY
INFO: teet retwweted: True
```



Como se aprecia, el formato por defecto de NetworkX no es muy usable. El modelado es distinto, ya que la tipología de nodos por ejemplo, debe ser un parámetro más del nodo (en Neo4j se modela directamente como nodos distintos)

e. Modelo del grafo

Para modelar el grafo, se ha utilizado parte de los campos de los campos de un dataframe utilizado para el análisis de sentimiento.

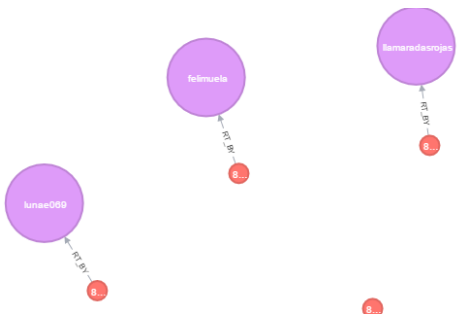
Del formato JSON de los tweets originales, se ha obtenido los datos pertenecientes tanto al autor como al publicador, generado con ellos Nodos de tipología “user”. Para estos nodos, se obtuvo también información de sus “followers”. Este dato permitiría verificar con el grafo, si las personas con más actividad son las más seguidas en la red, de cara a verificar su importancia en los medios.

Para los tweets, se maneja los datos de su identificador, su texto y su polaridad, creando así nodos de tipología “tweet”.

Las relaciones entre los tweets y usuarios, se califican de 2 tipos:

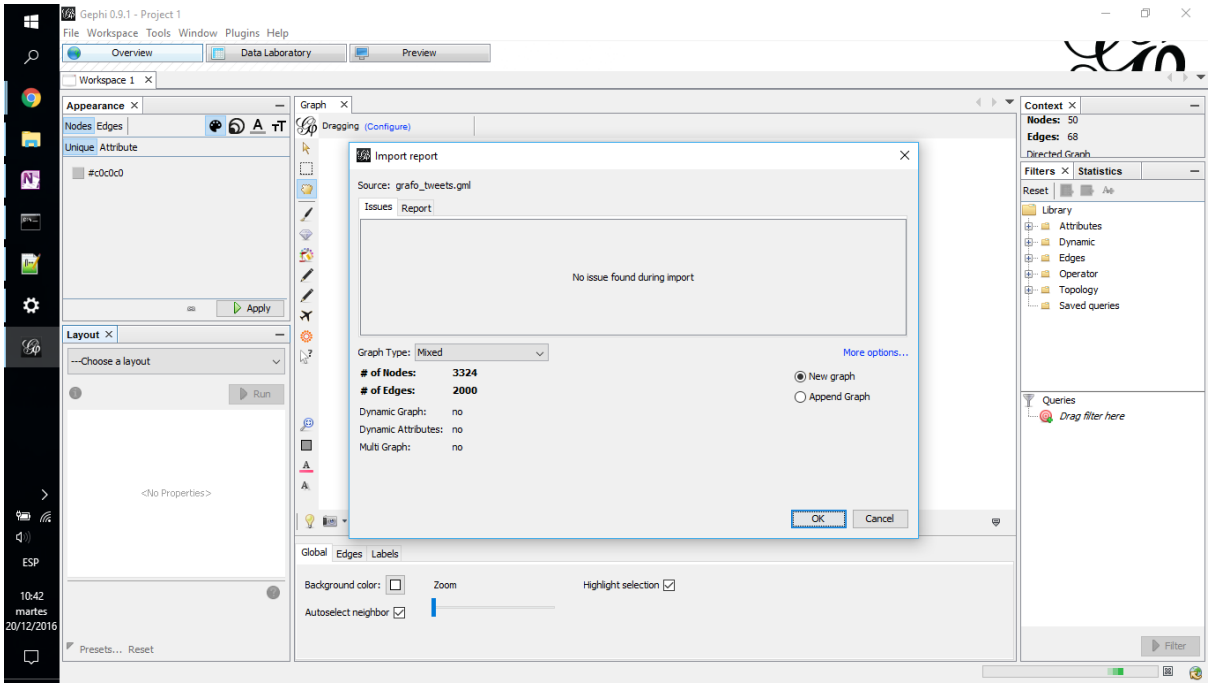
- RT (aristas rojas)
- CREATE (aristas azules)

De forma que se pueda identificar qué usuarios son aquellos que crean más contenido y más seguido en la red (aportan valor a la red) y aquellos que solamente retweetean información).



f. Importación en Gephi

Como se ve en la imagen adjunta, el grafo generado tiene más de 3300 nodos y 2000 aristas.

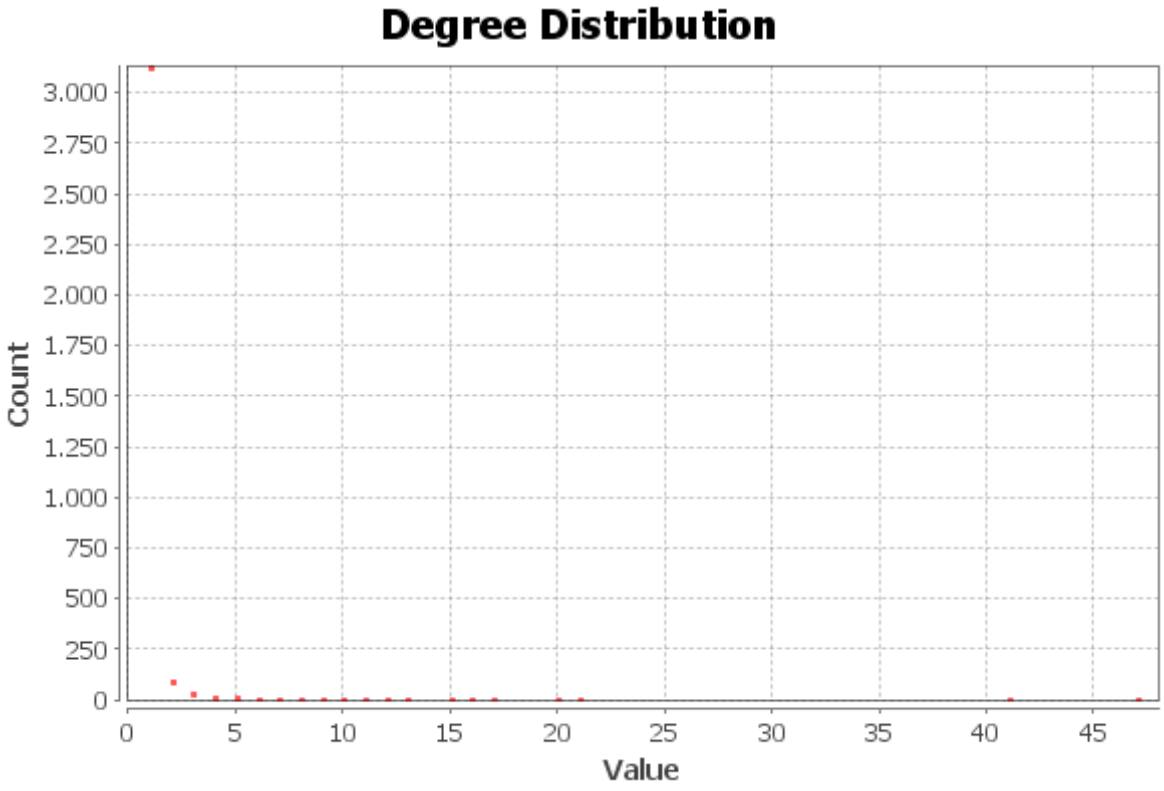


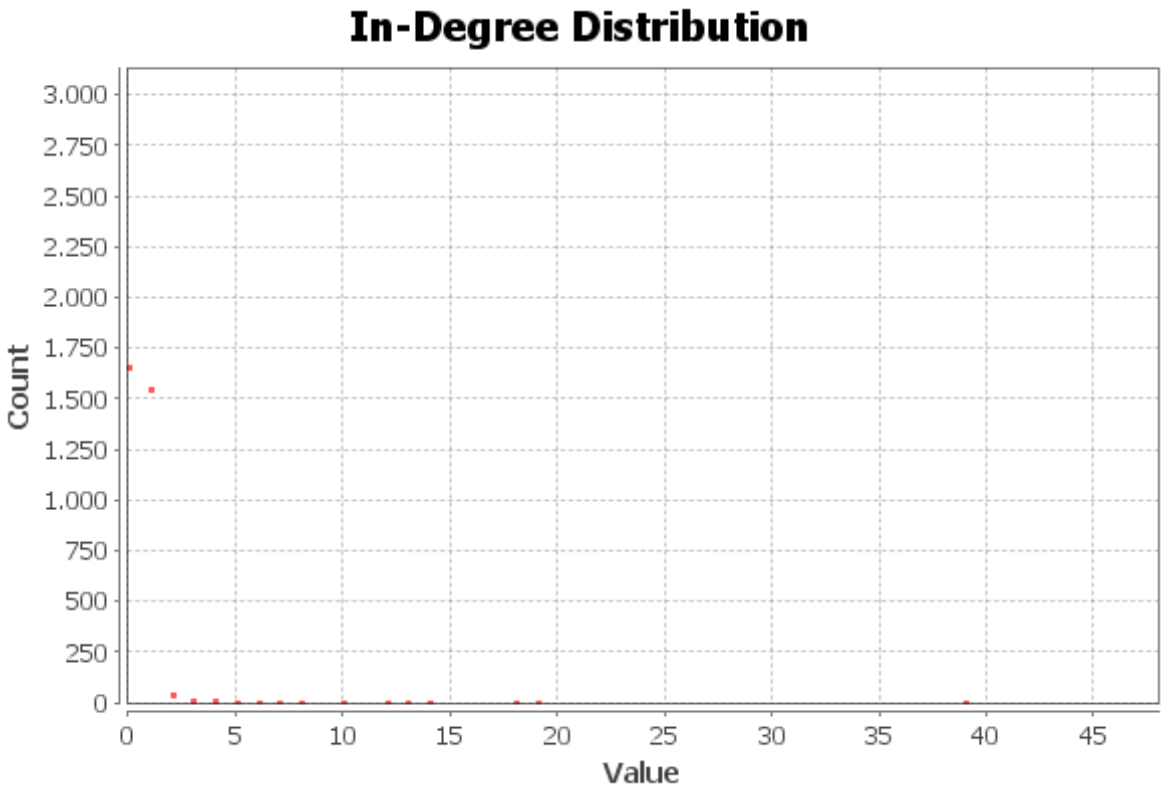
3. Análisis estadísticos

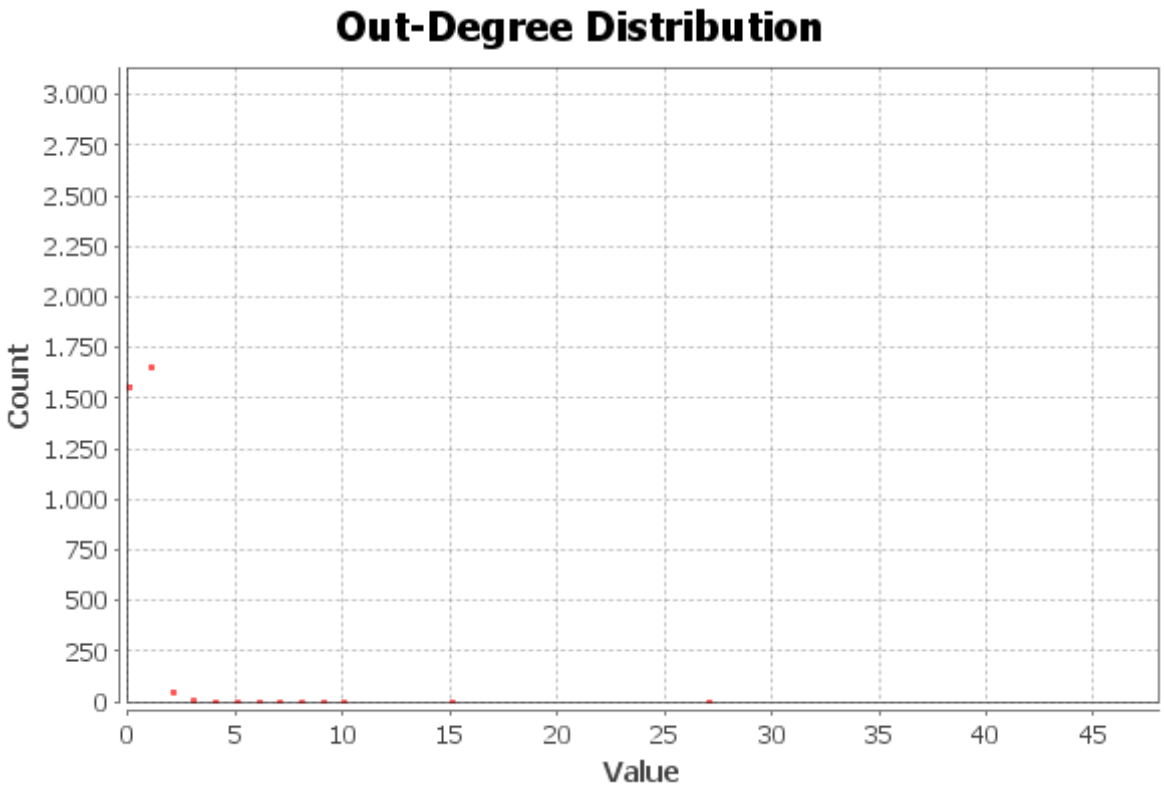
a. Cálculo de métricas

Graph Distance Report

El grado de un nodo es el número de aristas que entran y salen del nodo (la suma de in-bound y out-bound).
En las gráficas de distribución se muestra que prácticamente la mayoría de tweets tienen un grado de 1, de manera que son tweets creados (una sola vez) o retweeteados. El grafo se ha creado de tal forma que no se ha especificado la relación entre un tweet creado y retweeteado lo que seguramente nos daría más información de por donde pasa un tweet). La topología creada solo muestra la relación entre tweets y usuarios de cara a contestar las preguntas posteriores.
Los nodos con un grado mayor que 1 evidentemente son usuarios.

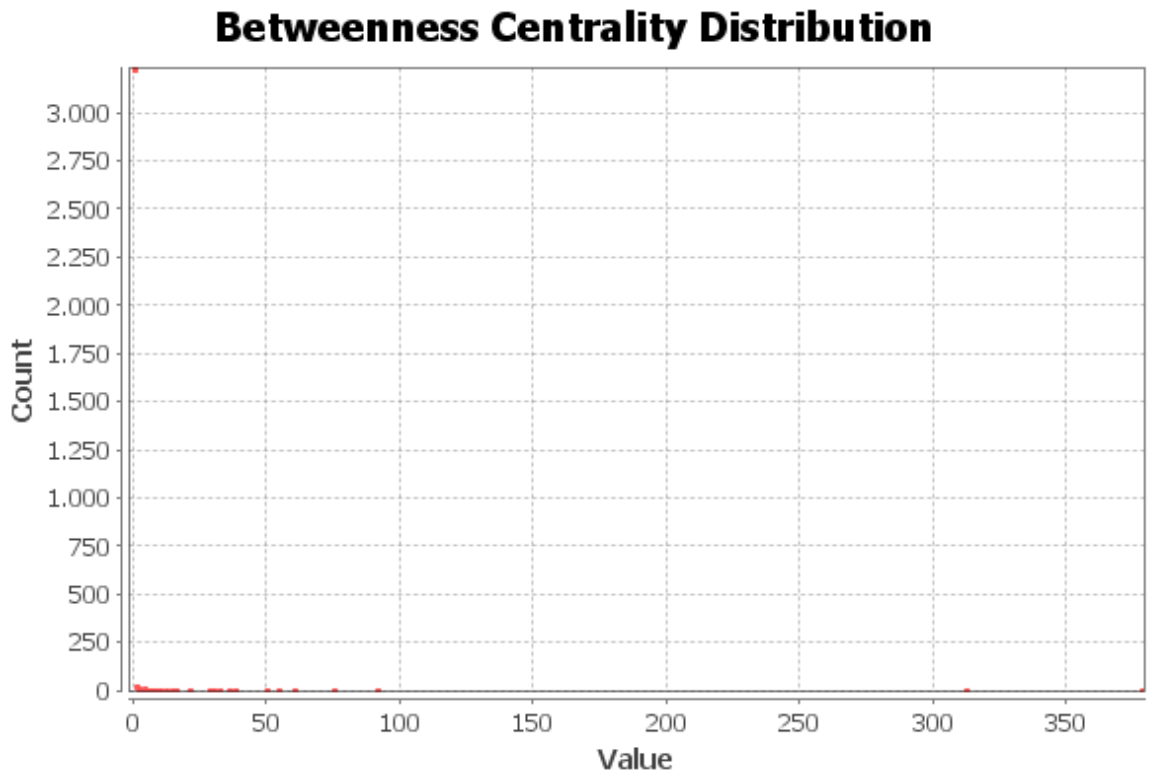

Degree Distribution
Count
3.000
2.750
2.500
2.250
2.000
1.750
1.500
1.250
1.000
750
500
250
0
0
5
10
15
20
25
30
35
40
45
Value


In-Degree Distribution
Count
3.000
2.750
2.500
2.250
2.000
1.750
1.500
1.250
1.000
750
500
250
0
0
5
10
15
20
25
30
35
40
45
Value

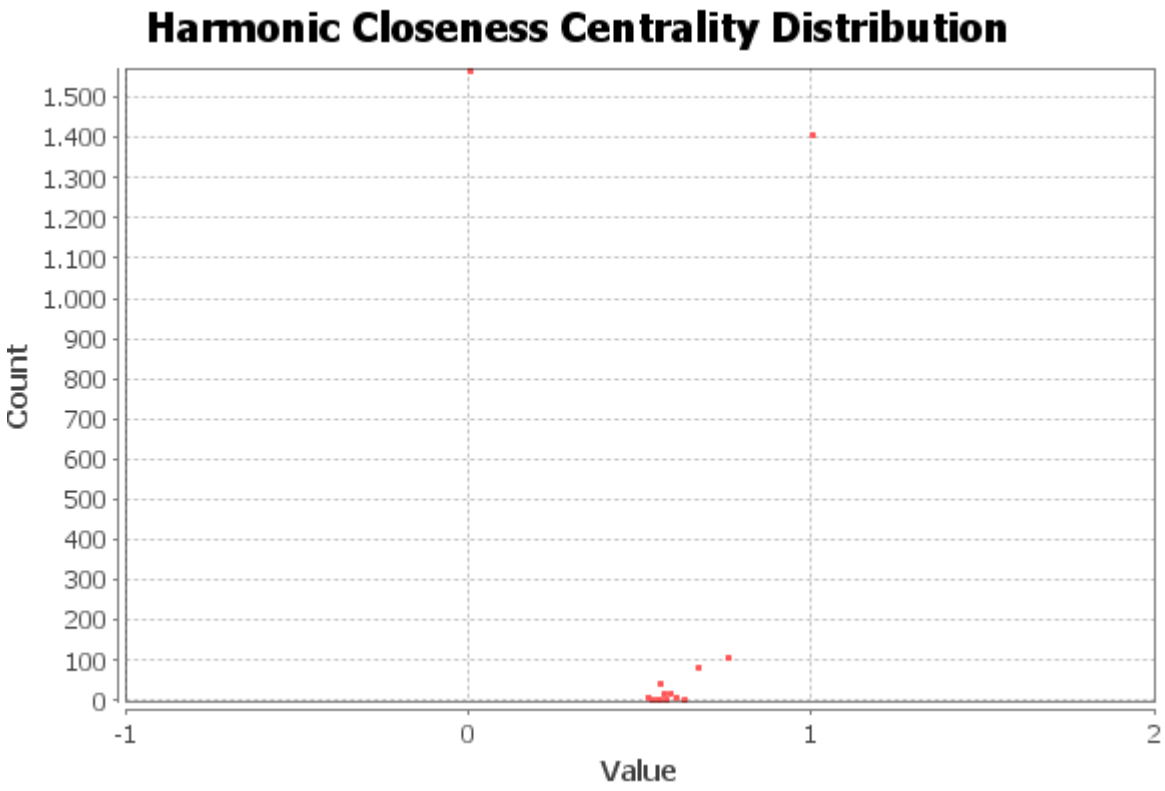


La gráfica de Betweenness mide las veces que un nodo está en el camino de otro. Como hay muchísimos tweets relacionados con un usuario, la mayoría delos nodos al ser tweets tienen un betweenness de 0.

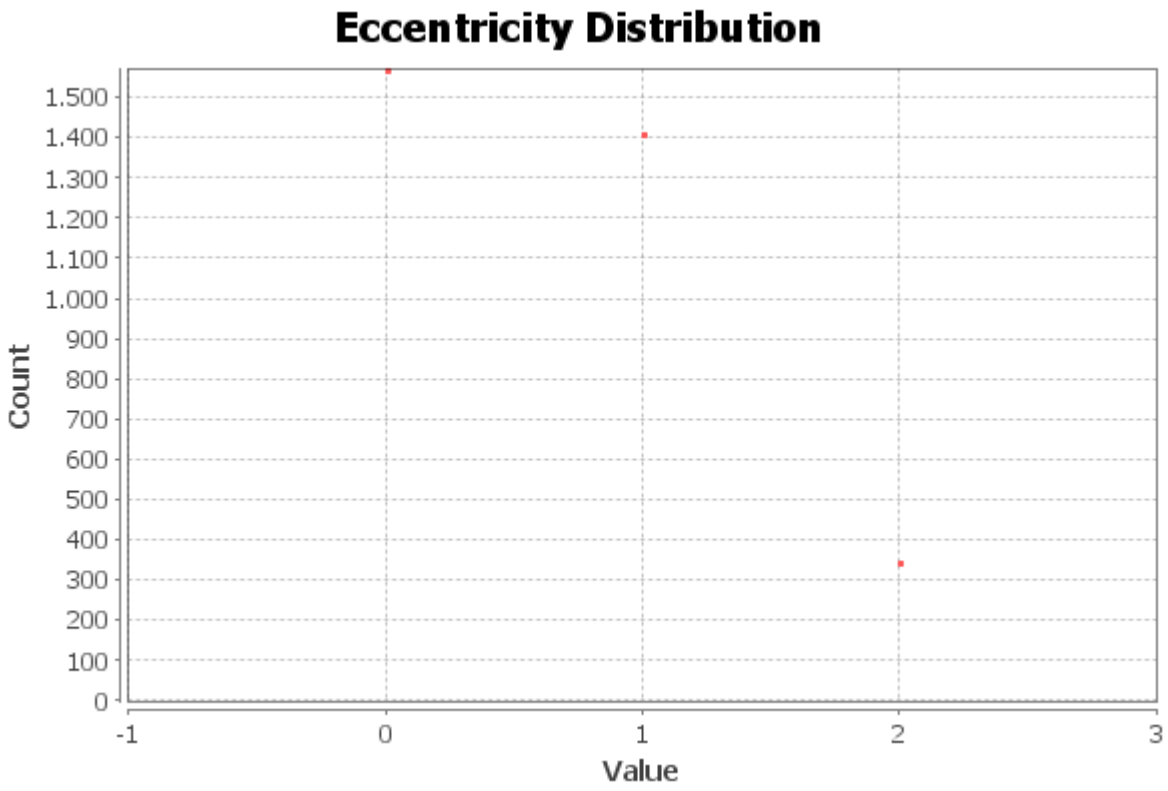
Los nodos “user” tienen un betweenness centrality mayor, ya que actúan de puente entre todos los tweets creados o retweeteados. Se ha marcado como dirigido. Si no lo llega a ser, probablemente el betweenness seria el doble.



El “closeness centrality” mide la distancia de un nodo a los demás de una forma aleatoria. Como la mayoría la distancia es 1, y muchos de los tweets y usuarios están a 1 salto, cuando los caminos son de un tweet a otro a través de un mismo usuario, equilibra la distribución al 0,5.

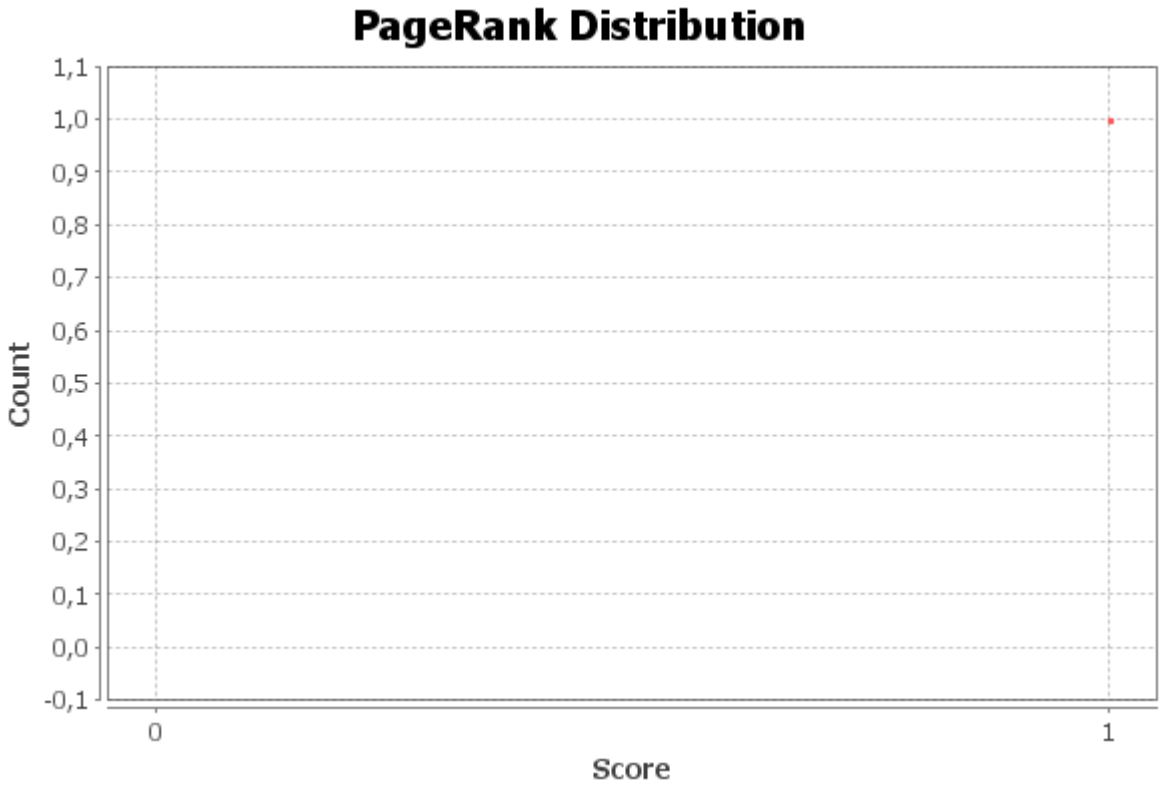


Eccentricity nos indica el máximo de las distancias de ese nodo y el resto. Los valores de 0 indica todas aquellas relaciones usuario-tweet aisladas.
2 indica la distancias de los tweets creados o retweeteados por un usuario.



A través de este algoritmo se realiza un scoring de cada uno de los nodos en función de su relevancia en la red. El algoritmo se basa principalmente en las aristas de un nodo y de la importancia de los nodos vecinos que relaciona. Valores altos de PageRank son considerados nodos importantes y estos mismos subyacentemente ayudan a nodos próximos a que lo sean.

En nuestro caso todos los nodos tienen un valor de PageRank de 1, lo que indica que no hay diferencia de peso entre la importancia de los nodos y sus cercanos.



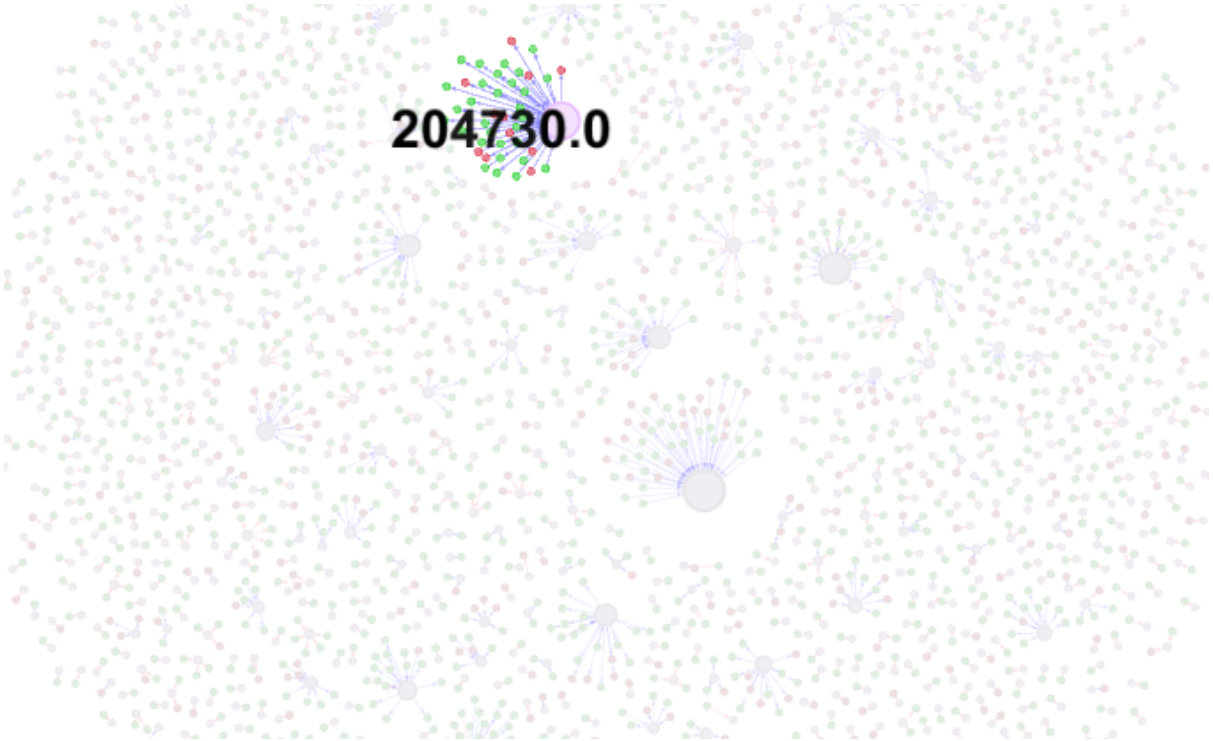
b. Resolución de cuestiones asociadas al grafo

Con los grafos obtenidos podríamos contestar alguna posible pregunta como los análisis que se realizan a continuación.

¿Hay relación entre los más seguidos y la cantidad de contenido que publican?

Pues vemos en la imagen adjunta como no es así en el grafo que hemos tenido. El tamaño de los nodos blancos (usuarios) es acorde al grado del nodo (entrada+salida). Podemos ver como Nodos con 30.610 followers, generan más información que algún nodo con mas de 2 millones de followers.


30610.0
2622341.0



¿Podríamos identificar tendencias políticas?

Claramente si, en función del tipo de tweets de algunos usuarios

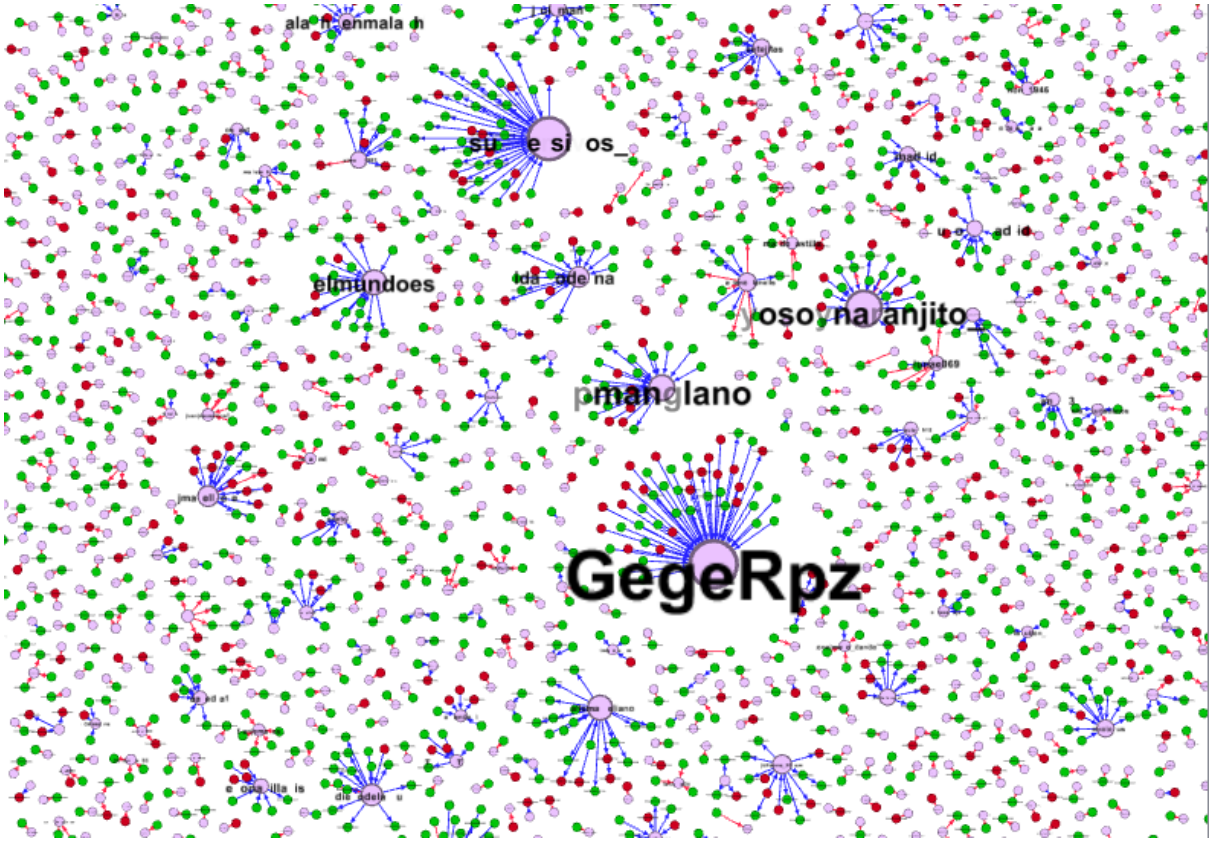
Pero hay alguna sorpresa al analizar tendencias políticas y es que, un medio como el mundo, parece que alaba en mayor medida la gestión de Carmena y ahora Madrid. Como se aprecia en el grado, las aristas rojas indican la creación de contenido relacionado con Ahora Madrid y Carmena y una proporción considerable de tweets positivos hacia su gestión. Esto quizás puede chocar con la ideología del propio medio o puede indicarnos que no es tan parcial.

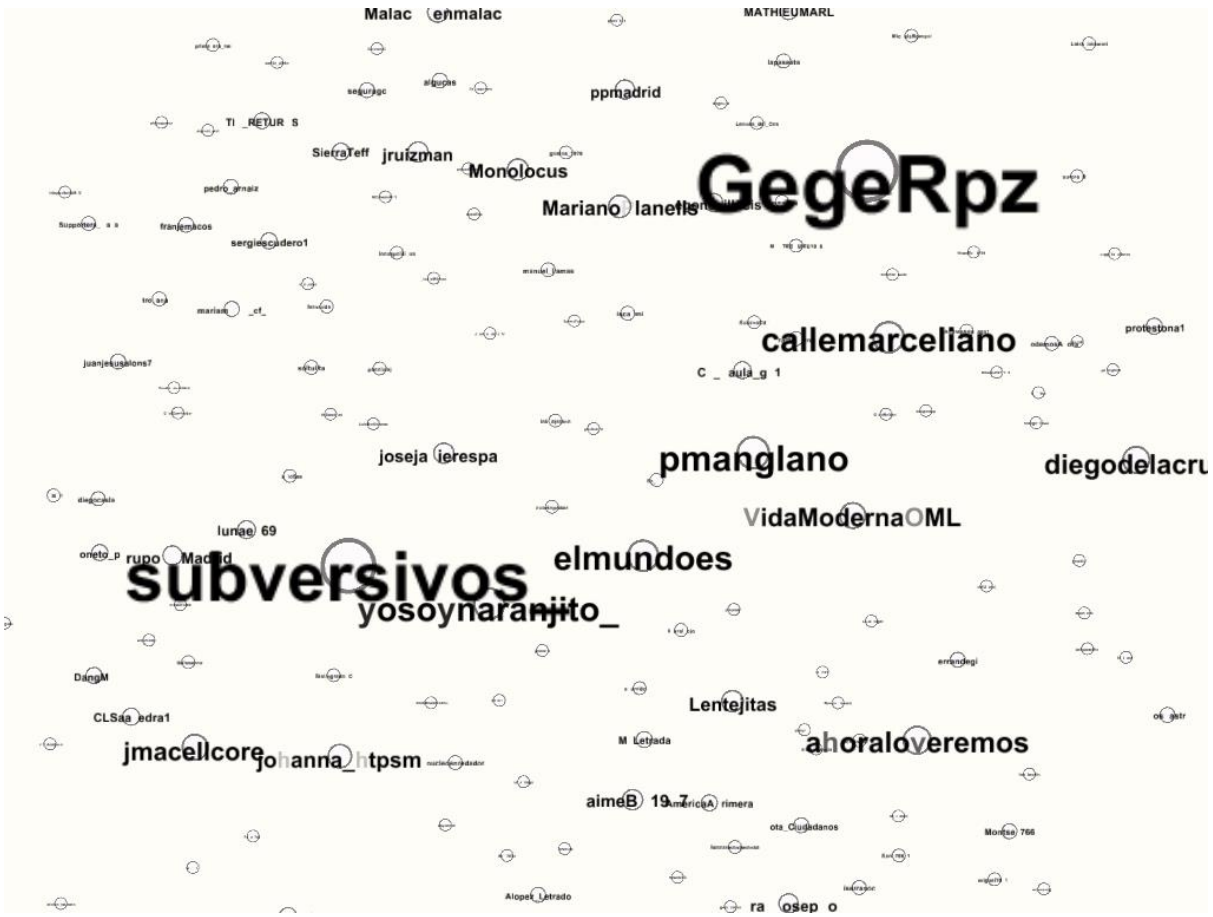
¿Podemos identificar medios o personas físicas hablando de estos temas?

Evidentemente teniendo el usuario, podemos identificar algunos medios, en este caso, el periódico el mundo con más de 2 millones de seguidores.

¿Podemos identificar quien genera más información y quien retweetea más en la red?

La respuesta también es clara, y al estar coloreados los tipos de relación, podemos ver que entidades generan más información sobre estos contenidos en la red.





NOTA: Por falta de tiempo no se puede replicar la red de retweeteo para ver caminos o tweets importantes. Pero es una posibilidad fácil de realizar teniendo los datos origen.