

Reto 3

Plantear un proyecto de análisis de redes sociales

CÉSAR GUTIÉRREZ PÉREZ

MÁSTER EN SOCIAL MEDIA (UOC) — ANALÍTICA AVANZADA

1. Situación inicial

① Este apartado forma parte de la *Entrega parcial*.

Worten es una tienda de electrodomésticos que desde 2021 se centra únicamente en la venta *online* (Romera, [2021](#)). Tiene un perfil de Instagram ([@wortenes](#)) relativamente peculiar en su sector. Su contenido consiste, principalmente, en vídeos de gatos y recetas que se han hecho populares en Instagram y TikTok. También suelen publicar memes, y, en general, contenido de terceros que genere *engagement* rápido.

A partir de esta situación, este trabajo plantea las siguientes hipótesis:

1. La empresa puede aumentar su *engagement* de forma notable si conoce más profundamente los intereses de su audiencia y los grupos internos que existen dentro de ella.
2. Conocer los usuarios mejor conectados dentro de su red de seguidores puede ayudar a la marca a orientar el contenido que publica hacia nichos más grandes de público.

2. Objetivos

① Este apartado forma parte de la *Entrega parcial*.

En base a las hipótesis formuladas, se marcan los siguientes objetivos:

1. Creación de un mapa que revele los grupos internos existentes dentro de la red de seguidores de la cuenta *@wortenes* en Instagram.
2. Desglose de grupos según su influencia dentro de la red y sus puntos en común (priorizando intereses compartidos, en caso de encontrarlos).
3. Identificación de los usuarios mejor conectados dentro de la red de seguidores de la cuenta *@wortenes* en Instagram.

3. Elección de herramientas

Para realizar este análisis usaremos la herramienta *instaloder*, debido a que es gratuita y podemos garantizar su seguridad gracias a que es de código abierto.

Procedimiento para extraer los datos

La herramienta nos permite crear una red de seguidores analizable a través de Gephi, aunque para ello tenemos que seguir los siguientes pasos:

1. Crear un entorno de Python e instalar Instaloder.

Para ello, debemos instalar la última versión de Python y ejecutar

```
python3 -m venv instaloder-venv
```

De esta forma, habremos creado un entorno apto para instalar programas a través de *pip*, el gestor de paquetes de Python. Para instalar *instaloder*, (en Linux) podemos ejecutar lo siguiente.

```
source bin/activate # Activa el entorno de Python  
pip install instaloder # Instala el paquete con pip
```

2. Generar un CSV de seguidores para analizarlo con Gephi.

Nuestra meta es obtener una red que incluya los seguidores de [@wortenes](#) y

realizar un análisis de los grupos distinguibles dentro de la red con Gephi. He creado el siguiente script, que nos permite generar un archivo .csv con la información de la red, y está anotado para resultar entendible a trabajadores no familiarizados con Python (explicado con detalle en el [anexo](#)):

```
import instaloader

# Inicia sesión en mi cuenta
L = instaloader.Instaloader()
L.login('██████████', '██████████')

# Define el perfil de worten como 'profile'
profile = instaloader.Profile.from_username(L.context, "wortenes")

# Consigue el nombre de cada uno de los seguidores de la cuenta
followers = profile.get_followers()

# Escribe el nombre de 'wortenes' seguido de ';'
print(profile.username, end=';')

# Escribe el nombre de cada usuario seguido de ';',
# excepto si es el último de la lista. Entonces,
# sigue el nombre únicamente con un enter.
for i, follower in enumerate(followers):
    if i < followers.count - 1:
        print(follower.username, end=';')
    else:
        print(follower.username)
```

He llamado al archivo de código *followers.py*, y quiero generar la tabla *worten.csv*. Puedo hacerlo situándome en la carpeta raíz del entorno de Python y ejecutando (en Linux) el siguiente comando:

```
./followers.py >worten.csv
```

3. Importarlo en Gephi

Cuando termine de ejecutarse el *script*, bastará con iniciar Gephi, dirigirnos a *Archivo* → *Importar hoja de cálculo* y seleccionar nuestro archivo *worten.csv*, que está ya preparado para ser leído sin problemas por Gephi.

Adecuación al problema planteado

El *script* cumple nuestro primer objetivo (la creación del mapa), y tiene todos los datos necesarios para poder distinguir los grupos que existen dentro de la red aplicando el algoritmo *ForceAtlas 2* y un análisis de modularidad.

El segundo objetivo requiere de cierta aportación personal. El *script* no recoge datos de las cuentas más allá de su nombre; sin embargo, cualquier analista debería poder identificar intereses comunes en los grupos mostrados por Gephi a través de su propio trabajo de investigación.

Nuestro tercer objetivo es imposible de conseguir sin crear una red que abarque un nivel más de profundidad. Esto es posible *en teoría*, y el código para generarla sería el siguiente:

```
import instaloader

# Inicia sesión en mi cuenta
L = instaloader.Instaloader()
L.login('██████████', '██████████')

# Define el perfil de Worten como 'profile'
profile = instaloader.Profile.from_username(L.context, "wortenes")

# Consigue el nombre de cada uno de los seguidores de la cuenta
for ifollower in profile.get_followers():
    # Imprime el nombre del usuario seguido de ';'
    print(ifollower.username, end=';')

    # Imprime el nombre de cada seguidor de dicho usuario,
    # seguido de ';' si no es el último, y seguido de un
    # enter si sí es el último.
    jfollowers = ifollower.get_followers()
    jfollowers_count = jfollowers.count

    for j, jfollower in enumerate(jfollowers):
        if j < jfollowers_count - 1:
            print(jfollower.username, end=';')
        else:
            print(jfollower.username)
```

No obstante, el tiempo de ejecución de este *script* sería inasumible para cualquier empresa, y lo más probable es que Instagram cortara su ejecución en cualquier momento por el volumen de solicitudes recibidas. Además, es probable que generara una red un tamaño inabarcable para cualquier procesador de redes.

Podemos conformarnos con una extracción del número de seguidores de cada seguidor de Worten y algo de investigación sobre los intereses de las cuentas que más seguimiento tengan.

4. Visualización de los datos

Un *dashboard* resulta muy atractivo como presentación de datos para cualquier empresa. Sin embargo, en este caso existen varios problemas a considerar:

1. No queremos mostrar diferentes gráficos, sino el análisis de una sola red. Esto hace nuestra visualización demasiado simple para un *dashboard*.
2. Un *dashboard* que muestre una red a un cliente incapaz de analizarla no es un producto atractivo ni beneficioso para dicho cliente.

La visualización única también resulta inservible para el cliente cuando lo que recibe es la representación de una red con un gran número de nodos y aristas, incluso a pesar de su estructura simple. Lo que genera valor a esta visualización es la explicación de cada grupo interno de la red y una explicación que aclare cómo aplicar los datos extraídos en cada caso.

Por estas razones, el trabajo debe ser presentado en un informe (o *report*), que muestre partes relevantes de la red acompañadas de un texto que explique (a) la razón de su relevancia, y (b) cómo puede la marca generar *engagement* con esa parte de la red.

Puedo citar como ejemplo para la estructura de dicho informe mi propia resolución del *Reto 2* de esta misma asignatura. En él, se descomponen los grupos de una red que muestra los seguidores de 5 *youtubers* autoubicados en la izquierda del espectro ideológico y 5 situados a la derecha del mismo.

La diferencia es que este informe debería presentar más capturas y centrarse más en la investigación de cada grupo analizado. También sería recomendable que el análisis se centre en explicar cómo puede acercarse la marca a cada módulo, proporcionando ejemplos de contenido compartido muchas veces en cada grupo interno de la red.

5. Anexo

Explicación del *script*

El código es simple. Se realiza una llamada al método [`Profile.get_followers\(\)`](#), definido en la documentación de `instaloader` (Instaloader, [2024](#)).

Tras realizar dicha llamada, se recoge el nombre de usuario ([`Profile.username`](#)) y se imprime con el método `print()` de Python a la salida especificada al ejecutar el programa. Por eso, en el comando de ejecución escribimos `>worten.csv`, para que use como salida nuestro archivo en lugar de la salida estándar (la consola).

Limitaciones del *script*

El código se puede ejecutar en cualquier ordenador personal, pero **es probable que tarde mucho tiempo en recibir los datos de los más de 100 mil seguidores de la cuenta de Worten** en Instagram. Además, si la conexión es demasiado larga, Instagram podría cerrarla aunque todavía no estén todos los datos listos, lo cual causaría un fallo interno en `instaloader`.

En cualquier caso, el *script* sí resulta válido para una empresa que trabaje con ordenadores potentes (sobre todo si puede conectarlos a través de SSL y distribuir el trabajo entre varias CPUs).

El informático Alfred Carpenter documentó cómo pudo conseguir, usando PhantomBuster en su ordenador personal, una red de seguidores de su perfil con más del doble de nodos que la que nosotros intentamos conseguir (Carpenter, [2022](#)). En cualquier caso, este *script* debería ser más rápido debido a que `instaloader` realiza las peticiones más rápido e implementa de forma automática los mecanismos de seguridad que este informático tuvo que escribir a mano.

5. Bibliografía

Carpenter, A. (13 de abril de 2022). Visualising my Instagram network with Python and Gephi. *Medium*. <https://medium.com/@alfredcarpenter/visualising-my-instagram-network-with-python-and-gephi-c7eb7346fd90>

Instaloader (18 de marzo de 2024). *Instagram Structures*. Instaloader. <https://instaloader.github.io/module/structures.html>

Romera, J. (13 de enero de 2021). MediaMarkt compra casi todo el negocio de Worten en España, que se queda solo con Canarias y una tienda en Madrid. *ElEconomista.es*. <https://www.eleconomista.es/empresas-finanzas/noticias/10990083/01/21/Media-Markt-compra-las-tiendas-de-Worten-en-Espana-que-se-queda-solo-en-Canarias.html>