Estrategias para la descarga masiva de seguidores

Resumen

Se buscan identificar las esferas de influencia de los usuarios con más de 10 mil seguidores. Debido a la enorme cantidad de seguidores acumulados, es imposible obtener esta información de manera directa mediante la API. En su lugar se propone obtener las cuentas que estos usuarios siguen, y mediante un algoritmo identificar a las primeras n cuentas más cercanas para posteriormente verificar su conexión.

0. Problemática y definiciones

La motivación principal detrás de la descarga masiva de seguidores es conocer las relaciones principales que existen entre los actores dentro del universo de datos descargado, principalmente de los usuarios más influyentes con el resto de la comunidad.

Uno de los factores más influyentes para comprender estas esferas de influencia son los seguidores y seguidos de cada usuario. Este tipo de relaciones puede expresarse con grafos dirigidos, de modo que un usuario puede seguir a otro sin que éste le siga de regreso. De hecho, se ha encontrado que el usuario promedio sigue a un determinado número máximo de cuentas, comúnmente con más seguidores que él o ella. El 10% de los usuarios más influyentes tiene más seguidores que cuentas seguidas, mientras que para el resto ocurre lo contrario (Wojcik & Huges, 2019). Esto puede verse claramente dentro de nuestra misma muestra. Esta información será de utilidad al momento de elegir la metodología a seguir.

Cantidad de seguidores promedio	Cantidad de seguidos promedio
136,282.97	1,817.75

El propósito de la descarga masiva es entonces consultar a la API de Twitter, no para descargar nuevos usuarios, sino para poder almacenar las relaciones entre estos. Dicha información se almacena en la tabla UserFollower, que consiste únicamente de dos columnas: userID (id de usuario) y followerID (id de seguidor). Se busca entonces descargar los seguidores de las cuentas influyentes.

La API ofrece tres endpoints para obtener esta información:

- GET followers/ids: Permite realizar una solicitud por minuto, y regresa páginas con 5k id's de seguidores de una cuenta.
- GET friends/ids: Permite realizar una solicitud por minuto, y regresa páginas con 5k id's de seguidos de una cuenta.

- GET friendships/show: Permite realizar 12 requests por minuto, recibe ids de dos usuarios, y regresa información sobre la relación entre estos.

1. Estadísticas actuales y metodología

Tras realizarse la descarga masiva, actualmente se tiene información en la base de datos de |U| = 511,544 usuarios. En Twitter existen alrededor de 300 millones de usuarios. Por lo tanto, esto equivale a $\frac{|U|}{|U_T|} = 0.0017$.

Como se mencionó anteriormente, de esta muestra de 511,544 usuarios, únicamente se descargan seguidores de aquellos más influyentes, definidos por tener al menos k seguidores. En este caso k = 10000 y en nuestra muestra, esto es igual a $|U_I| = 11623$.

Conviene resaltar que estos usuarios más influyentes acumulan 4,162,298,849 seguidores y 70,358,523 seguidos. Es importante mencionar que es necesario multiplicar ambas cantidades por $\frac{|U|}{|U_t|}$ para obtener los seguidores y seguidos que esperan encontrarse dentro de la base de datos.

De tal forma se cuenta con las siguientes estadísticas:

Estadística	Valor
Cantidad de usuarios descargados	511,544.00
Cantidad de usuarios en Twitter	300,000,000.00
% de usuarios en base de datos	0.20%
Mínimo de seguidores para cuenta influyente	10,000.00
Cantidad de cuentas influyentes	11,623.00
Acumulado de seguidores de cuentas influyentes	4,162,298,849.00
Acumulado de seguidos de cuentas influyentes	70,358,523.00
Acumulado de seguidores de cuentas influyentes en BD	7,055,908.05
Acumulado de seguidos de cuentas influyentes en BD	119,609.49

A partir de estas estadísticas puede evaluarse la conveniencia de cada uno de los endpoints disponibles en la base de datos.

- GET friends/ids. La cantidad de cuentas seguidas es mucho menor a la de seguidores. Puede ser de gran utilidad para observar las cuentas a las que siguen los usuarios influyentes.
- GET friendships/show. Aunque regresa una menor cantidad de datos, podemos controlar cómo se realizan las comparaciones, por lo que se reducen las consultas en vano. Puede ser de gran utilidad para verificar cuentas que estén muy relacionadas entre sí.
- GET followers/ids. Debido a la enorme cantidad de seguidores acumulados por las cuantes influyentes, y al bajo porcentaje de éstos que espera obtenerse en la base de datos, esta opción es la menos viable, a pesar de ser la más directa.

A partir de lo anterior, puede proponerse la siguiente metodología.

Identificar las cuentas seguidas por los usuarios influyentes, asumiendo que serán influyentes también para el resto de la comunidad. Posteriormente, identificar s pares de cuentas (una influyente, otra regular) con cercanía para comprobar su relación con la API de Twitter.

2. Encontrar relaciones entre cuentas influyentes

Como se mencionó anteriormente, las cuentas influyentes guían la conversación pública debido al amplio alcance que tienen. Por lo tanto, conocer las cuentas seguidas por las cuentas influyentes ayuda a comprender mejor su posición y motivaciones, a la vez que da una idea general de sus agrupamientos y esferas de influencia. Por lo tanto, se propone descargar estas conexiones.

Metodología

Se leerán las cuentas significativas de la base de datos. Usando las herramientas existentes, se realizarán llamadas a la API por cada una hasta agotar todas las cuentas seguidas. Esta información será almacenada en la tabla UserFollower.

Tiempo de ejecución

Como se mencionó anteriormente, el endpoint GET friends/ids permite 1 consulta por minuto, arrojando 5,000 id's en cada uno. Recordando que tenemos 5 clientes de Twitter, el tiempo de ejecución es de 25000 ids por minuto.

$$\frac{119{,}609.49 \text{ ids}}{25000 \text{ ids/min}} = \frac{2{,}814.34 \text{ min}}{60 \text{ min}} = 46.9 \text{ horas}$$

El tiempo de ejecución sería de alrededor de 48 horas.

3. Encontrar relaciones entre cuentas influyentes y no influyentes

Aunque el procedimiento anterior permite analizar los círculos en los que se desenvuelven las cuentas influyentes, no da mucha luz sobre la influencia que éstas tienen con las cuentas regulares. Usando la información existente en la base de datos es posible medir el valor de cercanía de dos cuentas, mediante un algoritmo de cercanía. Este algoritmo, que se describirá con más detalle más adelante, recibe dos id's de usuario y regresa la probabilidad de que haya algún tipo de relación entre los mismos.

Metodología

El algoritmo puede entonces ejecutarse para cada combinación de usuario influyente y no influyente para indicar su porcentaje de cercanía (p). Los resultados serán guardados en una tabla UserRelationAnalysis en orden descendente de p. La tabla será descrita con detalle más adelante.

Posteriormente, se leen los primeros s pares de cuentas que cumplan con cierto porcentaje p_0 y se llamará al endpoint friendships/show para realizar la comprobación. En la tabla UserRelationAnalysis también se registrará si un par ya fue comprobado, y si existió algún tipo de conexión entre los usuarios. Después de un determinado porcentaje de comprobaciones fallidas, se detiene la ejecución para poder ajustar los parámetros adecuadamente.

Tiempo de ejecución

Primeramente, será necesario ejecutar el algoritmo para cada combinación de usuarios influyentes y no influyentes. Por su naturaleza, este no depende de la API y la cantidad de instancias paralelas que pueden ejecutarse está limitada únicamente por el hardware del equipo. Realizado de manera eficiente, este procedimiento podría tardar tan solo unos días.

Por otra parte, el endpoint friendships/show permite realizar 12 consultas por minuto. Utilizando 5 clientes, esto se reduce a una consulta por segundo. El tiempo de ejecución depende entonces del número de seguidores estimados (s) que se desee verificar.

Para un valor s de $s = 3{,}527{,}954$ (la mitad de los seguidores esperados acumulados de cuentas influyentes en la base de datos), esto serían alrededor de 40 días.

4. Plan de ejecución

A partir de lo anterior, es posible identificar 3 procedimientos:

1. Consulta de cuentas seguidas por usuarios influyentes, con tiempo estimado de 2 días
2. Ejecución de algoritmo de probabilidad de cercanías, para cada cuenta influyente con el resto de las cuentas no influyentes, con posibilidad de ejecución en paralelo
3. Comprobación de pares de seguidos y seguidores en la API, con tiempo máximo estimado de 40 días

Al respecto, es importante mencionar que aunque el paso anterior es necesario para el siguiente, es posible realizar estos procedimientos de forma escalonada, con pequeños intervalos en que dos de ellos se ejecuten simultáneamente.

Igualmente, para la ejecución de cada procedimiento, se tomarán en cuenta todas las formas de incrementar la eficiencia.

A partir de lo anterior, el tiempo máximo esperado de la descarga masiva de seguidores es de un mes y medio.

5. Algoritmo de cercanía o similitud de usuarios y tabla UserRelationAnalysis

A partir de la información ya existente en la base de datos, y de las conclusiones encontradas durante la investigación, es posible formular una hipótesis acerca de la relación entre dos usuarios aleatorios. Algoritmos como estos están documentados en la literatura para un gran número de aplicaciones. Twitter inclusive, utiliza un algoritmo de cercanía o similitud de usuarios para ponderar el orden del feed de noticias, sugerir cuentas a seguir y realizar segmentaciones de usuarios. El algoritmo propuesto recibe dos id's de usuarios y usando información únicamente de la base de datos devuelve un valor de probabilidad.

Este valor es calculado a través de una suma ponderada entre varias señales:

a. Similitud de comunidades: Entre mayor sea la cantidad de seguidores/seguidos en común, mayor es la cercanía entre dos usuarios.
b. Similitud de menciones: Entre más veces un usuario mencione a otro, mayor es la cercanía entre ambos.
c. Similitud de retweets: Un retweet implica distribuir el mismo contenido que el autor original, por lo que entre más veces retweetee un usuario a otro, mayor es la cercanía entre ambos.
d. Similitud de likes*: Cuando un usuario le da like a un tweet, indica que le gusta su contenido y pueden tener algo en común.
e. Similitud de hashtags: Cuando dos usuarios comparten un hashtag, comparten interés por el mismo tema.
f. Intereses en común*: Entre mayores intereses en común existan, mayor será la cercanía entre ambos usuarios.
g. Similitud de perfiles: Entre más similares sean los metadatos de los usuarios, puede asumirse que más similares serán estos.

Por lo tanto, a partir de lo anterior, se obtiene la siguiente fórmula

$$
\begin{aligned}
p(u_i, u_j) = \quad & \alpha_1 \text{sim}_{\text{comunidad}}(u_i, u_j) \quad + \\
& \alpha_2 \text{sim}_{\text{menciones}}(u_i, u_j) \quad + \\
& \alpha_3 \text{sim}_{\text{retweets}}(u_i, u_j) \quad + \\
& \alpha_4 \text{sim}_{\text{likes}}(u_i, u_j) \quad + \\
& \alpha_5 \text{sim}_{\text{hashtags}}(u_i, u_j) \quad + \\
& \alpha_6 \text{sim}_{\text{topicos}}(u_i, u_j) \quad + \\
& \alpha_7 \text{sim}_{\text{perfil}}(u_i, u_j)
\end{aligned}
$$

Los resultados serán almacenados en la tabla UserRelationAnalysis, con los siguientes campos

- aUserID: Id del usuario A
- bUserID: Id del usuario B
- simCommunity Similitud por comunidad
- simMentions Similitud por menciones
- simRetweets Similitud de retweets
- simLikes Similitud por likes
- simHashtags Similitud por hashtags
- simTopics Similitud de temas de interés
- simProfile Similitud de metadatos de perfile
- executionDate Fecha en que fue ejecutado el análisis
- verificationDate Fecha en que fue verificado el análisis con la API
- verificationResult A partir de la verificación, indica si existe o no alguna conexión

Se utilizará una vista view_UserRelationAnalysisResults para calcular cada suma ponderada, así como el resultado final, con el objetivo de facilitar el ajuste de cada peso, con los campos.

A continuación se explicará a detalle cada uno de los análisis requeridos.

5.1. Similitud por comunidades

Se parte de la base de que los usuarios similares tienden a seguirse, y que una cuenta es similar a o cercana con sus seguidores. Por lo tanto, entre más similares sean los círculos de interacción de dos cuentas, mayor será la similitud entre éstas.

Similitud de comunidades directas

Entiéndase por comunidad directa la lista de todos los seguidos y seguidores de una cuenta[1]. Usando una vista, se obtienen las comunidades directas de ambos usuarios. A partir de ello, se calcula:

$$\text{sim}_{\text{comunidad directa}}(u, w) = \frac{|c_u \cap c_w|}{|c_u \cup c_w| - |c_u \cap c_w|}$$

Similitud de comunidades de segundo grado

Entiéndase por comunidad de segundo grado, la unión de las comunidades directas de cada elemento de la comunidad directa del usuario (los seguidos y seguidores de sus seguidos y seguidores). Se toman en cuenta únicamente los usuarios repetidos, y se excluyen los usuarios en común del paso anterior. Se obtiene para cada usuario y se cuenta la cantidad de usuarios repetidos en común.

$$\text{sim}_{\text{comunidad indirecta}}(u, w) = \frac{|c_u \cap c_w|}{|c_u \cup c_w| - |c_u \cap c_w|}$$

Resultado

Naturalmente, la similitud de comunidades directas tiene un mayor peso para determinar la similitud entre ambos usuarios, mientras que la similitud de comunidad de segundo grado ayuda a ponderar el peso. De tal forma se obtiene que la similitud por comunidades es:

$$\text{sim}_{\text{comunidad}}(u, w) = 0.75\text{sim}_{\text{comunidad directa}}(u, w) + 0.25\text{sim}_{\text{comunidad indirecta}}(u, w)$$

Tiempo de ejecución

Debido a la iteración necesaria para recuperar la similitud de comunidades de segundo grado, este análisis requiere un mayor tiempo de ejecución. Sin embargo,

[1] Si una cuenta aparece tanto en la lista de seguidos como de seguidores, se cuenta una única vez.

considerando la cantidad de datos existentes, esto podría significar unos cuantos milisegundos adicionales.

5.2. Similitud por menciones

Cada que un usuario menciona a otro en un tweet, existe un tipo de comunicación o relación entre estos. Tal relación se fortalece con mayor número de menciones.

Menciones a usuario

La vista view_UserMentions se seleccionan todas las menciones de un usuario a otro, ponderadas individualmente por el número de menciones del tweet. Posteriormente, se divide entre el número de menciones que ha hecho el usuario.

Resultado

El procedimiento anterior se realiza para ambos usuarios, y el resultado es dividido entre dos (para asegurar un valor entre 0 y 1)

Tiempo de ejecución

Realizar este análisis implica realizar dos consultas de lectura a la API, así como sencillas operaciones matemáticas, por lo que el tiempo de ejecución será de unos cuantos milisegundos.

5.3. Similitud por retweets

Cuando un usuario retuitea, distribuye el mismo contenido que el autor original, sugiriendo que existe algo en común. Por tanto, entre mayor sea la cantidad de retweets de un usuario al otro, mayor es la conexión entre estos.

Mediante la vista view_UserRetweet, se obtienen los retweets de un usuario a otro, y se dividen entre el número total de retweets de ese usuario. El procedimiento se ejecuta para ambos usuarios, dividiendo el resultado a la mitad para asegurar un valor entre 0 y 1.

Tiempo de ejecución

Considerando que la cantidad de retweets en la base de datos no es muy alta, y que únicamente se realizan dos consultas a la misma, la ejecución de este proceso es de unos cuántos milisegundos.

5.4. Similitud por hashtags

Cuando dos usuarios utilizan mucho el mismo hashtag, puede identificarse que comparten interés por un tema en común, incrementando la posibilidad de una conexión entre estos.

Usando la vista view_UserHashtags, se recopilan todos los tweets del usuario, y se seleccionan los 10 más repetidos. Se realiza el mismo procedimiento para el otro usuario. Se cuenta el número de coincidencias y se divide el resultado entre 10.

Tiempo de ejecución

Este procedimiento implica realizar dos consultas sencillas a la base de datos, por lo que debe completarse en unos cuántos milisegundos.

5.5. Similitud de perfiles y metadatos

Aunque con menor relevancia, pueden compararse diversos metadatos de los perfiles.

Ubicación: Regresa 1 si ambos usuarios tienen la misma ubicación

Lenguaje: Regresa 1 si ambos usuarios tienen el mismo lenguaje

Tiempo de cuenta: El porcentaje de tiempo que ambas cuentas han existido en Twitter

$$\frac{\min(t_u, t_w)}{\max(t_u, t_w)}$$

Resultado

Se suman las variables anteriores y se divide el resultado entre 3.

Tiempo de ejecución

Este procedimiento implica realizar únicamente dos consultas y comparar, por lo que su demora será prácticamente negligible.

5.6. Similitud por likes e intereses en común

Debido a falta de información en la base de datos que permita realizar estos ejercicios, ambos regresarán un valor de 1. Estos análisis se incluyen dentro de la estrategia para futura referencia.

5.7. Resumen

El análisis consta entonces de 7 sub-análisis, con las siguientes características.

Análisis	Tiempo de ejecución	Cantidad de información actual
Similitud por comunidad	Unos cuantos segundos	Poca
Similitud por menciones	Menos de un segundo	Amplia
Similitud por retweets	Menos de un segundo	Poca
Similitud por likes	0 segundos	Nula
Similitud por hashtags	Menos de un segundo	Amplia
Intereses en común	0 segundos	Nula
Similitud de perfiles	Unos cuantos milisegundos	Amplia

Por su naturaleza, estos análisis pueden realizarse de manera independiente, e incluso simultánea, por lo que es posible completarlos en un tiempo menor a un segundo.

El resultado de cada uno de estos análisis, así como la fecha de ejecución, serán entonces almacenados en la base de datos.

6. Ajuste de parámetros

Los análisis se almacenarán sin tomar en cuenta el peso asignado a cada uno. Esto permite modificarlos incluso una vez haya concluido el proceso. La literatura propone asignar los pesos siguiendo tres tipos de criterio: por contenido, por interacción y por personalidad.

	Peso		
Análisis	Por contenido	Por interacción	Por personalidad
Comunidades	0	10	0
Menciones	0	3	0
Retweets	2	0	1
Likes	2	0	0
Hashtags	1	0	0
Intereses	2	0	3
Perfil	0	0	2
Total	7	13	6

A partir de lo anterior, puede verse cómo la similitud por interacción es prominente, seguida de la de contenido. Eliminando las variables que no se analizarán, y ajustando por los datos faltantes de seguidores, se obtiene

	Peso		
Análisis	Por contenido	Por interacción	Por personalidad
Comunidades	0	5	0
Menciones	0	3	0
Retweets	2	0	1
Hashtags	1	0	0
Perfil	0	0	2
Total	3	8	3

Considerando la poca información disponible a analizar, y su poca relevancia para el estudio, se disminuirá el peso del perfil a 0.5. Los valores de la tabla anterior pueden combinarse para obtener

	Peso	
Análisis	Suma	Peso
Comunidades	5	0.4
Menciones	3	0.24
Retweets	3	0.24
Hashtags	1	0.08
Perfil	0.5	0.04
Total	12.5	1

Estos serán los pesos iniciales, pero pueden modificarse a voluntad, y pueden realizarse pruebas con resultados conocidos para verificar su margen de error.

7. Conclusión y aplicaciones futuras

Aunque la ejecución de esta propuesta no garantiza descargar todos los seguidores de los usuarios influyentes, permite identificar y cuantificar la conexión que existe entre dos usuarios arbitrarios, lo que puede ser de gran utilidad para otras áreas de esta misma investigación. De este modo, en un tiempo similar a un mes, será posible obtener las principales relaciones de usuarios de la muestra.

Igualmente, pude definirse un requerimiento mínimo de seguidores para reducir aún más la muestra y mostrar relaciones significativas.

8. Referencias

AlMahmoud, H., & AlKhalifa, S. (4 de Octubre de 2018). TSim: a system for discovering similar users on Twitter. *Journal of Big Data*. doi:https://rdcu.be/cxVUg

Goel, A., Sharma, A., Wang, D., Yin, Z., & Twitter, I. (s.f.). Discovering Similar Users on Twitter. Obtenido de http://snap.stanford.edu/mlg2013/submissions/mlg2013_submission_20.pdf

Iqbal, M. (5 de Julio de 2021). *Twitter Revenue and Usage Statistics (2021)*. Obtenido de Business of Apps: https://www.businessofapps.com/data/twitter-statistics/

Wojcik, S., & Huges, A. (24 de Abril de 2019). *Sizing Up Twitter Users*. Obtenido de Pew Research Center: https://www.pewresearch.org/internet/2019/04/24/sizing-up-twitter-users/