Universidad De Chile

Facultad de Ciencias Físicas y Matemáticas

LDA con datos de Twitter

Javier Castro Medina

Abril 2021

1 Análisis Exploratorio

En la primera parte del proyecto se realiza un proceso de adaptación a la librería `twint`, el formato de los datos que permite obtener y lo que podemos deducir de estos con un análisis exploratorio. Se decide buscar tweets en un radio de 10 km con centro en las coordenadas de la facultad, que a simple vista coincide con el centro de la región Metropolitana. La razón de esta decisión es porque en esta región hay una población más grande y por lo tanto más probabilidad de encontrar cuentas de twitter. La presente sección se divide en dos partes, en la primera se mostrarán worclouds y en la segunda gráficos de la frecuencia de tweets diarios.

1.1 Wordclouds

Acá mostramos dos hitos importantes, el primero es el estallido social del 18 de octubre del 2019 y el segundo es el día de la mujer del 2020. El tercer wordcloud que se muestra corresponde a los tweets que contienen la palabra "piñera" durante el 2020.

Figure (2) Tweets estallido social.

Figure (3) Tweets día de la Mujer.

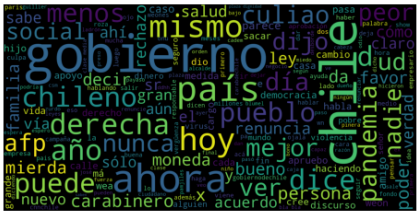

Figure (4) Tweets con la palabra "piñera" el 2020.

1.2 Series de tiempo

Se expondrán series de tiempo relativas a la cantidad de tweets diarios que contengan cierta palabra clave. El primero tiene que ver con la pandemia.

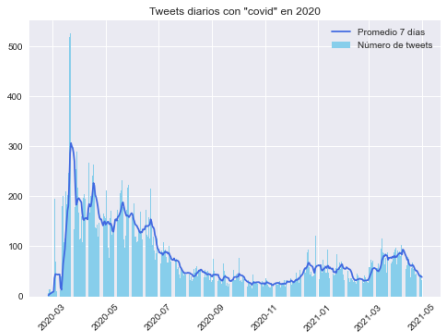

Figure (5) Tweets diarios con la palabra “covid”.

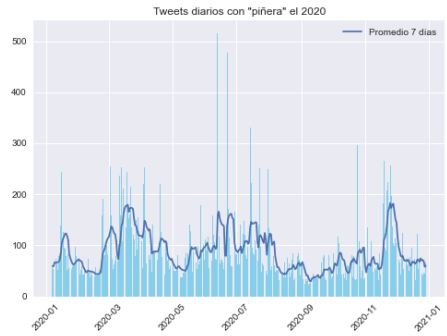

Figure (6) Tweets diarios con la palabra “piñera”.

2 Latent Dirichlet Allocation

Este algoritmo cae en la categoría del **topic modelling** en la cual se busca, a partir de datos de texto, encontrar tópicos latentes. Para este modelo se necesita un set de palabras V y una cantidad de tópicos K. Los tópicos se modelan como distribuciones sobre el conjunto de palabras V. Si bien este algoritmo tiene una fase generativa, su objetivo es calibrar parámetros $\alpha \in [0,1]^K$ y $\beta^{K\times V}$ optimizando una cota variacional. Estos parámetros determinan el modelo.

Primero se hace un review a los datos mostrando distintas estadísticas asociadas a estos. Luego, se expone la limpieza y preprocesamiento que se realiza sobre el texto para dejarlo en condiciones de ser usado por el modelo y los métodos de la librería `gensim`.

2.1 Datos

Podríamos aplicar el modelo a todos los tweets de cierto espacio y tiempo pero se inclina por otro approach que es trabajar con los tweets de una cuenta en específico dado que el rango de temas o tópicos tocados por una cuenta es menor o más controlado que al trabajar con tweets de muchas cuentas. La cuenta que se escoge para partir es `@gabrielboric`, obtenemos sus tweets hasta junio del 2021. Las características de este data set se pueden apreciar en las Figuras 6a y 6b

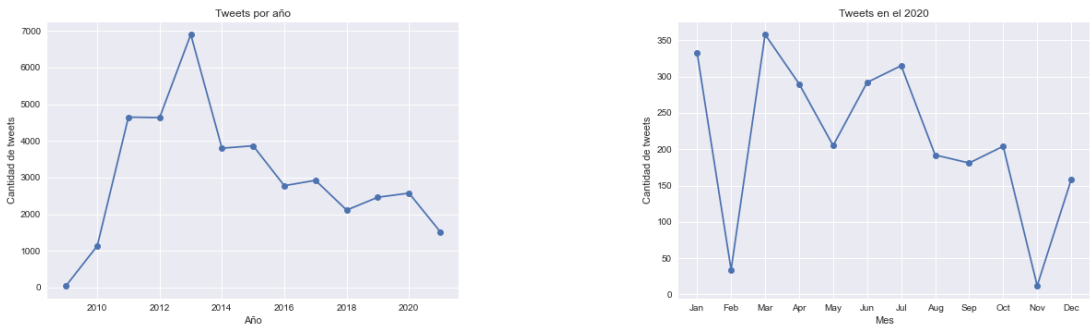

Figure 6: Tweets por año (izq) y por mes en el 2020 (der).

	Media	**Min**
Total	cell5	cell6
7 días	cell8	cell9

2.2 Limpieza y preprocesamiento de datos

Los datos que se tienen disponible corresponden a tweets en formato de texto. El procesamiento de base o inicial que se realiza sobre este corpus corresponde a eliminar **stop words**, links, palabras de largo menor o igual a 3, caracteres indeseables[1] y finalmente pasar todas las palabras a minúscula. El procesamiento anterior se realizará siempre y al inicio de todo código. Lo siguiente que se puede

[1] Se eliminaron: #, comas, signos de exclamación y pregunta, paréntesis y el signo igual.

hacer es quitar palabras de baja frecuencia, sin quitar estas palabras el diccionario[2] resultante tiene un tamaño de 35892 palabras. En el gráfico a continuación muestra la cantidad de palabras resultantes al variar este parámetro.

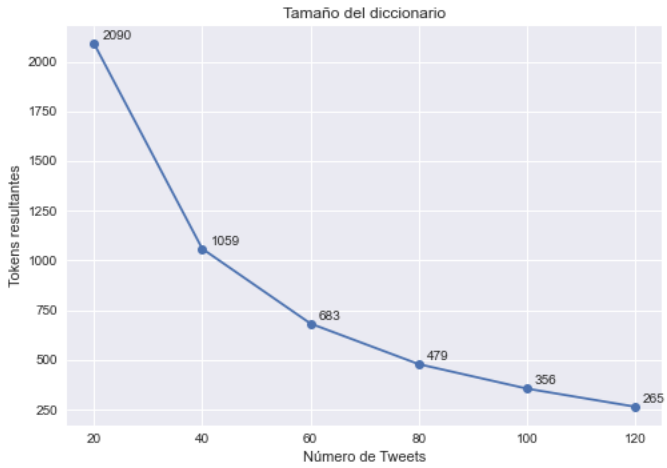

Figure 7: Tamaño del diccionario en función de la cantidad de tweets mínima a la que las palabras deben permanecer. Es decir, y corresponde a las palabras que resultan al filtrar las que aparezcan en menos de x tweets.

La siguiente etapa en el preprocesamiento es transformar los tweets en vectores. Para esto, a cada palabra del diccionario, digamos V, se le asigna un único índice de forma que el conjunto se palabras (o diccionario) se modela como $\{1, ..., |V|\}$. Luego, cada tweet, entendido como una colección de palabras en V de la forma $t = \{v_1, ..., v_n\}$, se transforma en $\{(v_1, r_1), ..., v_n, r_n)\}$ donde v_i es el índice de la palabra en V y r_i la cantidad de veces que v_i aparece en t. A continuación, un ejemplo de esto.

```
bow tweet 304:
 [(256, 1), (343, 1), (653, 1), (662, 1), (755, 1), (1078, 1), (1340, 1), (1457, 1), (1646, 1),
(1650, 1), (1805, 1), (1862, 1), (1870, 1), (2263, 1), (2264, 1), (2295, 1), (2434, 1), (2435, 1),
(2438, 1), (2454, 1), (2455, 1), (2456, 1)]
bow tweet 666:
 [(4, 1), (10, 1), (273, 1), (445, 1), (492, 1), (733, 1), (752, 1), (769, 2), (1008, 1), (1011, 1),
(1471, 1), (2274, 1), (4151, 1), (4152, 1), (4153, 1), (4154, 1)]
```

Figure 8: Estado final de dos tweets escogidos al azar. Esta representación se conoce como "bow" (**bag of words**).

Remark 2.1. *Notar que en los ejemplos que se muestran en la Figura 8 la segunda componente de las tuplas es mayoritariamente* 1. *Esto pasa porque el largo de cada tweet en el corpus es relativamente pequeño en comparación a otros set de datos como por ejemplo una colección de libros o notas periodísticas. Esto sugiere que esta variable no es muy decisiva en este contexto.*

Remark 2.2. *Se decide dejar las palabras que hacen referencia a cuentas de Twitter, por ejemplo* `@javier`. *La razón de esta decisión es que la aparición de una cuenta en un tópico nos indica que tal persona es relevante para la cuenta sobre la que se está trabajando.*

[2]Entendemos por diccionario al conjunto de todas las palabras utilizadas en el corpus

2.3 Aplicando el modelo

Primera Iteración: En la Figura 9 se muestra el output de pedirle los tópicos al modelo.

```
TOPIC: 0
WORDS: 0.010*"educación" + 0.009*"política" + 0.008*"chile" + 0.007*"acuerdo" +
0.006*"izquierda" + 0.006*"derecha" + 0.005*"gobierno" + 0.005*"puede" + 0.005*"mejor" +
0.005*"recomiendo"
TOPIC: 1
WORDS: 0.010*"saludos" + 0.008*"@jschaulsohn" + 0.008*"@valenzuelalevi" + 0.007*"alguien"
+ 0.006*"@jparedesgodoy" + 0.005*"quiere" + 0.005*"marcha" + 0.004*"bueno" +
0.004*"mercurio" + 0.004*"trata"
TOPIC: 2
WORDS: 0.014*"abrazo" + 0.013*"ahora" + 0.012*"magallanes" + 0.011*"buena" +
0.011*"mañana" + 0.010*"vamos" + 0.010*"fech" + 0.009*"arenas" + 0.008*"aguante" +
0.008*"punta"
TOPIC: 3
WORDS: 0.013*"muchas" + 0.012*"gracias" + 0.010*"@jen_abate" + 0.008*"@matiasdelrio" +
0.008*"compa" + 0.007*"feliz" + 0.006*"concerta" + 0.006*"@saladehistoria" +
0.005*"concertación" + 0.005*"buenos"
TOPIC: 4
WORDS: 0.012*"@cbellolio" + 0.010*"casa" + 0.008*"@lboric" + 0.007*"terrible" +
0.006*"facultad" + 0.006*"rector" + 0.006*"@donmatas" + 0.005*"universidades" +
0.005*"stgo" + 0.005*"superior"
```

Figure 9: En esta iteración aplicamos el modelo fijando los parámetros al ojo y sin filtrar palabras poco frecuentes.