Identificación de comunidades en intervalos de tiempo a través del lenguaje

Martín Igal Browarnik[1], Juan Manuel Ortiz de Zarate[1], and Esteban Feuerstein[1,2]

[1] Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Argentina
[2] Fundación Sadosky, Argentina

Abstract. En este trabajo desarrollamos distintas metodologías para entrenar modelos PLN que logren identificar comunidades en redes sociales exclusivamente a través de su jerga, es decir por el lenguaje que utilizan, a lo largo del tiempo. Analizamos 3 experimentos distintos logrando resultados con un ROC AUC superior al 0.8

Keywords: Modelos de PLN · Detección de comunidades · Redes Sociales.

1 Introducción

La identificación de comunidades es una tarea ampliamente estudiada desde diversas disciplinas hace ya largos años. En el famoso estudio de los 70' "An information flow model for conflict and fission in small groups" [1] Zachary analiza las comunidades que se encontraban latentes dentro de un grupo de Karate basándose únicamente en la estructura del grafo generado por sus relaciones.

Hoy las redes sociales juegan un rol sumamente preponderante en nuestras interacciones, influenciándolas directamente y produciendo, debido a sus algoritmos de segregación, "Filter Bubbles" [2] que agudizan la homofilia[3]. Para cada usuario, se genera entonces, un mundo de confort en el cual todo parece coincidir con sus puntos de vista, opiniones y gustos. Esto produce a su vez comunidades sumamente homofílicas a su interior, es decir con usuarios que coinciden entre sí fuertemente en diversos temas.

Continuando con lo investigado en [4], nuestra hipótesis es que en comunidades donde se observan grandes polarizaciones, como en el ámbito de la política, las jergas se vuelven sumamente específicas y distinguibles dentro de cada comunidad. En esos casos, mediante algoritmos de PLN, se podrían generar modelos que permitan una buena predicción basándonos exclusivamente en la forma de escribir.

En este trabajo, nuestro objetivo es encontrar la mejor estrategia para desarrollar modelos que permitan predecir la pertenencia de un usuario a una comunidad a lo largo del tiempo. Nuestra hipótesis es que en el área de la política y más específicamente en el contexto de las elecciones presidenciales, los cambios

de tópico (eventos específicos que se discuten durante unos pocos días) son algo frecuente, y de alguna manera contaminan la jerga propia de las comunidades, de naturaleza más estable.

Debido a esto, consideramos que seleccionar algunos días de cada semana del período de entrenamiento de manera uniforme nos permitirá capturar mejor la jerga que seleccionar el período completo ya que así se podría evitar el overfitting sobre los tópicos tratados en ese período.

Para validar nuestra hipotesis, tomaremos como casos de estudio, las discusiones en Twitter[3]en torno a las elecciones presidenciales en Argentina del 2019, las cuales se desarrollaron en un contexto de una muy alta polarización entre Cambiemos y el Frente de Todos. Para eso, entrenaremos diversos modelos de PLN mediante Fasttext[5] capaces de predecir la comunidad de pertenencia de los usuarios, utilizando únicamente los tweets publicados entre los meses de agosto y diciembre, es decir, desde las PASO hasta la asunción de Alberto Fernandez como presidente en Diciembre, utilizando a Walktrap[6] como ground-truth a la hora de medir su accuracy.

2 Metodología

Basándonos en el método desarrollado en [4], dividimos el proceso en 3 etapas. La primera es la obtención de datos utilizando Twitter como fuente, dado que esa aplicación provee libremente los textos de los tweets e información sobre las interacciones entre los usuarios. Mediante los retweets generaremos el grafo de interacciones entre los usuarios utilizando a los mismos como nodos y los retweets como aristas dirigidas que los comunican. Se obtienen entonces todos los tweets generados desde el 12 de Agosto del 2019, día posterior al día de votación de las PASO, hasta el 31 de Diciembre del mismo año utilizando como criterio el uso de la palabra *Macri*, dado que como presidente en este período, es uno de los ejes más importantes de la discusión electoral.

La segunda etapa consiste en utilizar a Walktrap en el grafo dirigido para identificar a las principales comunidades y luego etiquetarlas según su pertenencia cómo M o K. Para esto, previamente seleccionamos manualmente un conjunto de usuarios de cada comunidad para poder clasificar las comunidades obtenidas con Walktrap. Posteriormente normalizamos los Tweets, pasando su codificación a ASCII, convirtiendo en los emojis en texto, eliminando Links y caracteres especiales.

Para generar los modelos, seleccionamos los días a partir de los cuales queremos entrenarlos, eliminamos los textos duplicados y ejecutamos FastText en modo supervisado, utilizando *pretrained vectors* provistos por FastText[4] con información de la lengua española para mejorar la eficacia de nuestro modelo, esto creemos que resulta muy útil para modelos entrenados con pocos datos porque nos permite tener una mejor interpretación del lenguaje independientemente del contexto.

[3] https://www.twitter.com
[4] https://fasttext.cc/docs/en/crawl-vectors.html

Finalmente utilizamos los modelos entrenados para predecir la pertenencia de los usuarios a las principales comunidades, luego los clasificamos con Walktrap, como ground-truth, y calculamos la métrica de ROC para verificar la efectividad de estos modelos. Para mayor detalle, remitirse a [4].

3 Experimentos y resultados

Realizamos 3 experimentos distintos con el objetivo de evaluar el comportamiento de los modelos en diferentes contextos.

El primer experimento consistió en evaluar la capacidad de predicción dentro de un intervalo utilizando días de este como entrenamiento. Para tener una buena estimación de la performance, el ROC AUC de los días de entrenamiento se estimó mediante cross-validation utilizando 90% de los datos para entrenar y el 10% restante para testear.

En este experimento predijimos el mes de Diciembre utilizando solo 8 días, entrenamos un modelo A, con los días 1 al 4 y 28 al 31 de Diciembre y otro B utilizando 2 días de cada semana del mes es decir, 3,5,10,12,17,19,24,26 de Diciembre. En la Fig. 1, graficamos el ROC AUC por día de cada modelo. Se observa que, para el modelo A, el AUC promedio es de 0.78 y su SD de 0.13 mientras que para el modelo B, el AUC promedio es de 0.82 y su SD de 0.129

El segundo, consistió en entrenar al modelo con días anteriores para predecir los próximos 7 días, para esto creamos 2 modelos con 8 días, el primero A, con 8 días anteriores a la semana que queremos predecir, es decir, del 7 al 14 de Noviembre y el Modelo B, utilizando 2 días de cada semanas de las 4 anteriores, siendo estos los 15, 17, 22 y 24 de Octubre y 5,7,12 y 14 de Noviembre y evaluamos sobre los siguientes 7 días. En la Fig. 2a, graficamos el ROC AUC por día de cada modelo. Se observa que, para el modelo A, el AUC promedio es de 0.73 y su SD de 0.105 mientras que para el modelo B, el AUC promedio es de 0.796 y su SD de 0.023

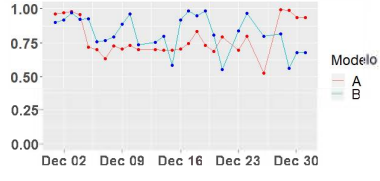
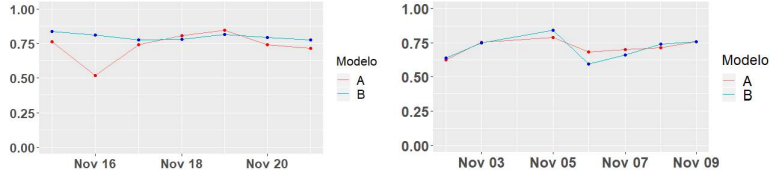


Fig. 1: AUC por día entrenando dentro del intervalo de predicción

El tercer experimento consistió en verificar si aumentar la cantidad de días de entrenamiento ayuda a mejorar la capacidad de predecir del modelo. Para esto

entrenamos un modelo A, con todos los días de Septiembre y Octubre e intentamos predecir en Noviembre y lo comparamos con otro B, usando solamente 2 días de cada semana de esos mismos meses, es decir los días 3,6,10,12,17,19,24,27 de Septiembre y los días 8,10,15,17,22,24 y 29 de Octubre. En la Fig. 2b, graficamos el ROC AUC por día de cada modelo. Se observa que, para el modelo A, el AUC promedio es de 0.716 y su SD de 0.054 mientras que para el modelo B, el AUC promedio es de 0.711 y su SD de 0.08.



(a) 8 días anteriores vs 2 de cada semana (b) 2 de cada semana vs el período completo

Fig. 2: Predicciones sobre los próximos 7 días

4 Conclusiones y trabajo futuro

Como era de esperar, si bien aumentar la cantidad de días es beneficioso para mejorar la performance del modelo, es importante elegir una buena estrategia para hacerlo, esto lo podemos ver en los resultados de los primeros 2 experimentos donde utilizando la misma cantidad de datos, la estrategia de seleccionar los días de entrenamiento de semanas distintas permite obtener un mejor resultado en la predicción y en el tercero, que a pesar de tener más datos y días de entrenamiento, el resultado no varía mucho.
En el primer experimento notamos que el predecir dentro del período de entrenamiento, nos permite obtener un mejor resultado, esto lo atribuimos a que los cambios de tópico no afectan tanto al predictor.
En los experimentos de predicción sobre los días futuros, detectamos que la performance del modelo empieza a disminuir pasados los 7 días, lo cual nos invita a pensar que sería necesario un re-entrenamiento para mejorar su funcionamiento.
Cabe destacar que como la metodología es agnóstica del lenguaje, a priori, podría aplicarse en cualquier contexto polarizado utilizando los *pretrained vectors* correspondientes.
Este trabajo preliminar nos alienta a pensar que existen formas de entrenar modelos confiables a lo largo del tiempo, para esto hay que seguir investigando cuáles pueden ser las técnicas de re-entrenamiento que nos permitan aumentar el poder de predicción a más de 7 días a futuro.

References

1. ZACHARY, Wayne W. An information flow model for conflict and fission in small groups. Journal of anthropological research, 1977, vol. 33, no 4, p. 452-473.
2. PARISER, Eli. The filter bubble: What the Internet is hiding from you.
3. MCPHERSON, Miller; SMITH-LOVIN, Lynn; COOK, James M. Birds of a feather: Homophily in social networks. Annual review of sociology, 2001, vol. 27, no 1, p. 415-444.
4. ORTIZ DE ZÁRATE, Juan Manuel; FEUERSTEIN, Esteban. Identificación de comunidades a través del lenguaje. En IV Simposio Argentino de GRANdes DAtos (AGRANDA 2018)-JAIIO 47 (CABA, 2018). 2018.
5. JOULIN, Armand, et al. Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759, 2016.
6. PONS, Pascal; LATAPY, Matthieu. Computing communities in large networks using random walks. En International symposium on computer and information sciences. Springer, Berlin, Heidelberg, 2005. p. 284-293.