Efecto presencia de fake news en sistemas recomendadores

Florencia Barrios
Pontificia Universidad Católica de Chile
Santiago, Chile
fbarrios1@uc.cl

Joaquín Tagle
Pontificia Universidad Católica de Chile
Santiago, Chile
jtagle2@uc.cl

***Abstract*— Hoy en día, el Internet y las grandes redes sociales como Facebook y Twitter han facilitado y acelerado el proceso de difusión de noticias. Por lo mismo, la desinformación o noticias falsas están cada vez más presentes en lo que leemos diariamente. Según Tambuscio et al., la desinformación corresponde a afirmaciones falsas que, en su mayoría, se propagan involuntariamente.**

El gran problema que surge de esto es que se ha demostrado que esta desinformación ha afectado por ejemplo, el precio de las acciones. Un caso real fue que luego de un *tweet* falso en donde se afirmaba que Barack Obama resultó herido en una explosión, el precio de las acciones de AP, una agencia de noticias de Estados Unidos y de donde provino el *tweet*, bajó en $130 billones de dólares. Poco después, AP dijo que su cuenta de Twitter había sido hackeada y el precio de las acciones logró recuperarse. Sin embargo, este no es un caso aislado de donde la desinformación tiene graves repercusiones. En las últimas elecciones de Estados Unidos, en donde Donald Trump venció ante todo pronóstico a Hillary Clinton, se cree fuertemente que la difusión de noticias falsas en Facebook y Twitter de ambos candidatos pudo haber afectado los resultados de estas elecciones.

Para contribuir a resolver este problema, se analiza cómo distintos algoritmos de recomendación impulsan la propagación de estas a través de las recomendaciones que hacen. Se presentan los *datasets* usados, la metodología, evaluación y finalmente, conclusiones y trabajo futuro.

I. INTRODUCCIÓN

En este trabajo, contribuimos a analizar el efecto de las *fake news* en distintos sistemas recomendadores, principalmente basados en contenido. En particular, según Vosoughi et al., la falsedad se difunde significativamente más lejos, rápido, profundo y ampliamente que la verdad en todas las categorías de información. Además, según estos mismos, las noticias falsas tienen un factor de novedad mayor que las noticias verdaderas por lo que se infiere que los usuarios tienden a compartir (o *retwittear* en el caso de Twitter) contenido novedoso.

Dado lo anterior, la hipótesis que planteamos es que en sistemas recomendadores no personalizados basados en popularidad, el porcentaje de noticias falsas recomendadas es mayor en relación a otros sistemas. Además, se analizará cómo se comportan dos algoritmos de recomendación basados en contenido: *TF-IDF* y *LDA* en presencia de *fake news*. Con esto, buscamos el algoritmo que propague la menor cantidad de noticias falsas puesto que así, estaremos evitando que el árbol de propagación de una noticia falsa siga creciendo y llegue a más usuarios y por otro lado, estaremos haciendo que el sistema sea más robusto puesto que se mantiene la estabilidad de la recomendación en presencia de información falsa. Como mencionan G. Shani y A. Gunawardana, cada día más personas confían en los sistemas de recomendación por lo que proveer un sistema robusto con información certera resulta ser imperante.

II. DATASET

Se utilizaron dos fuentes de información para el desarrollo de este *paper* y se describen a continuación.

En primer lugar, se utilizó un *dataset* obtenido del sitio Kaggle para entrenar al clasificador de noticias falsas. Este cuenta con 20.800 noticias que contienen título, autor, el cuerpo de la noticia y si son falsas o no, donde 49,9 % son reales y 50,1 % son falsas.

Por otra parte, para el sistema recomendador se extrajo 14.382 noticias provenientes de los perfiles de Twitter de BuzzfeedNews, Washington Post y el New York Times, con 242.098 usuarios y 712.558 relaciones indicando que usuarios hicieron *retweets* de las noticias. Las noticias fueron escritas entre el 3 de Octubre de 2018 y el 20 de Noviembre del 2018.

En promedio, cada noticia fue retuiteada por 44.5 usuarios distintos. En cuanto a la cantidad de *retweets* que tiene cada usuario, se tiene que en promedio cada usuario ha retuiteado 2.98 noticias. La Figura 1 muestra la distribución de cantidad de noticias que han retuiteado los usuarios.

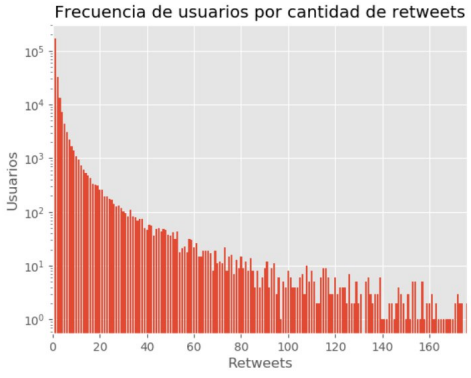

Fig. 1. Cantidad de lecturas por usuario en escala logarítmica

III. METODOLOGÍA

La metodología se divide en dos partes. La primera, fue la implementación de un clasificador para etiquetar las noticias según su veracidad y la segunda, la implementación de tres algoritmos de recomendación para ver como se comportaban en torno a las *fake news*: *Most Popular*, *TF-IDF* y *LDA*.

Para el etiquetado de noticias, se implementó un clasificador bayesiano. Este se utiliza para calcular la probabilidad posterior de que una noticia sea falsa condicional a los valores que ya se tienen de cuáles noticias son verdaderas y cuáles no. Luego, las probabilidades posteriores estimadas se utilizan para predecir la ocurrencia de noticias falsas en el conjunto de datos de prueba. El clasificador se basa en la regla descrita por la ecuación (1).

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)} \tag{1}$$

Donde A y B son eventos y:

- $P(A \mid B)$ es una probabilidad condicional: la probabilidad de que ocurra el evento A (que la noticia A sea falsa) dado que la noticia observada B es falsa.
- $P(B \mid A)$ también es una probabilidad condicional: la probabilidad de que ocurra el evento B dado que A es verdadero.
- $P(A)$ y $P(B)$ son las probabilidades de observar A y B independientemente el uno del otro.

Respecto a la técnica a utilizar para extraer la información de noticias, se decidió utilizar todo el texto de la noticia y no usar el título. Esto puesto que creemos que del texto podemos obtener una mayor cantidad de *features*. Para extraer las *features* de cada noticia, probamos por dos técnicas: *Bag of Words* y *Term Frequency–Inverse Document Frequency*.

La primera lo que hace es convertir una colección de textos (en este caso noticias) en una matriz de conteo de *tokens*, donde los *tokens* representan las palabras del corpus, sin incluir las *stopwords* (como *i*, *she*, *it*, etc). La segunda consta en computar un vector de valores *TF-IDF* para cada *token* del corpus. Este valor representa cuan relevante es el *token* (o palabra) para una noticia dentro de la colección de noticias y se calcula realizando el producto entre la frecuencia de término y la frecuencia inversa de documento. La ecuación (2) muestra el cálculo del valor para cada *token*.

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t) \tag{2}$$

TF representa la frecuencia del término dentro de un documento e IDF, que representa la frecuencia inversa del documento es calculado de acuerdo a lo presentado en (3).

$$\text{IDF}(t) = \log \frac{1 + n_d}{1 + DF(t, d)} + 1 \tag{3}$$

Donde n_d equivale al número total de documentos y $DF(t, d)$ corresponde al número de documentos que contienen al término t. Finalmente, los vectores resultantes son normalizados por la norma Euclideana presentada en (4).

$$v_{norm} = \frac{v}{||v||_2} = \frac{v}{\sqrt{v_1^2 + v_2^2 + \cdots v_n^2}} \tag{4}$$

Luego de analizar que clasificador tenía mejores resultados (presentados en la próxima sección), se procedió a etiquetar todas las noticias de nuestro *dataset* de Twitter según su veracidad. En total, se clasificaron 1164 noticias como falsas y 13218 como verdaderas, lo que resulta en un 8 % de noticias falsas.

Para analizar el comportamiento de distintos algoritmos de recomendación frente a las *fake news*, se implementaron tres de éstos. El primero fue *Most Popular*. Para éste, se consideró que las noticias más populares correspondieran a las con la mayor cantidad de *retweets*. Luego, las 15 más populares fueron recomendadas a todos los usuarios, sin personalización alguna.

Posterior a esto se implementaron dos algoritmos basados en contenido. En el caso de *TF-IDF*, se utilizó la librería `sklearn`. Primero se computa la matriz con los valores `TF-IDF` de todas las noticias según la ecuación (1) y luego se calcula la relación que tienen todos los textos utilizando *cosine similarity*, que lo que hace es calcular el producto punto L2-normalizado de los vectores. Es decir, si x e y son vectores de fila, su similitud de coseno se define por la ecuación (5).

$$sim(x, y) = cos(\theta) = \frac{xy^T}{||x||||y||} \tag{5}$$

Donde θ corresponde al ángulo entre ambos documentos en el espacio vectorial. Luego, a cada usuario se le hicieron 15 recomendaciones según la mínima distancia con las noticias que este previamente hizo *retweet*.

Para el algoritmo *LDA*, cada documento o noticia puede verse como una mezcla de diversos tópicos donde se considera que cada documento tiene un conjunto de tópicos que se le asignan a través de *LDA*. La Figura 2 muestra la explicación de *LDA*.

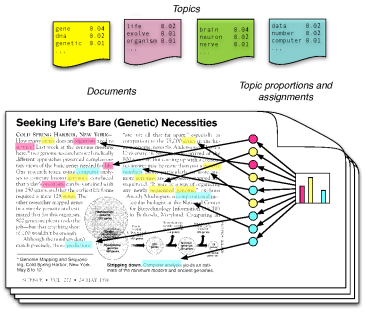

Fig. 2. Explicación *LDA*

Lo primero que se hizo fue generar un diccionario con todas las palabras del corpus utilizando la librería `gensim`. Luego, se generó un vector *bow* (*bag of words*) para cada documento y con eso se identificaron 10 tópicos en el *dataset*. Con esta información, se procedió a generar una matriz de 15 vecinos cercanos para poder identificar la similitud mediante la métrica coseno, descrita anteriormente por la ecuación (5). Finalmente, al igual que en el caso anterior, a cada usuario se le hicieron 15 recomendaciones según la mínima distancia con las noticias que este previamente hizo *retweet*.

IV. EVALUACIÓN

Para el etiquetado de noticias, se entrenaron clasificadores con ambas técnicas y obtuvimos una exactitud de 90.59 % utilizando la técnica de *Bag of Words* y una exactitud de 90.30 % utilizando *TF-IDF*.

En ambos casos, computamos la matriz de confusión para evaluar la exactitud la clasificación. Por definición, esta matriz C es tal que $C_{i,j}$ es igual al número de observaciones que se sabe que están en el grupo i pero que fueron predichas en el grupo j. En caso de clasificación binaria, la cantidad de verdaderos negativos (o noticias verdaderas) corresponde a $C_{0,0}$, la cantidad de verdaderos positivos (o noticias falsas) corresponde a $C_{1,1}$, los falsos negativos (o noticias que eran falsas pero fueron clasificadas como verdaderas) corresponde a $C_{1,0}$ y finalmente, los falsos positivos (o noticias que eran verdaderas pero fueron clasificadas como falsas) corresponde a $C_{0,1}$. La Figuras 3 y 4 muestran ambas matrices.

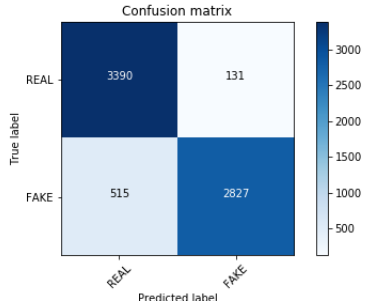

Fig. 3. Matriz de confusión utilizando Bag of Words

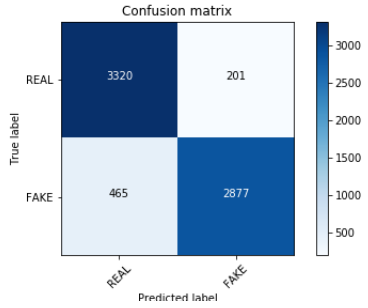

Fig. 4. Matriz de confusión utilizando TF-IDF

De las figuras, vemos que utilizando la técnica de *Bag of words*, hubieron 515 falsos positivos y 131 falsos negativos, mientras que utilizando *tf-idf* hubieron 465 falsos positivos y 201 falsos negativos. En términos de las métricas *precision* y *recall*, vemos que la primera técnica tiene una precisión de 88.3 % y un *recall* de 84.6 % mientras que la segunda tiene una precisión de 87.2 % y un *recall* de 86.1 %. Dado que nuestro fin es poder clasificar de manera correcta la mayor cantidad de noticias falsas (y con esto, reducir al máximo los falsos negativos), decidimos optar por la técnica que tiene un mayor *recall* (puesto que la proporción entre los verdaderos positivos sobre los verdaderos positivos y verdaderos negativos es mayor). Así, el etiquetado del *dataset* fue utilizando la técnica de *TF-IDF*.

Para analizar la hipótesis planteada, se analizan dos métricas, una para evaluar la calidad del recomendador y la segunda, para evaluar la robustez de éste. Para evaluar la calidad, se utilizó el MAP@15, que mide la calidad de un ranking. Este va del 0 al 1 y mientras mayor mejor. Se obtuvo que los recomendadores basados en contenidos son los que poseen mejores resultados siendo el algoritmo de *TF-IDF* el más alto, luego sigue *LDA* y finalmente, encontramos a *Most Popular*.

	MAP@15	% FKR	RFKR
TF-IDF	0.00034	6.27	0.78
LDA	0.000016	6.08	0.75
Most Popular	0.0000012	13.33	1.64

TABLE I
MÉTRICAS DISTINTOS ALGORITMOS

Respecto a la robustez de los distintos algoritmos, primero medimos el porcentaje de noticias falsas que recomienda cada uno (% FKR). *Most Popular* fue el que tuvo un peor rendimiento en este ítem, pues del total de recomendaciones que realizó, un 13.3 % eran noticias falsas, versus un 6.08 % y 6.27 % que obtuvo *LDA* y *TF-IDF* respectivamente.

Para analizar el factor de propagación que obtuvieron los distintos recomendadores, se analiza la relación entre el porcentaje de *fake news* que recomienda cada uno de los algoritmos, en comparación al porcentaje de noticias falsas que posee el set de datos (RFKR). De aquí, se obtiene que *Most Popular* no solo es el algoritmo que más noticias falsas está recomendando, sino que tiende a aumentar la propagación de las noticias falsas. Cada una noticia falsa que hay en el *dataset*, *Most Popular* está realizando 1.6 recomendaciones de noticias falsas, es decir, está impulsando a las noticias falsas a que sean más leídas. Por el contrario, los algoritmos basados en contenido es todo lo contrario, pues cada una noticia falsa que posee el *dataset*, están recomendando cerca de 0.8 noticias falsas, lo que significa que están restándole importancia a las *fake news*, y por ende frenando su propagación.

V. CONCLUSIONES Y TRABAJO FUTURO

Luego de implementar los distintos algoritmos y evaluarlos, se puede concluir lo siguiente. Comprobamos la hipótesis planteada, puesto que *Most Popular* es el algoritmo que recomienda la mayor cantidad de noticias falsas. Además, el problema de las *fake news* va en aumento, pero existen las herramientas para combatirlo y hay que continuar investigando en el tema, basándose no solo en las formas de propagación y en la popularidad de las *fake news*, sino también analizando el contenido mismo de las noticias.

Además, de este *paper* se abren nuevas líneas de investigación que valdría la pena ser estudiados. En particular, identificamos tres principales aristas. Por un lado, mejorar el sistema de clasificación de noticias viendo distintas formas. Puede ser implementar clasificador de noticias usando *Deep Learning*, o pedir a usuarios expertos que investiguen y clasifiquen noticias. De esta forma, el sesgo que se puede generar debido a un problema del *dataset* se pierde.

Otra arista, es ampliar el análisis incluyendo otros modelos de recomendación, pues en este *paper* no se hace ningún modelo que considere el contexto, lo que en el ámbito de la difusión de noticias es un aspecto a considerar, pues la difusión de las noticias de distintos temas, depende fuertemente del contexto en dónde se esté.

Finalmente, una última arista tiene que ver con realizar una caracterización de usuarios dentro del análisis, pues en este *paper* se manejan cifras macro, pero no se profundiza en caracterizar usuarios y ver cómo se ven afectados por las *fake news*. Puede ser que ciertos tipos de usuarios son más propensos a leer y difundir *fake news* bajo ciertos contextos. Se puede investigar si generar tipos de usuarios permite utilizar distintos recomendadores a los distintos *clusters* para así evitar de mejor forma la propagación de noticias falsas.

VI. REFERENCIAS

[1] M. Tambuscio, G. Ruffo, A. Flammini, F. Menczer, “Fact-checking Effect on Viral Hoaxes: A Model of Misinformation Spread in Social Networks”. En WWW’15 Companion Conference, pages 977-982, 2015.

[2] K. Rapoza, “Can ‘fake news’ impact the stock market?”. www.forbes.com/sites/kenrapoza/2017/ 02/26/can-fake-news-impact-the-stock-market/, Forbes, 26 february 2017.

[3] S. Vosoughi, D. Roy, S. Aral, “The spread of true and false news online”. Science 359, 1146–1151, 2018.

[4] G. Shani, A. Gunawardana, “Evaluating Recommendation Systems”. En Recommender systems handbook, pp. 257-297. Springer US, 2011.

[5] I. Levin, J. Pomares, R. Alvarez, ”Using Machine Learning Algorithms to Detect Election Fraud”. En R. Alvarez (Ed.), Computational Social Science: Discovery and Prediction (Analytical Methods for Social Research, pp. 266-294), 2016.