Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2433 — Minería de Datos — 2' 2021

Detección de *Fake News* utilizando herramientas de PLN y clasificadores basados en Naive Bayes y Decision Trees

Benjamín Vicente — Belén Silva — Daniela Corral — Matías Masjuan

1. Resumen

Actualmente, comprobar la veracidad de un artículo requiere de los servicios de *fast-checking*, que mediante criterio humano y ardua investigación logran desenmascarar las *fake news*. Es por ello que se ha intentado avanzar en la automatización de estos servicios de verificación, que lograría reducir la cantidad de artículos a revisar por estos expertos. Este paper comprende el proyecto de investigación realizado dentro del curso de Minería de Datos 2021-2, que ahonda en la temática de las *fake news* a nivel internacional y su importancia, profundiza en el área de la IA del procesamiento de lenguaje natural (PLN), utilizando algoritmos de Machine Learning tales como Naive Bayes y Decision Trees para clasificar noticias como falsas o no según su contenido. Se presentan los resultados obtenidos junto a una reflexión sobre los principales problemas que se tuvieron al momento de realizar este estudio, y se señalan posibles medidas a considerar en futuros estudios que podrían mejorar los resultados obtenidos.

2. Introducción

2.1. Definición del problema

Las *fake news* (en español, noticias falsas) son un término referido a la divulgación de noticias falsas que provocan un peligroso círculo de desinformación. En el siglo XXI, las redes sociales y el internet han permitido que cualquier persona pueda ser creadora de contenido para la web, lo que ha facilitado la viralización de contenido engañoso, falso o erróneo. En efecto, un estudio de la revista Science (2018) demostró que las *fake news* son retuiteadas por hasta 100 veces más personas y mucho más rápido que la información genuina, especialmente al hablar de política (Revista Science). Así, se generan círculos viciosos y una noticia falsa se puede replicar miles de veces en solo segundos. (FIP, p.1) [Vosoughi et al., 2018]

En el presente trabajo de investigación se busca analizar el problema de detección de *fake news*. Para esto, se utilizará un dataset con miles de noticias, tanto falsas como verdaderas para entrenar un clasificador que permita detectar noticias falsas dado su contenido, utilizando como clasificadores las técnicas de Naive Bayes y Decision Trees. La meta del proyecto es lograr ser una primera aproximación a un detector automático de *fake news* que permita alivianar el trabajo de los *fast-checkers*.

2.2. Impacto del problema

La circulación de *fake news* no solo desinforma a las personas impactando su derecho a la información, sino que además afecta a la sociedad en su conjunto y a la integridad democrática. Casos conocidos de *fake news* usualmente involucran elecciones presidenciales o decisiones políticas que afectan a países completos o, en el caso de las elecciones presidenciales de EE.UU., al régimen mundial. [Acevedo Rodríguez, 2020]

Por otro lado, la difusión de información falsa puede provocar como consecuencia aumentar la hostilidad y odio en contra de ciertos grupos vulnerables de la sociedad (FIP, p.11). Por ejemplo, si se desean ganar votos anti-inmigración, suelen publicarse noticias que adjudican aumento de crímenes o enfermedades de la mano al aumento de migrantes.

Finalmente, las *fake news* no solo buscan desinformar para influenciar grandes eventos, sino que están presentes en miles de sitios web diariamente con el objetivo de generar dinero. Para esto, las personas utilizan engaños relacionados con problemas o acontecimientos que preocupan a grandes masas de personas, estando dispuestos a incluso generar pánico colectivo con tal de ganar más *clicks* y *likes*. [Ballarini et al., 2020]

A modo de opinión del equipo de investigación, es innegable la enorme influencia que han ejercido las *fake news* hasta el momento, y todavía se desconoce el potencial impacto que podrían ser capaces de generar en eventos futuros. El equipo sostiene que es parte de su rol como ingenieros en computación poner el conocimiento al servicio de la comunidad (en este caso de los internautas) para poder brindar soluciones a problemas de ingeniería presentes en la vida real como éste y así hacer de internet y las RRSS mejores cada día.

3. Trabajo realizado

Se utilizará el set de datos del ISOT Fake News Dataset, *Fake and real news*, de Ahmed H., Traore I. y Saad S., que posee una colección de 21.417 noticias reales recolectadas desde el sitio de noticias *Reuters.com* en un archivo `"True.csv"`; y una colección de 23.481 artículos falsos recolectados desde sitios poco confiables verificados por la organización estadounidense de *fact-checking* Politifact en otro archivo `"Fake.csv"`. El plan es procesar las noticias con herramientas de PLN (*word embeddings*), y clasificar noticias como verdaderas o falsas de acuerdo a su contenido utilizando como clasificadores las técnicas de Naive Bayes y Decision Trees.

Como primer paso, se realizó una limpieza de estos datos. Se estandarizaron los formatos de las fechas, eliminando aquellas inválidas. Así también, se eliminaron filas con valores nulos y las filas duplicadas. Después de dicha limpieza, se obtuvo un *dataset* de 21.197 artículos reales y 17.902 *fake news*, lo que da un total de 39.099 noticias. La distribución de estas noticias por temática se describe por el siguiente gráfico:

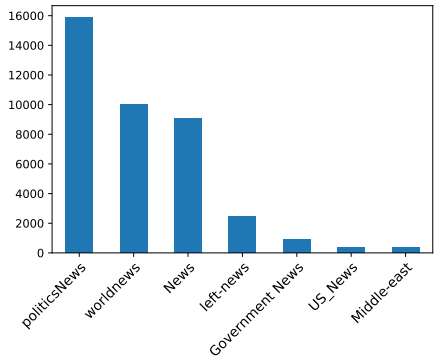

Figura 1: Distribución noticias por categoría

Nótese el alto número de noticias políticas por sobre el resto de categorías. La mayoría de las *fake news* políticas del *dataset* se relacionan a las elecciones presidenciales de EE.UU. del 2020 y a Donald Trump. Esta tendencia en las temáticas de las noticias podría inducir un sesgo en los clasificadores.

Para lograr extraer información de las noticias que pueda utilizarse en los clasificadores, se usó como herramienta de procesamiento de lenguaje natural el modelo pre-entrenado `en_core_web_lg` de spaCy. En particular, se utilizó la

técnica de *word embeddings* para vectorizar los textos de las noticias. El vector obtenido para cada frase es igual al promedio de los vectores que representan el *embedding* de cada palabra del texto.

Posterior a este procesamiento, se obtuvo un set que consta de una matriz X de $(17562, 300)$ con los *embeddings* de las noticias, y un vector y de $(17562,)$ donde las *fake news* tienen valor 1 y las noticias clasificadas como reales, 0. Dado que la matriz X solo tiene 300 *features*, no se consideró necesario aplicar técnicas de reducción de datos.

Finalmente, se separó este set de datos en uno de entrenamiento y otro de testeo, con una proporción 80 %/20 % respectivamente. A partir de estos datos se entrenaron dos clasificadores: uno basado en Naive Bayes y otro en Decision Trees. Cabe mencionar que a continuación se presentarán dos resultados por clasificador: uno considerando sólo los textos (cuerpos) de los artículos, y otro considerando los títulos de las noticias concatenados a los cuerpos.

3.1. Naive Bayes

El clasificador Bayesiano ingenuo es un clasificador probabilístico basado en la Teoría de Bayes, cuyo algoritmo busca optimizar la probabilidad de ocurrencia de que los elementos sean de cierta clase, asumiendo que dependen solamente de sus características y que sus características son independientes entre si. En esta ocasión, el algoritmo calculará la probabilidad de que cada noticia pueda ser considerada falsa o verdadera, y finalmente tomará la decisión basándose en cuál maximiza esta probabilidad.

Para el modelo del clasificador se utilizó `GaussianNB` de `sklearn.naive_bayes` con los hiperparámetros por defecto.

Con este modelo se alcanzó un **89.85 %** de *accuracy* en el set de test considerando sólo los textos de las noticias. Al considerar también los títulos, esta métrica disminuyó a un **88.43 %**. Respecto a las métricas de *precision* y *recall*, se observa que por lo general se detectan más a las noticias que son efectivamente reales que a las falsas.

Naive Bayes (solo textos)		
	Precision	Recall
False	0.90	0.91
True	0.89	0.88

Naive Bayes (títulos + textos)		
	Precision	Recall
False	0.89	0.89
True	0.87	0.87

Tabla 1: *Precision* y *recall* Naive Bayes

A partir de los resultados en el set de test, se generaron las matrices de confusión del clasificador, en las cuales se aprecian más equivocaciones al clasificar *fake news* como reales.

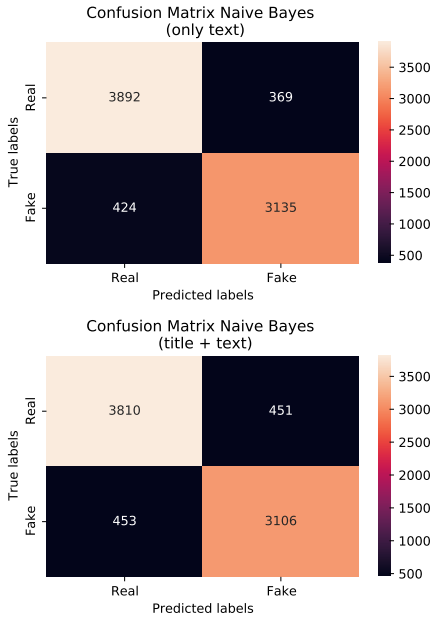

Figura 2: Matrices de confusión Naive Bayes

Finalmente, al profundizar sobre las noticias que fueron mal clasificadas, se puede ver que cerca del 50 % de estas son de temática política, seguidas por las mundiales y por las *left-news*. Dichos resultados pueden estar relacionados con la distribución de noticias en el *dataset*. Nótese también que al agregar los encabezados a las noticias, el porcentaje de noticias políticas mal clasificadas aumentó en un 6 %, lo

que podría ser un indicio de que los titulares confundieron más al clasificador respecto a las noticias políticas.

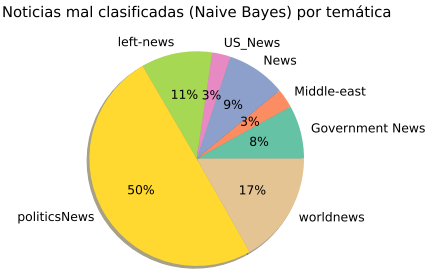

Figura 3: Noticias mal clasificadas por Naive Bayes por temática (sólo textos)

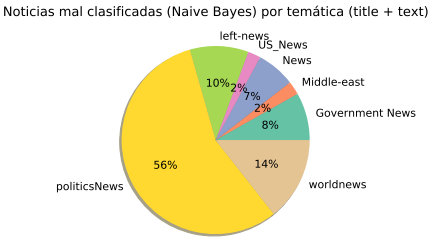

Figura 4: Noticias mal clasificadas por Naive Bayes por temática (titulares + textos)

3.2. Decision Tree

Los árboles de decisión son una técnica de aprendizaje supervisado en la que se recrea un árbol cuyos nodos internos representan atributos y sus conexiones/aristas representan valores de esos atributos. Las hojas de este árbol corresponderán a las clases que tiene nuestro problema. En este caso, para tomar decisiones el árbol deberá discretizar los atributos de la matriz X, que son en su totalidad numéricos.

Para el modelo de este clasificador se utilizó `DecisionTreeClassifier` de `sklearn.tree` con `max_depth=10` y el resto de hiperparámetros por defecto. Con este modelo se alcanzó un **89.71 %** de *accuracy* en el test considerando sólo los textos de las noticias, resultados muy similares a los obtenidos con el clasificador basado en Naive Bayes. De la misma manera, al considerar también los titulares de las noticias el *accuracy* disminuyó, pero en menor porcentaje que Naive Bayes, alcanzando un **89.2 %**. Respecto a las métricas de *precision* y *recall*, se observan las mismas tendencias que en Naive Bayes.

Decision Tree (solo textos)		
	Precision	Recall
False	0.90	0.91
True	0.90	0.88

Decision Tree (títulos + textos)		
	Precision	Recall
False	0.89	0.91
True	0.89	0.87

Tabla 2: *Precision* y *recall* Decision Tree

A partir de los resultados en el set de test, se generaron las matrices de confusión del clasificador, que se muestran a continuación.

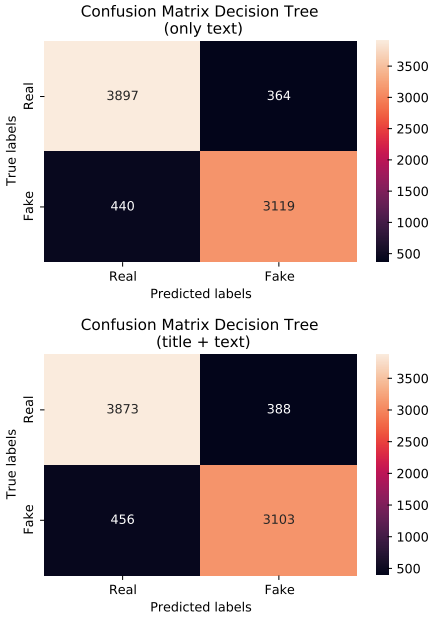

Figura 5: Matrices de confusión Decision Tree

Finalmente, al profundizar sobre las noticias que fueron mal clasificadas por este clasifica-

dor, nótese que, si bien las noticias en las que más se erró su clasificación fueron las políticas, el porcentaje de estas es un 5 %-10 % menor que en los errores de Naive Bayes, lo que da a entender que tal vez éste último tiene más dificultades con las noticias políticas, mientras que el Decision Tree distribuye un poco más sus errores en el resto de temáticas. Cabe mencionar que el efecto que tuvo añadir los titulares de las noticias se percibe bastante menor en este clasificador que en el de Naive Bayes, dado que la distribución de los errores y el *accuracy* apenas cambiaron al agregarlos.

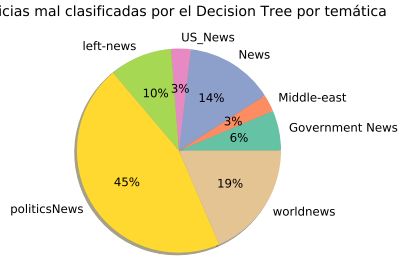

Figura 6: Noticias mal clasificadas por Decision Tree por temática (sólo textos)

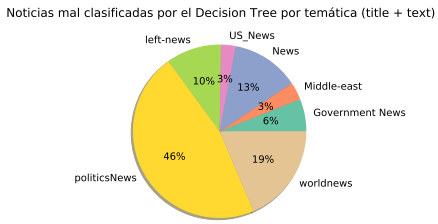

Figura 7: Noticias mal clasificadas por Decision Tree por temática (titulares + textos)

4. Análisis entrega anterior

En la entrega 2 de este proyecto, de carácter exploratorio, se limpió el *dataset* de noticias y se logró generar un algoritmo que transformaba los textos de éstas en vectores numéricos. Para aquel entonces, el equipo no contaba con muchos conocimientos previos sobre PLN, y sin notarlo se utilizó la librería SpaCy para procesar los textos mediante TDIDF *vectors*. El gran problema que se presentó en aquella oportunidad fueron los enormes tiempos de ejecución que esta técnica demandaba, que hicieron imposible la utilización de la totalidad del *dataset* para esa entrega.

Por lo anterior, el equipo decidió disminuir la cantidad de noticias con las que realizar ese análisis preliminar. En un comienzo sólo se pensó considerar las noticias políticas (∼17.000), pero como procesar sólo el 40 % de ellas tomó más de 5 horas, se redujo esta cantidad y se decantó por usar sólo 2.000 noticias. De ahí que se planteó como meta para la entrega final lograr entrenar los modelos utilizando el *dataset* completo, lo que requeriría investigación sobre formas más eficientes de procesar datos de tipo texto.

Pese a lo anterior, el equipo considera que el análisis preliminar realizado en la entrega anterior fue adecuado dado que, al utilizar un conjunto de datos más pequeño, en caso de no haber logrado una aproximación adecuada al problema, el equipo se ahorró largos tiempos de ejecución en procesos que podrían no haber sido los mejores para responder a la pregunta del proyecto.

Por otra parte, si bien el *accuracy* obtenido por los clasificadores Naive Bayes y árboles de decisión en aquella oportunidad fueron increíblemente buenos (sobre 98 %), esos rendimientos estaban sesgados por la cantidad de datos que se utilizaron, porque sólo se entrenaron los clasificadores con noticias políticas y porque no se incluyeron los titulares de las noticias en el análisis de datos. Por tanto, se concluyó que los resultados obtenidos no representaban clasificadores de noticias muy confiables, y se propuso trabajar en todos estos aspectos para la entrega final. Además, en el curso de Minería de Datos se explicaron nuevos métodos de procesamiento de texto que eran de interés probar en el proyecto de ser posible.

5. Análisis de resultados

A diferencia de la entrega 2, durante la nueva entrega se corrigieron los problemas ocurridos en el trabajo anterior. Esto se realizó a partir de una serie de cambios que afectaron al *dataset* trabajado junto al procesamiento realizado.

En primer lugar, el tamaño del *dataset* utilizado dejó de ser un conjunto de datos introductorio y preliminar, por lo que se utilizó el conjunto completo de 39.099 noticias en lugar de las 2.000 para la entrega anterior. Esto nos permitió obtener resultados representativos a partir de todo el conjunto estudiado, en lugar de solo una muestra minoritaria.

En segundo lugar, se utilizó la técnica de *word embeddings* para vectorizar los textos de las noticias haciendo uso de la librería spaCy. Esto permitió que el proceso de vectorización fuera mucho más eficiente que el uso de TDIDF *vectors*, lo que hizo posible tomar la muestra completa de datos.

Los modelos utilizados fueron el de Naive Bayes y el Decision Tree. El equipo de trabajo decidió utilizar estos clasificadores ya que Naive Bayes se considera bastante rápido para procesar datos de gran tamaño, y se caracteriza por asumir la independencia entre los atributos. Al estudiar datos de texto, las palabras no suelen generar dependencia entre ellas siempre y cuando se eliminen las *stop words* que generen este sesgo. Por otra parte, Decision Tree determina la clasificación de una instancia a partir de sus atributos, los cuales se concluyen a partir de una secuencia de nodos. Dado que la vectorización cuenta las palabras utilizadas para las instancias, es relativamente sencillo para el modelo separar los nodos y aristas a partir del valor de los atributos de una instancia, ya que se pueden determinar si son mayores o no a un cierto valor. Por lo tanto, Decision Tree se adapta correctamente a la problemática que se busca resolver.

En relación a los resultados obtenidos entre ambos modelos, se obtuvo un *accuracy* del **89.85** % para el caso de Naive Bayes, y un **89.71** % para Decision Tree. Dado que en la entrega 2 se obtuvo un *accuracy* por sobre el 98 % no se decidió realizar un cambio de modelos, ya que ambos presentaron resultados muy positivos.

Ahora bien, la razón de esta disminución se debe al uso completo del *dataset* en lugar del uso de una muestra de apenas 2.000 datos. En el caso anterior, estos datos correspondían a noticias de carácter político donde solamente se incluyó el cuerpo de la publicación como atributo, sin considerar el título de este. Esto pudo haber producido un sesgo muy grande en el estudio, ya que no existe forma de determinar si efectivamente las noticias políticas eran las que más contenían clasificación falsa, además de desconocer si la muestra era equitativamente representativa para los dos tipos de clasificaciones. Claro ejemplo de ello se podría explicar debido a que las noticias políticas tienen relación directa con fenómenos de *Donald Trump*, el cual fue caracterizado por utilizar medios de campaña a través de *fake news*. Por lo tanto, al considerar solo datos de este ámbito, existió la posibilidad de que la muestra haya concentrado publicaciones referentes a este personaje, sesgando los resultados.

Con respecto a la compración entre ambos modelos, se observa que el *accuracy* de Naive Bayes fue por muy poco superior al de Decision Tree. Sin embargo, si se compara el *accuracy* haciendo uso de los títulos de las publicaciones, obtenemos que Naive Bayes tuvo un *accuracy* de **88.43** % mientras que Decision Tree de **89.2** %, lo que deja superior a este último, sin mucha diferencia. La razón principal de esto se debe a que Naive Bayes trabaja asumiendo que los atributos son independientes. Sin embargo, esto no se cumple siempre, ya que los títulos de las publicaciones suelen estar relacionadas con el cuerpo y pueden contener palabras clave que sean dependientes entre cada atributo. Lo mismo aplica para los tipos de publicaciones. Se nota claramente cómo Naive Bayes clasifica mal las noticias políticas, mientras que Decision Tree falla de forma proporcional a la distribución de los tipos de noticias, lo cual es mas equiparado. Por esta razón, es posible explicar una leve disminución del rendimiento de Naive

Bayes en relación a Decision Tree.

6. Conclusiones

Es innegable que las *fake news* tienen un gran efecto en la sociedad, en la integridad democrática y en el derecho a la información de la ciudadanía; y proyectos como el presentado en este paper pueden ser un contundente avance para ayudar a reducir la desinformación de las RRSS y así alivianar la carga de los servicios de *fast-checking*.

A lo largo de esta investigación, se logró limpiar y pre-procesar con *word embeddings* de SpaCy un dataset de más de 40.000 noticias, lo cual era una de las metas iniciales y que costó trabajo conseguir a costa de pruebas con distintos métodos de PLN. También se construyeron y compararon los rendimientos de dos clasificadores: uno basado en el método de Naive Bayes y otro en árboles de decisión, obteniendo para ambos *accuracy* valores cercanos al 90 %. Con ello, se cumplió la meta del proyecto de construir un detector de *fake news* que, si bien puede pulirse más, cumple su función de manera simplificada pero confiable. Inclusive se analizó la importancia de los titulares de las noticias en la clasificación de éstas, lo cual fue una meta interpuesta en entregas anteriores del proyecto. De esta forma, el equipo se encuentra satisfecho con los frutos de la investigación.

A pesar de estos logros, es importante insistir en que la IA es una herramienta, y debe ser utilizada con mucho cuidado en esta área: el poder que es otorgado a quién realiza la distinción entre algo verdadero o falso es muy grande, por lo que esta responsabilidad no puede ser entregada a una máquina sin supervisión. Este proyecto busca ser una ayuda y no un reemplazo de los *fast-checkers*.

Para terminar la investigación de este proyecto, sería ideal trabajar con otros *datasets* de noticias que tengan más temáticas y mejor equilibradas en cantidad, y así diversificar más las fuentes de datos. El set de datos *Fake and real news* utilizado contaba con una alta concentración de noticias de temática política, y entre estas mismas destacan los temas relacionados a las elecciones presidenciales 2020 de EE.UU. y a Donald Trump; lo que indica el alto sesgo que induce el ocupar este *dataset*. Por otro lado, los datos de noticias reales provienen del sitio *Reuters.com*, el cual si bien puede ser confiable, podría ser un medio con alguna inclinación política. Esto no sería un aspecto negativo en el caso de que se tuvieran datos de noticias provenientes de variados medios de comunicación, pero ese no es el caso. Así, los algoritmos de *Machine Learning* utilizados podrían aprender ese sesgo y por ende clasificar como *fake news* a las noticias que no presentaran una inclinación política similar a la de *Reuters.com*.

Por otro lado, a modo de análisis adicional, sería interesante combinar los clasificadores que se utilizaron en el proyecto con *boosting* para mejorar el rendimiento general de la clasificación. Así, los modelos ensamblados tomarían en cuenta los errores cometidos por los anteriores y disminuiría el sesgo al clasificar.

Como trabajo futuro, es importante diversificar más las fuentes de datos para entrenar modelos que soporten múltiples fuentes de información, idiomas y tipos de textos (noticias, *tweets*, entre otros). Asimismo, se podría trabajar en encontrar indicadores que caractericen a las *fake news*, con el propósito de informar a la población sobre cómo identificar este tipo de artículos. También se podría mejorar la accesibilidad de estas herramientas a la sociedad utilizando, por ejemplo, por medio de una página web para proveer un análisis preliminar de un texto que ingrese el usuario.

Referencias

[Acevedo Rodríguez, 2020] Acevedo Rodríguez, C. (2020). £qué son las fake news?

[Ballarini et al., 2020] Ballarini, F., Bonnin, J. E., and Bekinschtein, P. (2020). £por qué decimos cosas que sabemos que no son verdad?

[Vosoughi et al., 2018] Vosoughi, S., Roy, D., and Aral, S. (2018). The spread of true and false news online. Science, 359(6380):1146–1151.