ISSN 1870-4069

Identificación automática de divulgadores de noticias falsas mediante el perfilado de autor

Cesar Macias, Miguel Soto,
Hiram Calvo, José E. Valdez-Rodríguez

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Laboratorio de Ciencias Cognitivas Computacionales,
México

{cmaciass2021,msotoh2021,
hcalvo,jvaldezr2018}@cic.ipn.mx

Resumen. El campo del procesamiento de lenguaje natural tiene una tarea encargada de realizar el perfilado de autores. Su objetivo es extraer características semánticas y de estilo de los textos que se analizan, para identificar a quién los escribió. En el marco del congreso *Conference and Labs of the Evaluation Forum* (CLEF) del 2020, los organizadores propusieron una tarea en conjunto con el grupo *Web Technology & Information Systems Group* (Webis), líderes de la organización PAN. Dicha tarea constaba en realizar el perfilado de los autores que divulgan noticias falsas en Twitter. En el presente artículo, se exploran diversos algoritmos de aprendizaje automático, con múltiples combinaciones de características del texto. Una de las ideas aquí exploradas es darle a los clasificadores la capacidad de generalizar la información, para su futura implementación en distintas plataformas sociales.

Palabras clave: Perfilado de autor, procesamiento de lenguaje natural, redes sociales, noticias falsas.

Automatic Identification of Fake News Spreaders through Author Profiling

Abstract. The field of natural language processing has a task of author profiling, which aims to extract stylistic and semantic features from the texts being analyzed in order to identify who wrote them. During the 2020 Conference and Labs of the Evaluation Forum (CLEF), the organizers proposed a task in conjunction with the Web Technology and Information Systems Group (Webis), leaders of the PAN organization. The task was to profile authors who spread fake news on Twitter. In this paper, various machine learning algorithms, with multiple combinations of text features, are explored. One of the ideas explored here is to give the classifiers the ability to generalize the information for future implementation on different social platforms.

Keywords: Author profiling, natural language processing, social networks, fake news.

1. Introducción

El perfilado de autor (PA) es una tarea importante en el procesamiento del lenguaje natural (NLP, por sus siglas en inglés), que consiste en inferir características de los autores de un texto a partir de su contenido, basándonos en las características de estilo y aquellas propias del texto.

La edad, el sexo, la personalidad y la ocupación del autor son algunas de las características que se suelen analizar. Esta tarea tiene múltiples aplicaciones, tales como en la verificación de la identidad de los autores, en la detección de la autoría de textos anónimos, en la clasificación de usuarios en redes sociales y en el análisis de la opinión pública, entre otras.

Desde su creación, la Internet se ha convertido en una de las principales herramientas para la consulta y divulgación de información. Actualmente, con el desarrollo y expansión de las redes sociales, la forma en la que los seres humanos interactuamos se ve muy afectada, ya que nos hemos acostumbrado a tener relaciones interpersonales exclusivamente en línea, mediante el uso de las redes sociales. Aunado a esto, la reciente pandemia de COVID-19 ha causado una revolución en este fenómeno, ya que se nos ha forzado a dialogar utilizando medios digitales exclusivamente.

Aproximadamente $4,760$ millones de habitantes utilizamos activamente las redes sociales como Facebook, Twitter o Reddit, lo que representa el $59{,}4\,\%$ de la población mundial; en promedio, cada uno de los usuarios activos invertimos dos horas, treinta y un minutos diariamente en esta actividad, de acuerdo con Digital 2023[1].

Las plataformas en línea intervienen cada vez más en el discurso público y los algoritmos nos ayudan a unirnos a grupos sociales, a clasificar el ruido del discurso público y estar al tanto de la actualidad [1].

Las redes sociales permiten a sus usuarios compartir fácilmente sus publicaciones favoritas con cientos de sus contactos, y, con el rápido crecimiento de las mismas, los motores de búsqueda nos facilitan la diversidad de voces al ofrecernos acceso a una amplia variedad de opiniones e información.

Pero, con tal cantidad de información, la variedad se mezcla con la ambigüedad, que al final produce un flujo de información impreciso. El manual *Journalism, "Fake news" & Disinformation* [8] clasifica al periodismo en:

- Periodismo de calidad. Cumple con los estándares profesionales y de ética.
- Periodismo débil. No cumple con los estándares profesionales ni éticos.
- Desinformación. Intentos deliberados de confundir o manipular a las personas mediante la entrega de información deshonesta.
- Información errónea. Información engañosa, creada o diseminada, sin intención manipuladora o maliciosa.

Las noticias falsas abarcan la desinformación y la información errónea. En [9], Kumar y Shah categorizan a la información falsa como basada en opinión (*opinion-based*) en la que no existe una verdad sólida y se presenta en casos como las reseñas de productos, o como basada en hechos (*fact-based*) que consiste en mentiras

[1] https://wearesocial.com/es/blog/2023/01/digital-2023/

sobre entidades que tienen un valor de verdad fundamental como las noticias falsas y los rumores.

Los autores involucrados en la creación de información, aprovechan las redes sociales para hacer la difusión de la misma, aprovechando la dinámica existente en las redes en la que los usuarios comparten las publicaciones que más les gustan o que les parecen más interesantes con cientos de sus contactos en muy poco tiempo.

Esta dinámica convierte a las redes sociales en el medio ideal para propagar la información. Los autores cuyo objetivo es crear desinformación o engañar a sus lectores integran a sus filas ejércitos de bots, encargados de hacer que las noticias falsas sean consideradas por la plataforma como temas en tendencia, ampliando el alcance que pueden llegar a tener [9], es por ello que es importante darle solución a esta problemática.

El resto del trabajo se encuentra organizado de la siguiente manera: la sección 2 describe los trabajos anteriormente desarrollados que mantienen una estrecha relación con el tema aquí tratado. La sección 3 describe la metodología propuesta para desarrollar la investigación, y sus experimentos.

La sección 4 muestra los resultados obtenidos tras la experimentación, la comparación entre los resultados aquí obtenidos y aquellos mostrados en el estado del arte. Finalmente, en la sección 5 proveen las conclusiones y se listan las ideas para desarrollar como trabajo futuro.

2. Estado del arte

La basta propagación de noticias falsas en años recientes ha causado mucha atención por parte de la comunidad científica y de la industria. A lo largo de esta sección, describiremos algunas de las más recientes aportaciones que se han realizado para contrarrestar este problema.

En 2021, Sahoo y Gupta [19], recuperan información de noticias falsas, perfiles que comparten este tipo de noticias y características del contenido compartido mediante la *Application Programming Interface* (API) de Facebook[2] y utilizan esta información para entrenar modelos de aprendizaje automático clásicos y de aprendizaje profundo; sus mejores resultados fueron obtenidos con un modelo *Long sort-term memory* (LSTM) [6] con una exactitud de 99.4 %.

En ese mismo año, Zhang y colegas [21], proponen usar características de emoción duales, las cuales hacen referencia a la emoción de quien publica la noticia y de los lectores. Para la clasificación hicieron uso de distintos modelos basados en *transformers* [20] los cuales ensamblaron, de tal manera que su mejor aproximación obtuvo un 93.2 %.

Para 2022, Raza y Ding [18], proponen un enfoque basado en *transformers* para la detección de noticias falsas, utilizando un conjunto de noticias de múltiples fuentes y los respectivos contextos sociales de los consumidores de estas noticias.

Esta información fue combinada de tal manera que se generó una representación vectorial, la cual sería su bloque de entrada para su bloque del *transformer* y

[2] https://www.facebook.com

posteriormente esta información sería enviada a una capa de clasificación. Su modelo, al que llaman *Fake News Detection through News content and Social context* (FND-NS) obtuvo una exactitud de 74.8 %.

Davoudi y colegas [5], presentan un modelo profundo híbrido para la detección de noticias falsas mediante el uso de un árbol de propagación y un sistema de apoyo de decisiones (DSS). Dichos componentes se ejecutan de manera simultánea, y con ellos buscan encontrar las características con mayo valor discriminativo, la diferencia que existe entre el patrón y las características extraídas a lo largo del tiempo; su modelo obtuvo una exactitud de 98.4 %.

2.1. Enfoques basados en el perfilado de autor

La intención de estas aproximaciones es construir la reputación de un autor al compartir una noticia y con base en las características extraídas de este autor, identificar si es un posible dispersor de noticias falsas.

A lo largo de la última década, el CLEF, en conjunto con el PAN, han propuesto distintas tareas para el PA. Durante el marco del congreso CLEF del año 2020 [15], serían unos de los pioneros en proponer la identificación de noticias falsas mediante el PA, y propusieron una tarea titulada *"Profiling Fake News Spreaders on Twitter"* en el cual participaron 66 equipos, los cuales propusieron el uso de distintas combinaciones de características para perfilar autores.

Por mencionar algunas aproximaciones de este congreso, Pizarro [13] combinó n-gramas de palabras y caracteres, y obtuvo una exactitud promedio de 77.75 %. Por otro lado, Manna y colegas [10] combinaron el número promedio de emojis (clasificados en categorías como afecto, emoción, consternado, etc.), el número de signos de puntuación, *tags*, espacios, enlaces web, etcétera, y obtuvieron una exactitud promedio de 66 %.

Sin embargo, su mejor aproximación para esta tarea ese año fue llevada a cabo por Buda y Bolonyai [3] quienes obtuvieron una exactitud de 77.75 % y lo lograron realizando un ensamble de modelos de aprendizaje automático clásicos haciendo uso de n-gramas con algunas estadísticas para los tuits, así como de la longitud promedio de la diversidad léxica del conjunto de datos.

Sin embargo, el enfoque de PA para la detección de noticias falsas ha continuado hasta la fecha, ya que un año más tarde, en 2021, Rathod [17], retoma esta línea de investigación, enfocándose en una construcción más sólida de un perfil de autor utilizando características de autoría, *Named Entity Recognition* (NER), sentimiento y emociones y estilometría.

A diferencia de las aproximaciones realizadas en CLEF 2020, en este trabajo utilizaron un conjunto distinto de datos y obtuvieron una exactitud de 85 %. En 2022, Cervero y colegas [4] continúan con el uso de la información utilizada en [15]; uno de los grandes aportes de este trabajo es que logran extraer información visual de los tuits del conjunto de datos mediante la API de Twitter. Con esta información lograrían un aumento de datos y consigo una exactitud promedio de 80.5 %.

Tabla 1. Distribución de clases, conjunto de entrenamiento y prueba

Conjunto	**Inglés**	**Español**	
Entrenamiento	300	300	
Prueba	200	200	
Total	500	500	1,000

3. Metodología

En esta sección se describirá la metodología seguida para desarrollar la investigación del presente trabajo. Se presenta una variedad de algoritmos clásicos de clasificación del aprendizaje automático, se describe el preprocesamiento que se le dio a los textos, y los métodos de extracción de características: TF·IDF y los embeddings de palabras. Se pretende abordar la tarea "*Profiling Fake News Spreaders on Twitter 2020*" propuesta en el marco del congreso *Conference and Labs of the Evaluation Forum* (CLEF) en su edición 2020.

3.1. Corpus

El corpus seleccionado para realizar la tarea del perfilado de autores, fue el desarrollado por PAN[3] para el congreso CLEF 2020 [16]. El corpus contiene un total de $1,000$ autores, de los cuales 500 son autores de textos en inglés y 500 son autores de textos en español.

Para cada autor se recolectaron un total de 100 tuits etiquetados como 0 o 1, indicando si el texto corresponde a una noticia falsa o no. La distribución de clases del corpus se muestra en la Tabla 1. La descripción detallada del desarrollo del corpus y sus componentes se encuentra en [15, 16].

3.2. Preprocesamiento

Para el preprocesamiento de los textos, se utilizó un script desarrollado por los autores en conjunto[4]. Las operaciones realizadas en el script se describen de forma general a continuación.

- Entidades HTML: se remueven las entidades HTML que contenga el texto.
- Saltos de línea: se quitan los saltos de línea.
- Hashtags: En caso de haber hashtags, se separa el texto contenido en los mismos (p.e. #NoticiasFalsas $\rightarrow$ Noticias Falsas).
- Entidades de Twitter: se les dice así a las entidades que se utilizan propiamente en Twitter para denotar usuarios, etiquetas, hashtags y retuits, cada uno de estos tiene un identificador especial (@User, rt, #hashtag), se identifican estas entidades y se remueven del texto.
- URLs: se identifican y se remueven del texto.
- Las letras se convierten a minúsculas para homogeneizar el texto.

[3] https://pan.webis.de/

[4] https://github.com/CCogS-Mx/text-preprocessing

Tabla 2. Configuraciones de la experimentación.

Vectorizadores		**Modelos**	
TF·IDF	Secuencias de n-gramas de 1 a 3 (1G, 2G, 3G). Frecuencia mínima de aparición de palabras de 1 y 3 (1FM y 3FM).	LR	`penalty`: 'L2' `solver`: 'liblinear' `max_iter`: 10000
		LSVC	`penalty`: 'L2' `max_iter`: 10000
Word embeddings	Inglés: fasttext-english-twitter-100d [11] Español: fasttext-english-twitter-100d [11]	RF	`n_estimators`: 100 `max_depth`: 100

– Palabras auxiliares: en caso de que así se requiera, se remueven las palabras auxiliares que contenga el texto.
– Lematización: si se requiere, las palabras son lematizadas utilizando la librería spaCy [7].
– Apóstrofes: tras la lematización, se remueven los apóstrofes del texto, conservando el caracter sin el apóstrofe (p.e observación → observacion).
– Puntuación: se remueven los caracteres utilizados para puntuar el texto (puntos, comas, punto y comas, etc.).
– Caracteres repetidos: en caso de que un caracter se repita más de dos veces, este se corta a dos repeticiones (p.e. Holaaaaaaa → Holaa).
– Palabras alfanuméricas: si el texto contiene palabras compuestas por letras y números, como en el leet speaking, estas se remueven (p.e. P3*r4).
– Caracteres especiales: se remueven todos los caracteres especiales, signos de admiración, interrogación, etc.
– Espacios en blanco: en caso de que exista más de un espacio en blanco entre palabras, estos se remueven para homogeneizar el texto.

3.3. Extracción de características

Una vez terminado el preprocesamiento, se realizó la extracción de características y tokenización del texto. Para tal efecto, se utilizaron los métodos TF·IDF y embeddings de palabras.

TF·IDF. EL método TF·IDF hace uso de la frecuencia de los términos (TF, por sus siglas) y la frecuencia inversa de documento (IDF, por sus siglas). Para el cálculo de TF, se genera una bolsa de palabras con el vocabulario del conjunto de todos los documentos que van a ser analizados, posteriormente se obtiene el total de apariciones de la palabra en el documento analizado. Para calcular IDF se calcula el logaritmo del cociente de la cantidad de documentos en los que la palabra analizada aparece y la cantidad total de documentos en el conjunto analizado. Finalmente, se realiza el producto entre TF e IDF para cada una de las palabras del texto analizado, generando un vector de características.

***Embeddings* de palabras.** El *embedding* de una palabra, es la representación, usualmente representada por un vector n-dimensional, en la que se codifica el significado de la palabra, asignando una posición dentro de un espacio de

Tabla 3. Mejores resultados para los modelos vectorizados por TF·IDF medidos por la métrica *accuracy*.

		LR		**LSMV**		**RF**	
Característicasdel texto	Combinación	EN	ES	EN	ES	EN	ES
Texto en crudo	1g + 1FM	0.62	0.65	0.61	0.63	0.60	0.67
	1g + 2g + 1FM	**0.63**	**0.68**	**0.63**	**0.66**	**0.61**	0.67
	1g + 2g + 3g + 1FM	0.62	0.67	**0.63**	**0.66**	**0.61**	0.66
	1g + 3FM	0.61	0.65	0.60	0.63	0.60	0.67
	1g + 2g + 3FM	0.62	**0.68**	0.62	0.65	0.60	**0.68**
	1g + 2g + 3g + 3FM	0.62	**0.68**	0.61	0.65	0.59	**0.68**
Texto con preprocesamiento	1g + 1FM	0.60	0.65	0.60	0.63	**0.61**	0.67
	1g + 2g + 1FM	0.62	**0.68**	0.62	**0.66**	**0.61**	0.67
	1g + 2g + 3g + 1FM	0.62	0.67	0.62	**0.66**	**0.61**	0.66
	1g + 3FM	0.60	0.65	0.60	0.63	**0.61**	0.67
	1g + 2g + 3FM	0.62	0.67	0.61	0.65	**0.61**	**0.68**
	1g + 2g + 3g + 3FM	0.62	0.67	0.62	0.65	**0.61**	**0.68**

características, utilizando las palabras con significados más similares. Este vector de características, nos indica la zona del espacio de características en la que se ubica la palabra analizada, por su similitud con otras palabras.

3.4. Modelos propuestos

Para realizar la clasificación, se propusieron tres diferentes algoritmos de clasificación del aprendizaje automático clásicos, además estos modelos son continuamente utilizados en la literatura para afrontar esta tarea.

Los algoritmos de clasificación son: regresión logística (LR), máquina de soporte vectorial con kernel lineal (LSVM) y bosque aleatorio (RF). A continuación se da una descripción general de cada uno de los algoritmos y su funcionamiento.

Regresión logística (LR). Es un modelo estadístico que estima la probabilidad de que un evento ocurra. Para el caso binario, este modelo calcula la probabilidad $[0, 1]$ de que una muestra i pertenezca a la clase y_i. La representación matemática del modelo es la siguiente [12]:

$$P(y_i = 1|X_i) = \frac{1}{1 + exp(-X_i w - w_0)}, \tag{1}$$

donde X representa el tuit; y denota la etiqueta de clase; $w \in \mathbf{R}^n$ es el vector de pesos.

Máquina de soporte vectorial con kernel lineal (LSVM). La máquina de soporte vectorial con kernel lineal (LSVM) hace uso del kernel lineal para permitir a la SVM, operar en espacios de alta dimensión, haciendo la transformación de dimensiones con este kernel. Las SVMs son algoritmos de clasificación cuyo objetivo es encontrar el hiperplano de separación óptimo entre clases. Este hiperplano actúa como frontera de decisión para asignar la clase final a un dato de entrada que deba ser clasificado. El hiperplano es encontrado al maximizar el margen (distancia entre el hiperplano y los vectores de soporte).

Tabla 4. Mejores resultados para los modelos vectorizados por *word embeddings* medidos por la métrica *accuracy*.

	LR		**LSVC**		**RF**	
Características del texto	EN	ES	EN	ES	EN	ES
Texto en crudo	0.57	**0.61**	0.57	**0.61**	0.57	**0.61**
Texto con preprocesamiento	**0.59**	0.52	**0.59**	0.52	**0.61**	0.52

Bosque aleatorio (RF). Un bosque aleatorio es un algoritmo de clasificación que consiste en acoplar un conjunto de clasificadores estructurados en forma de árbol $\{h(x, \Theta_k), k = 1, \ldots\}$ en el que $\{\Theta_k\}$ son vectores aleatorios independientes, distribuidos de forma idéntica y, en el que cada árbol obtiene un voto unitario para la clase más popular para la entrada x [2]. La librería `scikit-learn` [12] implementa este bosque de árboles aleatorios como un ensamble de árboles construido de una muestra del conjunto de entrenamiento, que divide los nodos de cada árbol de un subconjunto aleatorio de tamaño igual a las características que se estén extrayendo. En el que la predicción probabilística de cada árbol se promedia para obtener la clase final.

3.5. Métricas de evaluación

En [15], se detalla la forma en la que se va a evaluar a los participantes de la tarea. La métrica de desempeño propuesta por los organizadores, que además se utilizó en este artículo para comparar con los resultados del estado del arte, es la exactitud (*accuracy*) que se obtuvo para cada uno de los lenguajes contenidos en el corpus. El cálculo de esta métrica se describe matemáticamente en la ecuación 2:

$$accuracy = \frac{\text{Numero de predicciones correctas}}{\text{Total de predicciones}} = \frac{V_P + V_N}{V_P + V_N + F_P + F_N}, \quad (2)$$

donde V y F corresponden a Verdadero y Falso y los subíndices P y N corresponden a positivo y negativo. Por lo tanto, V_N se lee como verdaderos negativos.

4. Experimentos y resultados

En esta sección se describen los experimentos realizados durante el desarrollo de la investigación, la configuración de cada uno de estos, los resultados obtenidos, así como una breve discusión de los mismos.

4.1. Experimentos

Para realizar nuestras aproximaciones, hicimos uso de los 3 modelos descritos en la sección 3. Para implementar y entrenar los modelos se utilizó la librería de `scikit-learn` [12] para Python (versión 3.9.5). Con la finalidad de que nuestros experimentos puedan ser replicables, asignamos la semilla 42 para la generación de números aleatorios.

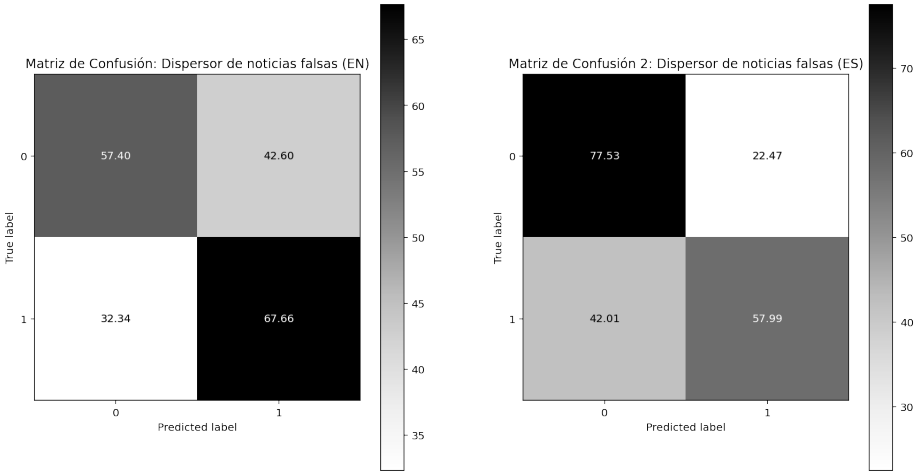

Fig. 1. Matrices de confusión del mejor modelo (RL), obtenido para ambos casos: inglés (izquierda) y español (derecha).

Se llevaron a cabo pruebas con varias configuraciones para cada modelo, utilizando diferentes secuencias de palabras en n-gramas, distintos hiperparámetros y dos opciones de *word embeddings*. Asimismo, se realizaron experimentos con el texto sin procesar y con el preprocesamiento propuesto.

Los modelos propuestos para realizar la clasificación, así como las combinaciones de los métodos de extracción de características (TF·IDF con secuencias de n-gramas y *embeddings*) se muestran en la tabla 2.

4.2. Resultados

Durante el proceso de experimentación, realizamos múltiples iteraciones variando los distintos hiperparámetros en cada modelo. Estos experimentos nos permitieron observar como los cambios en los parámetros afectan su desempeño.

A pesar de que obtuvimos resultados interesantes y valiosos en nuestro estudio del problema, nos enfocaremos únicamente en los mejores resultados, ya que nuestra intención es comparar este trabajo con el estado del arte.

En este sentido, presentaremos los modelos que lograron obtener un mayor *accuracy* en cada tarea, así como sus configuraciones respectivas. Estos resultados se pueden observar en las tablas 3 y 4, dónde, las mejores configuraciones fueron las siguientes:

En el caso de los modelos para los que se hizo uso de la vectorización TF·IDF, el rango de n-gramas que se utilizó, y el que mejor desempeño obtuvo, fue con secuencias $(1, 2)$, es decir, unigramas y bigramas con el modelo LR, con una exactitud promedio de ambos idiomas de 66 %.

Por otro lado, los mejores resultados que obtuvimos con los *word embeddings* se dieron con el modelo de RF, sin embargo, para lograr el mejor ensamble, se tuvieron

Tabla 5. Comparación contra los mejores resultados del estado del arte.

Autor	Inglés	Español	Promedio
Buda y Bolonyai [3]	**0.750**	0.805	0.7775
Pizarro [14]	0.735	**0.820**	0.7775
Nuestra propuesta	0.630	0.680	0.6550

que unificar los modelos con texto preprocesado y con texto en crudo para el idioma inglés y español respectivamente, obteniendo una exactitud promedio de 66 %.

De esta manera, nuestra mejor aproximación de todos los experimentos que se realizaron fue el modelo de RL con la combinación de unigramas, bigramas y frecuencia mínima de aparición de palabras de 1. Lo cual nos indica que para el modelo hay palabras o secuencias de palabras que, a pesar de que aparezcan solo una vez en el conjunto de entrenamiento, pueden ser de gran importancia cuando se prueban sobre el conjunto de pruebas.

En la Figura 1, se pueden visualizar las matrices de confusión del mejor modelo para cada uno de los idiomas. Para el caso del inglés, se logra observar que la clasificación de la etiqueta 0 es muy ambigua, ya que tiende a clasificar 57.40 % de los datos bien; para la etiqueta 1, esta clasificación mejora significativamente, otorgando un 67.66 % de clasificación correcta. Para el español, obtenemos una mejor clasificación de la etiqueta 0 con una exactitud de 77.53 % y una peor clasificación para la etiqueta 1 con un 57.99 % de exactitud con respecto al inglés.

4.3. Comparación con el estado del arte

En la tabla 5 se muestra una comparación entre el mejor modelo que obtuvimos durante el proceso de experimentación de nuestra propuesta contra los mejores resultados obtenidos de esta competencia [15].

Al darle un enfoque general a los clasificadores, los modelos no logran identificar por completo las características específicas que se pudieron haber obtenido del conjunto de datos. Es decir, al utilizar los hiperparámetros descritos en la tabla 2 no logramos captar estas características que pudieron influir de manera positiva en nuestra fase de experimentación.

5. Conclusiones y trabajo futuro

Los modelos propuestos durante el desarrollo de la investigación, obtuvieron resultados prometedores. A pesar de que el desempeño obtenido por nuestra mejor aproximación no sobrepasa al mejor de los concursantes durante el CLEF 2020, las características que estamos extrayendo son únicamente del texto, además de que a pesar de que se utilizaron los embeddings de palabras, la clasificación no mejora significativamente en comparación con el método TF·IDF.

En español, la identificación de autores que no propagan noticias falsas se hace mejor que en inglés, lo que indica que existen características propias de quienes hacen periodismo de calidad en español que son más fáciles de identificar para estos algoritmos.

Dado que en esta primera iteración de experimentación se obtuvieron resultados prometedores y que existe la posibilidad de mejorar el estado del arte, nuestras ideas para nuestro trabajo futuro son las siguientes: probar con otros métodos de obtención de características, como una bolsa de palabras por conteo, en la que se identifiquen el número de apariciones de ciertos patrones como los URLs, o las menciones de usuarios, utilizar algoritmos de aprendizaje profundo como redes neuronales, o transformadores con modelos de lenguaje a gran escala.

Aumentar el vector de características con el etiquetado gramatical (POS), la complejidad de las oraciones y el tipo de lenguaje utilizado en los textos.

Referencias

1. Bastick, Z.: Would you notice if fake news changed your behavior? An experiment on the unconscious effects of disinformation. Computers in Human Behavior, vol. 116 (2021) doi: 10.1016/j.chb.2020.106633
2. Breiman, L.: Random forests. Machine Learning, vol. 45, pp. 5–32 (2001) doi: 10.1023/A:1010933404324
3. Buda, J., Bolonyai, F.: An ensemble model using N-grams and statistical features to identify fake news spreaders on twitter. In: Conference and Labs of the Evaluation Forum (Working Notes) (2020)
4. Cervero, R., Rosso, P., Pasi, G.: Profiling fake news spreaders: Personality and visual information matter. In: Natural Language Processing and Information Systems: 26th International Conference on Applications of Natural Language to Information Systems, NLDB 2021, pp. 355–363 (2021) doi: 10.1007/978-3-030-80599-9_31
5. Davoudi, M., Moosavi, M. R., Sadreddini, M. H.: DSS: A hybrid deep model for fake news detection using propagation tree and stance network. Expert Systems with Applications, vol. 198 (2022) doi: 10.1016/j.eswa.2022.116635
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation, vol. 9, no. 8, pp. 1735–80 (1997) doi: 10.1162/neco.1997.9.8.1735
7. Honnibal, M., Montani, I., Van Landeghem, S., Boyd, A.: Industrial-strength natural language processing in python: spaCy. (2020)
8. Ireton, C., Posseti, J.: Journalism ”fake news”& disinformation: Handbook for journalism education and training. Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (2018)
9. Kumar, S., Shah, N.: False information on web and social media: A survey. Computer Science, vol. 1, no. 1 (2018)
10. Manna, R., Pascucci, A., Monti, J.: Profiling fake news spreaders through stylometry and lexical features. UniOR NLP@ PAN2020. In: Conference and Labs of the Evaluation Forum (Working Notes) (2020)
11. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A.: Advances in pre-training distributed word representations. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (2018)
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, vol. 12, pp. 2825–2830 (2011)
13. Pizarro, J.: Profiling bots and fake news spreaders at PAN'19 and PAN'20: Bots and gender profiling 2019, profiling fake news spreaders on twitter 2020. In: 2020 IEEE 7th International

Conference on Data Science and Advanced Analytics (DSAA), pp. 626–630 (2020) doi: 10.1109/DSAA49011.2020.00088
14. Pizarro, J.: Using N-grams to detect fake news spreaders on twitter. In: Conference and Labs of the Evaluation Forum (working notes) (2020)
15. Rangel, F., Giachanou, A., Ghanem, B., Rosso, P.: Overview of the 8th author profiling task at PAN 2020: Profiling fake news spreaders on twitter. In: CEUR Workshop Proceedings, vol. 2696, pp. 1–18 (2020)
16. Rangel, F., Rosso, P., Ghanem, B., Giachanou, A.: Profiling fake news spreaders on twitter. Zenodo, (2020) doi: 10.5281/zenodo.4039435
17. Rathod, S.: Exploring author profiling for fake news detection. In: 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 1614–1619 (2022) doi: 10.1109/COMPSAC54236.2022.00256
18. Raza, S., Ding, C.: Fake news detection based on news content and social contexts: A transformer-based approach. International Journal of Data Science and Analytics, vol. 13, no. 4, pp. 335–362 (2022) doi: 10.1007/s41060-021-00302-z
19. Sahoo, S. R., Gupta, B.: Multiple features based approach for automatic fake news detection on social networks using deep learning. Applied Soft Computing, vol. 100 (2021) doi: 10.1016/j.asoc.2020.106983
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
21. Zhang, X., Cao, J., Li, X., Sheng, Q., Zhong, L., Shu, K.: Mining dual emotion for fake news detection. In: Proceedings of the Web Conference 2021, pp. 3465–3476 (2021) doi: 10.1145/3442381.3450004