ISSN 1870-4069

Análisis de texto tóxico en redes sociales

Miguel Soto, Hiram Calvo, Alexander Gelbukh

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

msotoh2021@cic.ipn.mx,
hcalvo@cic.ipn.mx, gelbukh@cic.ipn.mx

Resumen. Las redes sociales se han convertido en una parte importante de nuestra sociedad y, en particular, de la vida de los jóvenes. En muchos casos, las redes sociales se convierten en un lugar para que la gente exprese sus opiniones y comparta sus pensamientos con el mundo. Mientras que los usuarios puedan expresarse libremente, esto también puede conducir a la propagación del lenguaje tóxico; ya que por desgracia algunos usuarios utilizan lenguaje intencionalmente hiriente, despectivo o dañino, y en algunos casos como forma de ciberacoso. Este tipo de comportamiento tóxico es un problema al que hay que hacer frente. Es por esto que en este trabajo proponemos la revisión de distintos modelos computacionales tanto de aprendizaje automático como de aprendizaje profundo, los cuales trabajarán bajo *K-Folds Cross-validation* con las mismas condiciones, con la finalidad de identificar cuál de estos modelos se desempeña de mejor manera para resolver el problema de identificar el texto considerado como tóxico; y en caso de ser identificado de esta manera, definir a qué etiquetas corresponde como parte de la explicación del por qué es tóxico.

Palabras clave: Lenguaje tóxico, redes sociales, ciberacoso, aprendizaje automático, aprendizaje profundo, validación cruzada, clasificación de texto, procesamiento de lenguaje natural.

Toxic Language Analysis on Social Media

Abstract. Social media has become an essential part of modern life, especially among young people, by enabling the free expression of ideas and opinions. However, this openness also facilitates the spread of toxic language, characterized by hurtful, derogatory, or harmful expressions that, in many cases, manifest as cyberbullying. This work proposes the review and comparison of different machine learning and deep learning models, evaluated under a K-Folds cross-validation scheme with homogeneous conditions. The objective is to determine which model achieves the best performance in toxic text detection and, additionally, to classify the text into different toxicity categories (e.g., insults, threats,

discrimination), thus providing a more complete explanation of why a text is considered toxic.

Keywords: Toxic language, social media, cyberbullying, machine learning, deep learning, cross-validation, text classification, natural language processing (NLP).

1. Introducción

En los últimos años se ha visto un crecimiento exponencial respecto al número de personas que utilizan redes sociales, y a pesar de que pueden ser una herramienta muy poderosa para las interacciones humanas virtuales, como conectar personas a largas distancias o inclusive para hacer crecer un negocio, estas traen consigo algunos problemas. Uno de ellos es la mala comunicación con los demás usuarios, esto sucede debido a que las personas que interactúan con los contenidos que ofrecen estos sitios no necesariamente tienen las mismas opiniones cuando algún tema surge en una discusión. Sumándole a esto, el poco control de estas plataformas para erradicar el mal uso, incita a las personas a utilizar lenguaje tóxico. Para este trabajo definiremos el lenguaje tóxico como: "Aquel lenguaje que tiene como intención insultar, amenazar o mostrar desprecio hacia una persona o grupo de personas". El uso de este lenguaje ha demostrado tener un impacto negativo en las personas convirtiéndolo en una de las razones que corrompe la salud mental de los adolescentes [21] y una de las principales razones de suicidio [13]. Por consiguiente, el detectar esta clase de lenguaje es un problema relevante que necesita ser tomado en cuenta.

Existen diferentes enfoques y herramientas disponibles para identificar el lenguaje tóxico u ofensivo, pero se han diseñado con un propósito muy general: estos enfoques proponen la extracción de palabras clave de los textos con contenidos tóxicos con criterios como la frecuencia. En este sentido, esto puede representar un problema debido a que una palabra relevante para un texto que no es ofensivo no debe catalogarse como una palabra clave. Por tal motivo es necesario identificar las palabras que son relevantes en los textos ofensivos y a su vez poco relevantes en los textos no ofensivos.

La razón principal por la que debemos erradicar el lenguaje tóxico de las redes sociales es para evitar que los usuarios que hacen uso de estas plataformas sigan sufriendo, ya que a menudo las personas que utilizan este tipo de lenguaje juzgan a los demás por sus publicaciones en redes sociales, y a partir de dichas publicaciones más usuarios pueden realizar comentarios ofensivos. De esta manera, las personas que ven comentarios ofensivos hacia otra persona pueden sentirse tentados a tomar represalias sobre la gente que esta haciendo daño. Esto genera un círculo vicioso, en el que la gente publica comentarios hirientes sobre otros, lo que provoca mas comentarios y palabras hirientes. Para contrarrestar el lenguaje ofensivo vive en la Internet, algunas redes sociales como Facebook han implementado medidas para detectar texto en imágenes con ayuda de algoritmos tipo OCR (Optical Character Recognition) [4].

Mientras que otros sitios suelen revisar manualmente o tener moderadores que revisen publicaciones y comentarios con la finalidad de eliminar este tipo de comentarios. Sin embargo, este tipo de revisiones manuales requieren de mucho trabajo, por lo que no son sostenibles ni escalables en el tiempo.

Es por esto que en este trabajo proponemos la creación de un modelo que sea capaz de implementar una mejor clasificación entre comentarios tóxicos a través de las múltiples subcategorías que este pueda tener y los comentarios limpios o libres de toxicidad.

2. Estado del arte

Debido a que la detección del lenguaje tóxico u ofensivo es una de las grandes problemáticas que se han buscado erradicar en el campo del procesamiento del lenguaje natural, se han realizado numerosas investigaciones al respecto, comenzando desde lo más básico del lenguaje utilizando simplemente las características léxicas como diccionarios [17] o la bolsa de palabras (BoW) [5]. Sin embargo, y a pesar de que el uso de las características léxicas funcionan, es necesario entender el contexto o la estructura sintáctica del texto que queremos analizar. Es por esto que más adelante se utilizaron enfoques basados en N-gramas donde se presenta un conjunto de datos con las etiquetas de *odio*, *lenguaje ofensivo* o *ninguna*, y se construye un clasificador basado en características con transformación TF-IDF (*Term frequency – Inverse document frequency*) sobre n-gramas, *part of speech*, análisis de sentimientos, entre otras características, donde el mejor modelo presentado por los autores fue entrenado utilizando regresión logística [8].

En 2017 [2], los autores experimentan con múltiples arquitecturas de aprendizaje profundo para la tarea de detección de discursos de odio en Twitter, obteniendo sus mejores puntuaciones F1 utilizando redes de memoria a largo plazo (LSTM) [14] y refuerzo de gradiente [2]. Esto daría pie a más desarrollos de aprendizaje con arquitecturas similares, como Georgakopoulos y colegas optan por comparar las Redes Neuronales Convolucionales (CNN) con el enfoque tradicional de bolsa de palabras [11], combinándolo con una selección de algoritmos como k-Vecinos más Cercanos (kNN), Máquinas de Vectores de Soporte (SVM), Naive Bayes (NB) o Asignación Latente de Dirichlet (LDA) [19] que utilizan una CNN híbrida con la intuición de que la entrada a nivel de caracteres contrarrestaría las palabras mal escritas a propósito o por error y los vocabularios inventados, demostrando que las CNN superan los enfoques tradicionales de la minería de textos en la clasificación de comentarios tóxicos, lo que supondría un gran potencial para el desarrollo de la clasificación de identificación de comentarios tóxicos.

Desde la introducción de algoritmos basado en *transformers*, el uso de los modelos lingüísticos preentrenados en la clasificación de texto se ha convertido en la corriente principal de investigación, ya que el proceso básico consiste en añadir capas específicas para la tarea a realizar, posteriormente añadir el modelo preentrenado y, a continuación, entrenar el nuevo modelo en el

que solo se entrenan las capas específicas de la tarea desde cero. Algunos de los modelos preentrenados más utilizados por la comunidad científica son *Bidirectional Encoder Representations from Transformers* (BERT) [9], *A Robustly Optimized BERT Pretraining Approach* RoBERTa [18], y *Cross-lingual Language Model Pretraining* XLM [16]. El uso de estos modelos se debe a que fueron preentrenados con corpus extremadamente grandes, como en el caso de BERT [9] que fue entrenado con mas de tres mil millones de palabras, y para este tipo de modelos por lo general las capas de salida se sustituyen por las capas específicas de la tarea a realizar y este nuevo modelo se entrena de forma supervisada.

El uso de los modelos preentrenados ha dado pie a la mejora de estos, ya sea añadiendo capas de atención (LSTM) entre la capa de clasificación lineal y el modelo preentrenado [3,7] o explorando el prentrenamiento continuo utilizando corpus del dominio correspondiente y luego transfiriendo el modelo recién afinado a las tareas de clasificación objetivo [12]. De esta manera la literatura nos expande el horizonte para ajustar los modelos preeentrenados a nuestra conveniencia para poder lograr una mejor exactitud a la hora de probar nuestros modelos.

En cuanto a los conjuntos de datos que se han utilizado a lo largo del tiempo, hay que tomar en consideración que la mayoría se han realizado para el idioma inglés. Por mencionar algunos, en 2017 Davidson y sus colegas presentan un conjunto de datos con alrededor de 25,000 tuits anotados con las etiquetas: odio, lenguaje ofensivo o ninguno de los dos [8]. Posteriormente en 2018, Founta y sus colegas anotan un conjunto de datos de 80,000 tuits y los dividen en ocho categorías: ofensivo, abusivo, discurso de odio, agresivo, ciberacoso, spam y normal; su anotación para su uso a gran escala se divide en cuatro categorías que son: abusivo, odioso, normal o spam [10]. OLID (*Offensive Language Identification Dataset*) [22], presentado en 2019, contiene únicamente 14,000 tuits anotados manualmente como ofensivos y no ofensivos. Para 2020, Kurrek y colegas presentan un conjunto de datos el cual fue dividido en múltiples etiquetas, las cuales son: despectivo, apropiado, no despectivo/no apropiado, homónimos y ruido [15].

A pesar de que para el idioma español no exista la misma cantidad conjuntos de datos para la detección de texto ofensivo o tóxico como para el inglés, existen algunos que vale la pena mencionar. El primero de ellos fue presentado en 2018, donde se anotaron 11,000 tuits en español mexicano en 2 etiquetas: agresivo y no agresivo [6]. En 2021, [1] presentan un conjunto de datos con la finalidad de promover la detección del lenguaje ofensivo en las variantes del español, y se encuentra dividido en las siguientes categorías: ofensivo, el objetivo es una persona (OFP), ofensivo, el objetivo es un grupo de personas o un colectivo (OFG), ofensivo, el objetivo es diferente de una persona o un grupo (OFO), no ofensivo, pero con lenguaje impropio (NOE), no ofensivo (NO).

2.1. Diferencia de esta propuesta con respecto al estado del arte

Tras la investigación del estado del arte, proponemos una estrategia distinta para el preprocesamiento del texto, el tratamiento de los datos, así como para el uso de los modelos propuestos que se encuentran descritos en la sección 3. Esto con la intención de observar si hay alguna mejora significativa o inclusive si tienen un peor desempeño los resultados al compararlos contra el estado del arte, y de igual manera para realizar una comparación contra modelos basados en *transformers* dónde se busca obtener resultados similares con mayor rapidez.

3. Desarrollo de la solución

El proceso que se llevará a cabo para implementar la solución al problema que busca resolver este trabajo constará de distintos pasos, los cuales se muestran en el esquema presentado en la figura 1. Cada una de las etapas será descrita a detalle más adelante en esta sección.

3.1. Recolección de datos

Para comenzar la implementación de la solución de este trabajo nos enfocaremos en trabajar el conjunto de datos obtenido por [8], al cual nos referiremos a partir de este momento como HSOL (por sus siglas en Inglés, *Hate Speech and Offensive Language*), con la finalidad de realizar una comparativa contra el estado del arte. HSOL es una recopilación de tuits, el cual fue dividido en 3 categorías por trabajadores de CrowdFlower (después llamado Figure Eight y posteriormente Appen)[1], dichas categorías son: *hate speech*, *offensive language*, *neither*. Posteriormente, con la finalidad de observar que el procedimiento realizado en este trabajo no solo funciona en HSOL, se hará uso del conjunto [15] presentado en el estado del arte, el cual será llamado SLUR para futuras referencias.

3.2. Preprocesamiento

El preprocesamiento del texto para los conjuntos de datos es muy similar ya que el objetivo es que la limpieza del texto sea lo más parecida posible.

Para lograrlo se realizaron los siguientes pasos:

- En caso de existir etiquetas de entidades HTML, serán eliminadas.
- Si el texto contiene *hashtags*, se separará por palabras en mayúsculas, en caso de ser algún acrónimo, este quedará como acrónimo.
- Se eliminarán los nombres de usuario y enlaces a sitios web.
- Para el idioma inglés, se expandirán las contracciones que se encuentren de manera tal que se pueda visualizar de una mejor forma el texto. Ejemplo: *didn't* pasaría a ser *did not*.

[1] https://appen.com/figure-eight-is-now-appen/

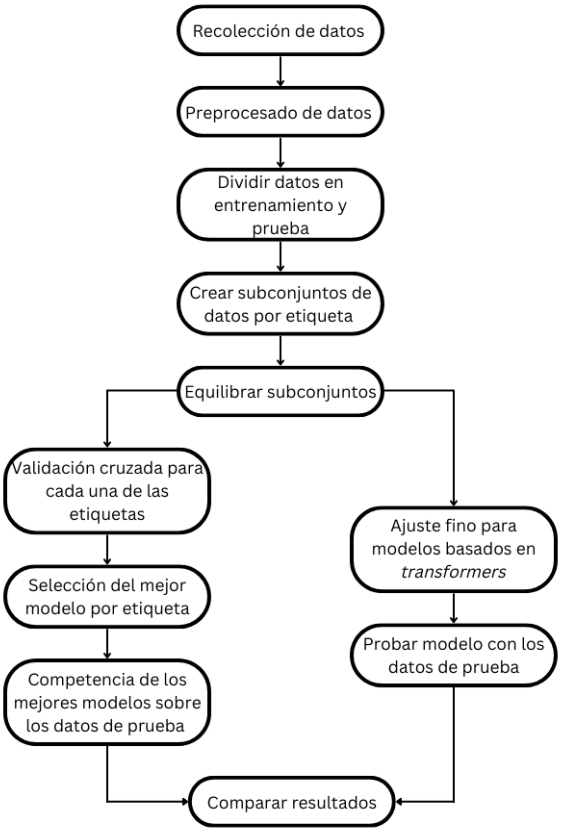

Fig. 1. Esquema planteado para resolver el problema.

- Los signos de puntuación serán removidos.
- Los emojis serán convertidos a texto.
- Se eliminarán las palabras que contengan números. Ejemplo: *y0u*.
- Todo el texto se transformó a minúsculas.

3.3. Partición de datos en entrenamiento y prueba

Como se menciona en la sección 3.1 se busca realizar una comparación contra el estado del arte, por lo cual tomaremos las características que se usaron en el trabajo de [8] para todos los conjuntos de datos. Por lo tanto, la partición para los datos quedaría de la siguiente manera: 90 % para datos de entrenamiento y 10 % para datos de prueba.

Tabla 1. Ejemplo de forma deseada del conjunto de datos.

Texto	Etiqueta 1	Etiqueta 2	...	Etiqueta n
colored contacts in your eyes	0	0	...	0
basic bitch starter pack	0	1	...	1
...	...	...	...	...

Tabla 2. Ejemplo de la forma final de los subconjuntos de datos.

Texto	Etiqueta 1
colored contacts in your eyes	0
basic bitch starter pack	1
...	...

3.4. Crear y equilibrar subconjuntos por etiqueta

En esta sección buscamos en primera instancia que los conjuntos de datos de entrenamiento tengan la misma forma, tal como se muestra en la tabla 1, donde cada etiqueta será marcada con un 1 si corresponde a una etiqueta y se marcará con un 0 en caso contrario. Este proceso se realizará para cada conjunto de datos, ya que estos en su forma original se encuentran con una estructura diferente a lo que buscamos.

Una vez que los conjuntos de datos tienen la misma forma, es hora de dividirlos por etiqueta y los llamaremos subconjuntos de tal manera que nos queden tal y como se muestra en la tabla 2, donde cada una de estas tablas contendrá todos los documentos del conjunto de datos. Posteriormente, identificaremos cuál es la parte de texto limpio, ya sea si es proveniente de alguna etiqueta o si tenemos que realizar algún otro tipo de manipulación de los datos. Cuando tengamos identificados los documentos limpios de los documentos tóxicos, pasaremos a equilibrar cada uno de nuestros subconjuntos con la finalidad de que estos tengan 50 % de documentos limpios y 50 % de documentos pertenecientes a la etiqueta de texto tóxico del subconjunto.

En la tabla 3 se muestran los subconjuntos de datos creados con la cantidad respectiva de documentos limpios, tóxicos y el total de documentos. Como se puede observar ahora todos los subconjuntos se encuentran equilibrados.

3.5. Implementación de los modelos

Aprendizaje automático. Para evaluar los conjuntos de datos utilizaremos algunos de los modelos más utilizados para tareas de clasificación. En este trabajo en concreto se eligieron los siguientes modelos de aprendizaje automático que podemos encontrar en la librería de Python *sklearn* [20]:

Tabla 3. Cantidad de documentos de los subconjuntos equilibrados.

Conjunto	Subconjunto	Limpios	Tóxicos	Total
HSOL	*hate_speech*	1283	1283	2566
	offensive_language	3714	3714	7428
	neither	3714	3714	7428
SLUR	*derogatory*	9793	9793	19586
	homonym	1242	1242	2482
	appropiative	151	151	302
	noise	63	63	126
	non_dorogatory/non_homonym	9793	9793	19586

- *Logistic regression* (LR): este modelo se encuentra definido como *LogisticRegression*() y se basa en un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica en función de variables independientes o predictoras.
- *K-Nearest Neighbors* (KNN): este modelo se encuentra definido como *KNeighborsClassifier*() y es un algoritmo usado como método de clasificación de objetos o elementos que se basa en un entrenamiento mediante ejemplos cercanos en el espacio de los elementos, dónde la función se aproxima solo localmente y todo el cómputo es diferido a la clasificación.
- *Bernoulli Naive Bayes* (BNB): este modelo se encuentra definido como *BernoulliNB*() e implementa los algoritmos de entrenamiento y clasificación de Bayes ingenuo para los datos que se distribuyen de acuerdo con las distribuciones Bernoulli multivariadas; es decir, puede haber múltiples características pero se asume que cada una es una variable de valor binario. Por lo tanto, esta clase requiere que las muestras se representen como vectores de características de valor binario.
- *Multinomial Naive Bayes* (MNB): este modelo se encuentra definido como *MultinomialNB*() e implementa el algoritmo ingenuo de Bayes para datos distribuidos multinomialmente, y es una de las dos variantes clásicas de Bayes ingenuo utilizadas en la clasificación de texto (donde los datos se representan típicamente como recuentos de vectores de palabras, aunque también se sabe que los vectores tf-idf funcionan bien en la práctica).
- *Linear Support Vector Machines* (LSVM): las máquinas de vector soporte (SVM) son un conjunto de métodos de aprendizaje supervisado utilizados para la clasificación, la regresión y la detección de valores atípicos. LSVM se encuentra definido como *LinearSVC*() el cual admite entradas densas y escasas, y el soporte multi-clase se gestiona de acuerdo con un esquema de uno contra el reposo.
- *Random Forest* (RF): Un bosque aleatorio es un meta-estimador que ajusta varios clasificadores de árboles de decisión en varias submuestras del conjunto de datos y utiliza el promedio para mejorar la precisión

predictiva y controlar el sobreajuste. Este modelo se encuentra definido como *RandomForestClassifier*().

Cada uno de los modelos será entrenado con cada uno de los subconjuntos de cada conjunto de datos. Los parámetros que se tomarán en consideración serán: (1) Para convertir el texto a vectores se utilizará el vectorizador TF-IDF considerando unigramas y bi-gramas, y una frecuencia de aparición mínima de 3 veces. (2) Los modelos serán entrenados utilizando la validación cruzada también conocida como *K-Folds cross-validation* con $k = 7$. (3) Para la etapa del ensamble de los modelos, se elegirá el mejor clasificador binario para cada clase utilizando la métrica *precision* como el indicador de mejor desempeño.

Aprendizaje profundo. Por otro lado, realizaremos el ajuste fino de dos modelos pre-entrenados. El ajuste fino esta relacionado con el término de aprendizaje por transferencia, el cual se produce cuando utilizamos el conocimiento que se obtuvo al resolver un problema y lo aplicamos a un problema nuevo pero relacionado. Los modelos pre-entrenados que se utilizarán en este trabajo serán:

- BERT [9]: es un modelo que está diseñado para preentrenar representaciones bidireccionales profundas a partir de texto no etiquetado acondicionando conjuntamente el contexto izquierdo y derecho en todas las capas. Como resultado, se puede ajustar con solo una capa de salida adicional para crear modelos para una amplia gama de tareas, como la respuesta a preguntas y la inferencia del lenguaje, sin modificaciones sustanciales de la arquitectura específicas de la tarea.
- RoBERTa [18]: se basa en la estrategia de enmascaramiento lingüístico de BERT [9], en la que el sistema aprende a predecir secciones de texto ocultas intencionalmente dentro de ejemplos de lenguaje que de otro modo no serían anotados. Modifica los hiperparámetros clave en BERT [9], incluida la eliminación del objetivo de preentrenamiento de la próxima frase de BERT [9] y la formación con minilotes y tasas de aprendizaje mucho más grandes.

Cada uno de estos modelos será entrenado con los conjuntos de datos completos, y los parámetros que se tomarán en consideración serán los siguientes: (1) El tamaño de vocabulario de cada conjunto de datos. (2) El número de épocas serán 5, con una detención anticipada de 2. (3) Habrá 500 pasos de calentamiento o *warmup steps*.

3.6. Ensamble de modelos

El ensamble de modelos solo será aplicado para los modelos de aprendizaje automático y se realizará de la siguiente manera: una vez que todos los modelos hayan sido entrenados con cada uno de los subconjuntos, se elegirá el mejor modelo por cada subconjunto, dicha selección se realizará tomando

en cuenta el mejor desempeño sobre la métrica *precision* de los resultados de la validación cruzada. Una vez los mejores modelos sean elegidos, obtendremos una probabilidad de pertenencia de cada documento de los datos de prueba con cada uno de los modelos. Finalmente realizaremos una competencia de probabilidades de todos los modelos, y la mayor probabilidad será la que obtenga la pertenencia para cada documento.

3.7. Novedad científica

Como se ha descrito a través de la sección 3 se propone una distinta estrategia para el preprocesamiento y equilibrio de los datos, creando así un subconjunto de datos para cada una de las etiquetas. De igual manera, en la subsección 3.6 se describe de una manera más detallada la técnica de ensamble de modelos que se utilizará con los modelos de aprendizaje automático.

3.8. Evaluación

La hipótesis del trabajo se evaluará conforme las métricas que se manejan en la sección 2, siendo estas: $precision = TP/(TP+FP)$, $recall = TP/(TP+FN)$ y $F1\ Score = (2*precision*recall)/(precision+recall)$, donde TP son los verdaderos positivos, FP los falsos positivos y FN los falsos negativos para las predicciones de cada una de los subconjuntos. Una vez los resultados sean obtenidos, se comparará contra el estado del arte y se discutirá si los experimentos realizados en el presente trabajo logra un mejor desempeño, se mantiene en un margen similar o si empeora.

4. Resultados

El objetivo principal de la experimentación es crear un ensamble de modelos a partir de los mejores modelos de aprendizaje automático que se obtengan de la validación cruzada, con dicho ensamble se realizará una competencia con la finalidad de obtener una etiqueta predicha por documento. Posteriormente, los resultados obtenidos serán comparados contra los resultados de los los ajustes finos, y en el caso del conjunto de datos HSOL [8] se comparará contra el estado del arte.

Con la finalidad de que los experimentos sean replicables, las características de los modelos y de la experimentación en general se describen a continuación: (1) Para la partición de datos de entrenamiento y prueba, y los modelos que aceptan un estado aleatorio se asignó la semilla 42. (2) En el caso del modelo KNN, se asignó 2 para el número de vecinos. Para RF se asignó en número de estimadores en 100. Por último, para LR, *LinearSVC*, BNB y MNB no se asignó ningún parámetro adicional.

Esta sección se encuentra distribuida de la siguiente manera: se dividirá en subsecciones para cada conjunto de datos con el que se experimentó, en cada una de estas subsecciones se encuentran los mejores resultados de validación y los resultados sobre los conjuntos de prueba.

Tabla 4. Mejores resultados de validación para los subconjuntos de HSOL [8].

Subconjunto	Modelo	Resultados		
		F1-Score	*Precision*	*Recall*
hate_speech	RF	0.88	0.94	0.84
offensive_language	LR	0.94	0.97	0.91
neither	*LinearSVC*	0.95	0.93	0.96

Tabla 5. Resultados para el conjunto de pruebas de los modelos para HSOL [8].

Modelo	Subconjunto	Resultados		
		F1-Score	*Precision*	*Recall*
Modelo ensamblado	*hate_speech*	0.32	0.31	0.32
	offensive_language	0.92	0.95	0.90
	neither	0.83	0.75	0.92
BERT	*hate_speech*	0.45	0.59	0.37
	offensive_language	0.94	0.92	0.97
	neither	0.88	0.92	0.85
RoBERTa	*hate_speech*	0.39	0.60	0.29
	offensive_language	0.94	0.92	0.97
	neither	0.87	0.92	0.83

4.1. Conjunto de datos HSOL [8]

La tabla 4 muestra el mejor modelo de aprendizaje automático para cada uno de los subconjuntos en la etapa de validación cruzada. Con los resultados obtenidos, nuestro modelo a ensamblar estará compuesto por los modelos RF, LR y *LinearSVC*. Por otro lado, la tabla 5 muestra los resultados obtenidos sobre el conjunto de pruebas para cada una de las etiquetas, en este punto se puede observar que el modelo ensamblado obtiene mejores resultados en la validación.

4.2. Conjunto de datos SLUR [15]

Similar a los resultados presentados en la subsección 4.1, la tabla 6 muestra los resultados obtenidos con los datos de entrenamiento para todos los subconjuntos que nos ofrece este conjunto de datos. En esta ocasión, nuestro modelo a ensamblar consistirá de los siguientes modelos: MNB, BNB, KNN, MNB y LR. Mientras que en la tabla 7 se observan los resultados obtenidos con los conjuntos de pruebas de todos los modelos propuestos.

5. Análisis y discusión

De manera general hay que mencionar que a pesar de que se realizó un equilibrio de los datos al realizar los subconjuntos para el modelo propuesto

Tabla 6. Mejores resultados de validación para los subconjuntos de SLUR [15].

Subconjunto	Modelo	Resultados		
		F1-Score	*Precision*	*Recall*
derogatory	MNB	0.81	0.86	0.78
homonym	BNB	0.97	0.99	0.95
appropiative	KNN	0.63	0.82	0.53
noise	MNB	0.85	0.94	0.79
non_derogatory/non_homonym	LR	0.85	0.86	0.83

Tabla 7. Resultados para el conjunto de pruebas de los modelos para SLUR [15].

Modelo	Subconjunto	Resultados		
		F1-Score	*Precision*	*Recall*
Modelo ensamblado	*derogatory*	0.57	0.94	0.41
	homonyn	0.82	0.83	0.81
	appropiative	0.03	0.02	0.54
	noise	0.12	0.07	0.50
	non_derogatory/non_homonym	0.69	0.62	0.79
BERT	*derogatory*	0.91	0.90	0.91
	homonyn	0.92	0 .91	0.94
	appropiative	0.07	0	0.04
	noise	0.18	0	0.10
	non_derogatory/non_homonym	0.83	0.83	0.83
RoBERTa	*derogatory*	0.91	0.90	0.91
	homonyn	0.92	0.91	0.94
	appropiative	0.07	0	0.04
	noise	0.18	0	0.10
	non_derogatory/non_homonym	0.83	0.83	0.83

en este trabajo, los conjuntos de datos originales muestran un gran desbalance de documentos entre clases, lo cual supone un gran problema debido a que los clasificadores no tienen suficiente información para ajustar sus parámetros, y de esta manera realizar una clasificación adecuada. Dicho fenómeno se puede observar en ambos conjuntos de datos en las tablas 5 y 7.

En cuanto a los resultados obtenidos, para el conjunto de prueba de HSOL [8] de los ajustes finos de BERT [9] y RoBERTa [18] obtienen un resultado muy similar ya que ambos obtienen 0.81 en *precision* de manera global, 0.73 y 0.70 para *recall* y un *F1 Score* de 0.73 y 0.76 respectivamente. Mientras que el modelo propuesto en este trabajo obtuvo una precisión global de 0.67, para *recall* 0.71 y un *F1 Score* de 0.69. Con estos resultados, no se logró obtener una mejora

contra el estado del arte, los cuales demuestran resultados globales de 0.91 para la métrica *precision* y 0.90 para *recall* y *F1 Score*.

Por otra parte, los resultados sobre el conjunto de pruebas del conjunto de datos SLUR [15] obtenemos un resultado global en la métrica *precision* de 0.53 para ambos ajustes finos, 0.56 para *recall* y 0.58 en la métrica *F1 Score*. Y para el modelo propuesto se obtuvo un resultado global de 0.50 en *precision*, 0.61 para *recall* y 0.45 para *F1 Score*.

6. Conclusiones

A pesar de que no se obtuvieron los resultados esperados, el desempeño del ensamble de modelos propuesto no se aleja de los resultados obtenidos en los ajustes finos, ya que pudimos observar que para algunas etiquetas obtuvimos mejores resultados en métricas donde los ajustes finos no lo hicieron. Para esto hay que tener en consideración que los modelos de aprendizaje automático se tomaron en su forma más básica y que aún hay mucho que experimentar con distintos parámetros que dichos modelos puedan tomar. Por estas razones, se seguirá trabajando sobre este modelado mejorando algunos aspectos con la finalidad de competir con modelos basados en *transformers*.

7. Trabajo futuro

A corto plazo, se buscará mejorar los modelos ajustando sus parámetros, utilizando etiquetados gramaticales POST (*part-of-speech tagging*) e identificando las posibles pérdidas de información en los vocabularios. A la par se se buscará implementar una solución para aquellos conjuntos de datos para los que ciertas clasificaciones no se tengan datos lo suficientemente grandes como para realizar una buena clasificación. Más adelante, se probará la implementación del ensamble de modelos propuesto en este trabajo con conjuntos de datos en otros idiomas como el español, así como con conjuntos de datos más grandes.

Referencias

1. Plaza-del Arco, F. M., Montejo-Ráez, A., Ureña-López, L. A., Martín-Valdivia, M.-T.: OffendES: A new corpus in Spanish for offensive language research. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021). pp. 1096–1108. INCOMA Ltd., Held Online (Sep 2021), https://aclanthology.org/2021.ranlp-1.123
2. Badjatiya, P., Gupta, S., Gupta, M., Varma, V.: Deep learning for hate speech detection in tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion. pp. 759–760. WWW '17 Companion, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2017) doi: 10.1145/3041021.3054223 , https://doi.org/10.1145/3041021.3054223

3. Beltagy, I., Cohan, A., Lo, K.: Scibert: Pretrained contextualized embeddings for scientific text. CoRR, vol. abs/1903.10676 (2019)
4. Borisyuk, F., Gordo, A., Sivakumar, V.: Rosetta: Large scale system for text detection and recognition in images. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 71–79. KDD '18, Asssociation for Computing Machinery, New York, NY, USA (2018) doi: 10.1145/3219819.3219861 , https://doi.org/10.1145/3219819.3219861
5. Burnap, P., Williams, M. L.: Us and them: identifying cyber hate on twitter across multiple protected characteristics. EPJ Data Science, vol. 5, no. 1, pp. 11 (2016) doi: 10.1140/epjds/s13688-016-0072-6
6. Carmona, M. A., Guzmán-Falcón, E., Montes, M., Escalante, H. J., Villaseñor-Pineda, L., Reyes-Meza, V., Rico-Sulayes, A.: Overview of MEX-A3T at IberEval 2018: Authorship and aggressiveness analysis in Mexican Spanish tweets. In: Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) (08 2018)
7. Chronopoulou, A., Baziotis, C., Potamianos, A.: An embarrassingly simple approach for transfer learning from pretrained language models. CoRR, vol. abs/1902.10547 (2019)
8. Davidson, T., Warmsley, D., Macy, M. W., Weber, I.: Automated hate speech detection and the problem of offensive language. In: ICWSM (2017)
9. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, vol. abs/1810.04805 (2018)
10. Founta, A., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., Vakali, A., Sirivianos, M., Kourtellis, N.: Large scale crowdsourcing and characterization of twitter abusive behavior. CoRR, vol. abs/1802.00393 (2018)
11. Georgakopoulos, S. V., Tasoulis, S. K., Vrahatis, A. G., Plagianakos, V. P.: Convolutional neural networks for toxic comment classification. In: Proceedings of the 10th Hellenic Conference on Artificial Intelligence. SETN '18, Association for Computing Machinery, New York, NY, USA (2018) doi: 10.1145/3200947.3208069 , https://doi.org/10.1145/3200947.3208069
12. Gururangan, S., Marasovic, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., Smith, N. A.: Don't stop pretraining: Adapt language models to domains and tasks. CoRR, vol. abs/2004.10964 (2020)
13. Hinduja, S., Patchin, J.: Bullying, cyberbullying, and suicide. Archives of suicide research : official journal of the International Academy for Suicide Research, vol. 14, pp. 206–21 (07 2010) doi: 10.1080/13811118.2010.494133
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation, vol. 9, pp. 1735–80 (12 1997) doi: 10.1162/neco.1997.9.8.1735
15. Kurrek, J., Saleem, H. M., Ruths, D.: Towards a comprehensive taxonomy and large-scale annotated corpus for online slur usage. In: Proceedings of the Fourth Workshop on Online Abuse and Harms. pp. 138–149. Association for Computational Linguistics, Online (Nov 2020) doi: 10.18653/v1/2020.alw-1.17 , https://aclanthology.org/2020.alw-1.17
16. Lample, G., Conneau, A.: Cross-lingual language model pretraining. CoRR, vol. abs/1901.07291 (2019)
17. Liu, S., Forss, T.: New classification models for detecting hate and violence web content. In: 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K). vol. 01, pp. 487–495 (2015)

18. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A robustly optimized BERT pretraining approach. CoRR, vol. abs/1907.11692 (2019)
19. Park, J. H., Fung, P.: One-step and two-step classification for abusive language detection on Twitter. CoRR, vol. abs/1706.01206 (2017)
20. 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., Édouard Duchesnay: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, vol. 12, no. 85, pp. 2825–2830 (2011)
21. Suzuki, K., Asaga, R., Sourander, A., Hoven, C., Mandell, D.: Cyberbullying and adolescent mental health. International journal of adolescent medicine and health, vol. 24, pp. 27–35 (08 2012) doi: 10.1515/ijamh.2012.005
22. Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R.: Predicting the type and target of offensive posts in social media. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 1415–1420. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019) doi: 10.18653/v1/N19-1144 , https://aclanthology.org/N19-1144