Predicción automática de cyberbullying en redes sociales

Andres Antonio Campaña Acuña

Universidad Internacional de la Rioja, Logroño (España)

Fecha: 1 de marzo de 2023

Resumen

El cyberbullying es un problema tan grave y complejo como el bullying, puesto que, si bien el primero no es de forma presencial, causa problemas psicológicos, además de que no se termina al finalizar las actividades escolares, sino que, continúa cuando nos conectamos a un medio tecnológico.

Por ello, este trabajo pretende identificar algoritmos de Inteligencia Artificial, que, aplicando técnicas de procesamiento de lenguaje natural, realicen una predicción automática de cyberbullying, aplicando como métricas de selección la exactitud, F1-score y promedio de sensibilidad. Se estimaron dos tipos de modelos: binario (presencia o no de cyberbullying), y multi clase (no cyberbullying, edad, género, religión, etnicidad y otros tipos), tanto para inglés como español (traduciendo la base de datos).

De los resultados más relevantes, se encontró que el modelo BERT tuvo un buen rendimiento para la clasificación multi clase, con un valor de 0,87 en todas las métricas de selección, y para el modelo en español, aplicando BETO, se encontraron también valores altos en las métricas, siendo de 0,85 en todas, validando que es posible la estimación de un modelo en español con una base de datos en inglés, sin sacrificar considerablemente el resultado de las métricas.

Palabras Clave

Cyberbullying

Inteligencia Artificial

Procesamiento de Lenguaje Natural.

I. Introducción

El término cyberbullying, derivado del término bullying o acoso, se acuña a la agresión repetitiva empleando medios electrónicos [1], como lo pueden ser el empleo de celulares o computadores, ingresando a medios de comunicación como mensajería o publicaciones en redes sociales, videojuegos, entre otros. Asimismo, el cyberbullying puede considerarse como un bullying que, si bien es virtual, podría o pasarse a la vida real, o ser tan recurrente que no deje en paz a la víctima, aún sin presencia física [2].

Esta evolución del acoso hace que no solamente sea una preocupación de la víctima al verla, sino que, podría ocurrir en cualquier momento, generando ansiedad y consecuencias más graves como lo son la auto lesión y hasta el suicidio [3].

Una parte de la solución a este problema global sería contar con un detector en las redes sociales o videojuegos, sea en publicaciones o mensajería, puesto que estos son los principales medios por los que se propaga el cyberbullying.

Es por estas razones expuestas que se considera la opción de entrenar un modelo de IA comparando diferentes técnicas empleadas por otros autores para la detección de cyberbullying, así como su tipo, con el fin de encontrar una forma eficiente de su detección automatizada y clasificación.

II. Estado del Arte

Existe una amplia variedad de trabajos que han desarrollado algoritmos en el idioma inglés, principalmente con datasets libres disponibles en Kaggle, que fueron recopilados por Wang y otros [4] y Fatma [5], en donde se clasifica si existe o no cyberbullying en diversos tweets recopilados, así como otras fuentes como Bullying Traces que fue creado por Xu y otros [6], aunque este último requiere de la API de Twitter para reconstruirse. Por otro lado, están los datasets no disponibles de forma libre, que fueron recopilados por sus autores mediante APIs de diversas aplicaciones o técnicas de Web Scraping en páginas como Twitter, Wikipedia, Youtube, AskFm, entre otros como Luna [7] y Botti [8], recogida de chats directos de Whatsapp como Mamani [9] o simulaciones de conversaciones entre estudiantes como Van y otros [10].

A continuación, se definirán los alcances de los diversos autores para esta temática, así como las herramientas que han empleado y sus resultados

APRENDIZAJE AUTOMÁTICO

Random Forest

La técnica del bosque aleatorio o Random Forest (en adelante, RF) desarrollada por Ho [11], es un algoritmo que es frecuentemente aplicado en problemas de clasificación binaria cualitativa.

Este algoritmo busca separar la base de datos de entrenamiento en varias submuestras para entrenar varios árboles de decisión, se asignan pesos basados en sus niveles de error estimados, y se realizan votos para dar una clasificación única.

Existen diversos hiperparámetros por entrenar en este algoritmo, entre los cuales resaltan: el número de árboles por entrenar (n_estimators), cantidad máxima de predictores por árbol (max_features) y el nivel de profundidad del árbol (max_depth).

Support Vector Machine

El algoritmo conocido como máquina de vectores de soporte (en adelante, SVM) no probabilístico desarrollado por Cortes & Vapnik [12] sirve tanto para problemas de regresión como clasificación. Este algoritmo aplica el concepto de hiperplanos para poder separar los datos, y así clasificarlos correctamente.

Aun así, como la separación no siempre es lograble, se aplican márgenes máximos, que corresponden a hiperparámetros por entrenar, esto amplía la banda de separación, para así ser menos estrictos y converger a una solución.

Naive Bayes

Este algoritmo, basado en el cálculo de probabilidades mediante el teorema de Bayes, lleva el término de Naive (en español, ingenuo) porque asume independencia entre las variables explicativas, y pese a que este supuesto puede ser estadísticamente improbable, se reduce considerablemente el coste computacional en el cálculo de probabilidades y en el tiempo de ejecución del algoritmo.

En términos simples, sería eliminar la probabilidad a priori (previa) de ocurrencia de las variables explicativas, que se encuentra en el denominador de la fórmula del teorema de Bayes, y solo mantener el numerador para realizar los cálculos.

Autores como Botti [8], Cuzcano & Ayma [13], Mamani [9], Leon y otros [14] y Rosa y otros [15] aplican estos 3 algoritmos comentados.

Redes neuronales (RN)

Durante sus avances en las redes neuronales, McCulloch & Pitts [16] publicaron sus avances respecto a los funcionamientos de las neuronas y su compatibilidad matemática para la aplicación de operaciones complejas, por lo que estos algoritmos, basados en el funcionamiento de las neuronas del cerebro humano, permiten resolver problemas de clasificación como los modelos estadísticos convencionales, aunque uniendo todas las características.

Estas redes constan principalmente de 3 partes: capa de entrada, capas ocultas y capa de salida. En la capa de entrada se integran los datos de entrada (input), en las capas ocultas se colocan un número de neuronas determinado por el usuario, no existe un número en específico, dado que el rango de neuronas puede ser 10, 50, 100, etc., así como la misma cantidad de capas ocultas, que pueden ser 1, 2, 5, etc., siendo otra forma de agregar al análisis, la función de activación de cada neurona, puesto que la multiplicación del parámetro estimado por el input o neurona anterior se suele manipular para mejorar el entrenamiento. Entre las diversas funciones de activación se encuentran la ReLU (transforma el resultado en 0 si el valor es negativo, o lo mantiene en su mismo valor si es positivo), sigmoide (transforma al resultado entre 0 y 1) y tanh (transforma el resultado entre -1 y 1).

Finalmente, la capa de salida, si bien tiene también una función de activación, realmente depende del problema a analizar, por lo cual no es una complicación, para un problema binario, se puede ejecutar con 1 neurona de salida activada por la función sigmoide o 2 neuronas de salida con la función softmax (estandarizando el valor resultante de las neuronas en una misma capa para que sumen 1). Entre los hiperparámetros por entrenar se encuentran el ratio de aprendizaje (learning_rate) que permite regular la actualización de los parámetros estimados, número de capas, número de neuronas, función de activación (activation_function) y solucionador de gradiente (gradient_solver).

Dependiendo de los problemas específicos, se puede referir a diversas arquitecturas o diseños de redes, con el fin de mejorar el modelamiento de los problemas, por lo que a continuación se conceptualizarán algunas de las arquitecturas más empleadas en el cálculo de modelos con bases de datos textuales.

- RN Completamente conectada

Cuando una red neuronal conecta todas las combinaciones posibles entre capas, es considerada como una red neuronal completamente conectada (fully connected por su traducción del inglés), siendo una arquitectura básica de cálculo, aunque posee la limitante de que, al ser una red sin una causalidad pre definida, así como una gran cantidad de parámetros por estimar, comúnmente no llega a ser la mejor opción para el modelamiento final

- Codificadores-Decodificadores

Los codificadores (encoders en inglés) son arquitecturas que permiten reducir la dimensionalidad de unas características o variables independientes, pero, conteniendo similar información. Si se retorna un output similar al del input, se le denomina a esta segunda parte decodificador (decoder en inglés).

- RN Recurrentes

Es una arquitectura más compleja que permite, con el cálculo de un estado oculto en el tiempo t (h_t), las variables en el tiempo t (x_t) y la función de activación de tangente hiperbólica, la actualización del estado oculto (h_(t+1)) y la predicción en el siguiente período (y_(t+1)). El problema con esta arquitectura es que, al ser compleja y con cada vez menos impacto mientras más períodos exista, el aprendizaje es menor, por lo que la arquitectura LSTM (Long short term memory) reduce este problema de períodos lejanos mediante una puerta de olvido (forget gate), permitiendo conectar mejor el modelo.

Autores como Botti [8], Mamani [9] y Leon y otros [13] aplican el modelo LSTM, mejorando considerablemente sus resultados.

- Fine-tuning/Transfer Learning

Esto se aplica cuando una red neuronal se encuentra entrenada previamente para aplicar una operación específica, como por ejemplo entrenar una red neuronal para aplicar ruido a una imagen, pero, lo adicional de esto es que sirve como un primer filtro o transformador para llegar a un segundo fin, que es el de interés.

En esto se basan el modelo BERT (Bidirectional Encoder Representations from Transformers) creado por Devlin y otros [17], que aplican una primera capa a los datos de entrenamiento, para así calcular los pesos finales que permitirán la estimación del problema a entrenar. Asimismo, existen diversas variantes de este modelo pre entrenado, con diferentes fines, como detectar un idioma específico, este es el caso de BETO (Spanish BERT) creado por Cañete y otros [18], entrenado específicamente para el idioma español.• Fine-tuning model for cyberbullying prediction (Luna [7])

Como este modelo se encuentra disponible y gratuito, y sirve

para predecir la probabilidad de cyberbullying en un texto en español, es relevante probar su efectividad para comparar las herramientas disponibles, con los modelos entrenados en este trabajo. Según la página de Hugging Face, fue entrenado aplicando un modelo basado en RoBERTa (variante de BERT que modifica robustamente hiperparámetros clave), indicando una exactitud de 0,96, convirtiéndolo en un candidato válido para comparativas con este trabajo.

PROCESAMIENTO DE LENGUAJE NATURAL

TF-IDF

La técnica creada por Hans [19] y Spärck [20] conocida como Term Frequency-Inverse Document Frequency (en adelante, TF-IDF), la cual, en vez de asignar valores a las palabras en base a su número de apariciones, este lo hace mediante el valor de su aparición relativa tanto en un documento como en el total de documentos, logrando eliminar o restar la importancia a palabras que figuran en muchos textos a la vez, y enfocarse en los de menor frecuencia. La fórmula es la siguiente:

$$w_{x,y} = tf_{x,y} * \log\left(\frac{N}{df_x}\right)$$

$w_{x,y}$: peso de la palabra x en el documento y

$tf_{x,y}$: frecuencia de la palabra x en el documento y

df_x: número de documentos que contienen la palabra x

N: cuerpo o corpus (número total de documentos)

Como el valor del logaritmo de 1 es 0, entonces si una palabra figura en todos los textos, esta no posee información relevante y se elimina, asimismo, si una palabra no figura en un texto, su peso es 0.

Normalmente esto se combina con la lematización (pasar una palabra a su forma raíz), para reducir la carga del TF-IDF y quedar con características más relevantes.

Entre los autores que han aplicado la técnica del TF-IDF con lematización se encuentran Cuzcano & Ayma [13] y Rosa y otros [15], los cuales aparte aplican diferentes tipos de lematizadores, puesto que esta tarea no es perfecta y depende de la librería empleada.

Word2vec

Se puede aplicar según dos técnicas: Common Bag of Words y Skip-gram, logrando conseguir matemáticamente interpretaciones lingüísticas.

Con estas técnicas que permiten lematizar y contextualizar automáticamente los datos textuales, se logra incluso un mayor alcance pudiendo entrenar modelos a partir de estas características lingüísticas, por lo que modelos pre entrenados como BERT, que aplican el traslado del aprendizaje obtenido en su pre entrenamiento, permite hacer que el modelo aprenda una tarea específica como lo sería, para el objetivo de este trabajo, lograr clasificar entre cyberbullying o no mediante un fine-tune (re estimación de parámetros de la red neuronal).

Un autor que aplicó esta técnica con un resultado favorable fue Botti [8], que aplicó Word Embeddings más LSTM, consiguiendo métricas por encima del 0,8.

III. Objetivos y Metodología

El objetivo general del presente trabajo es identificar las mejores técnicas para la detección de cyberbullying o no, así como de tipos de cyberbullying, con la base de datos disponible en inglés y su traducción al español.

La base de datos disponible para el experimento será la de Kaggle, según lo citado en la sección anterior. Respecto a la preparación de los datos y entorno, se realizó lo siguiente:

- Tanto para el modelo en inglés como español, se empleó la base de datos de Kaggle, donde muestra 6 categorías: no bullying, bullying por edad, bullying por género, bullying por raza, bullying por religión y otros tipos de bullying, siendo aproximadamente 8000 tweets por cada categoría; sin embargo, para simplificar y seguir la comparativa del estado del arte, también se agruparon todas las categorías de bullying para crear un problema binario, dando un desbalanceo con 17% de la clase negativa, y 83% de la clase positiva.

- Para la traducción, se aplicó la librería Deep-translator, la cual posee un módulo de Google traductor, permitiendo una rápida conversión de textos del inglés al español, aunque, es preciso indicar que existe un límite de cantidad de palabras por solicitud, así que se tuvo que eliminar 1 registro debido a la longitud de este.

- Para las técnicas de pre procesamiento, mediante regex expressions, se eliminaron caracteres especiales como las comas, arrobas, puntos, entre otros, se eliminaron caracteres únicos y espacios adicionales, así como la eliminación de stopwords y lematización con su respectivo módulo de spaCy (en_core_web_md para inglés y es_core_news_md para español).

- Dependiendo de si se aplicó o no el paso anterior, se indica el término “base de datos original”, caso contrario, será considerado como “base de datos lematizada”.

- Para la vectorización, se aplicó de la librería sklearn la función CountVectorizer, la cual requiere de hiperparámetros como máxima cantidad de palabras, mínima frecuencia de aparición de palabras y máxima frecuencia de aparición de palabras, las cuales se les asignó los valores de 1000, 5 y 0,7, respectivamente, indicando que como máximo se guardan 1000 palabras en el corpus, la palabra debe aparecer como mínimo 5 veces para ser considerada, y como máximo debe aparecer en el 70% de los textos.

- Posterior a esto, se aplicó la técnica de TF-IDF.

- La separación de la base de datos constó de 80% para entrenamiento y 20% para prueba, con una semilla aleatoria igual a 17.

Para el idioma inglés, con la base de datos lematizada se aplicaron los algoritmos de RF, SVM, Naive Bayes y diversas arquitecturas de redes neuronales como las completamente conectadas, y modelos pre entrenados como BERT, aunque para esta última también se aplicó el entrenamiento con la base de datos original, puesto que al emplear vectores de palabras (Word2vec), se podría estar omitiendo información importante al lematizar.

Para el idioma español se aplicaría el modelo de Luna [7] tanto en la base de datos original como lematizada, para observar su rendimiento, así como los mismos experimentos de la base de datos en inglés, con la excepción que se aplicaría BETO (variante de BERT para el idioma español).

Estos experimentos serían aplicados tanto para el modelo binario como multi clase, exceptuando el de Hugging Face,

puesto que solo retorna la categoría de presencia o no de cyberbullying, por lo que no se le puede aplicar el modelo multi clase.

IV. Contribución

La primera contribución de este trabajo radica en que se estimen modelos del estado del arte respecto a la temática de predicción automática de intención de cyberbullying, y su sub tipo, puesto que esto apoyaría a detectar el problema a tiempo, manteniendo mayor tranquilidad por parte de los usuarios susceptibles a estos ataques, y logrando evitar daños físicos y psicológicos posibles.

Además de los modelos más frecuentes del estado del arte, se aplicaría una variante más compleja que, si bien es más complicado de entrenar, potencialmente aporta a un mejor resultado, lo que hace un ejercicio de experimentación importante.

La segunda contribución de este trabajo radica en experimentar estimando el modelo no solo en inglés, dado que es la única base de datos libre en internet, sino, demostrar que, dado los avances actuales en la IA, es posible su traducción rápida y gratuita al idioma español, para estimar un modelo específico para este idioma.

De mostrar resultados satisfactorios, sería factible el entrenamiento de modelos con datos traducidos, rompiendo las restricciones en idioma que presentan las bases de datos textuales, permitiendo que proyectos de IA sean más veloces en realizarse, al no tener que recopilar propios datos cuando ya existe una base para estos, más baratos, al no gastar en el etiquetado de la base de datos por expertos, y más efectivo, puesto que algunas bases ya fueron etiquetadas por expertos, transfiriendo su conocimiento a un diferente idioma.

V. Evaluación y Resultados

Los experimentos realizados han consistido en estimar los modelos y evaluar sus métricas de exactitud, F1-score y sensibilidad y especificidad (para el modelo binario) o promedio de sensibilidad (para el modelo multi clase), tanto para la base de datos en inglés como la traducida al español.

Evaluación 1: modelo binario

Los resultados para la base de datos, en donde se lematizó la base de datos para los algoritmos de RF, SVM, Naive Bayes, Redes Neuronales (completamente conectadas), así como un modelo de BERT, para inglés, y BETO, para español, con la base de datos original y uno lematizada, han sido los mostrados en la Tabla I.

Tabla I

Resultados para modelo binario (inglés)

Modelo	Exactitud	Especificidad	Sensibilidad	F1-score
TF-IDF + RF	0.87	0.5	0.95	0.86
TF-IDF + SVM	0.87	0.4	0.97	0.85
TF-IDF + Naive Bayes	0.64	0.87	0.59	0.68
TF-IDF + Redes Neuronales	0.84	0.68	0.88	0.85
TF-IDF + Redes Neuronales (ratio de aprendizaje=0.5, solucionador="SGD", capas ocultas=(150,100,50,10))	0.86	0.57	0.92	0.86
BERT (original)	0.88	0.69	0.92	0.88
BERT (lematizado)	0.88	0.54	0.96	0.88

En la Tabla I se observa que de los primeros 4 modelos, el que presentó las mejores métricas en términos de exactitud, F1-score y especificidad fue el de redes neuronales, puesto que su especificidad no fue tan baja como la del RF y SVM, por lo que su calibración fue de un ratio de aprendizaje de 0,5, un solucionador SGD (sthocastic gradient descent) y 4 capas ocultas con 150, 100, 50 y 10 neuronas, respectivamente, mejorando la exactitud y F1-score, aunque disminuyendo especificidad. Respecto a los modelos tipo BERT, se encontró que el entrenado con la base de datos original es mejor globalmente, tanto con los primeros modelos mencionados, como con el BERT con la base de datos lematizada.

Los resultados para la base de datos en español como problema binario se muestran en la Tabla II.

Tabla II

Resultados para modelo binario (español)

Modelo	Exactitud	Especificidad	Sensibilidad	F1-score
Hugging Face (normal)	0.51	0.94	0.41	0.55
Hugging Face (lematizado)	0.52	0.89	0.44	0.57
TF-IDF + RF	0.87	0.46	0.95	0.86
TF-IDF + SVM	0.87	0.37	0.97	0.85
TF-IDF + Naive Bayes	0.56	0.91	0.48	0.6
TF-IDF + Redes Neuronales	0.84	0.56	0.9	0.84
TF-IDF + Redes Neuronales (ratio de aprendizaje=0.3, solucionador="SGD", capas ocultas=(100,50,10))	0.85	0.61	0.9	0.85
BETO (original)	0.88	0.59	0.94	0.88
BETO (lematizada)	0.87	0.49	0.95	0.86

Se observa que el modelo encontrado en Hugging Face, si bien muestra una muy alta especificidad, da malos resultados en exactitud y F1-score, convirtiéndolo en un candidato con bajo rendimiento. Por el lado de los modelos convencionales, el que mostró el mejor rendimiento también fue el de redes neuronales, calibrándolo con un ratio de aprendizaje de 0,3, solucionador SGD y capas ocultas de 100, 50 y 10 neuronas, respectivamente. Por el lado de los modelos más avanzados, BETO con la base de datos original resultó ser el mejor modelo, como en la base de datos en inglés, con un rendimiento de 0,88 en exactitud y F1-score, pero, se mantiene el mal rendimiento en la especificidad, por lo que no se puede afirmar que se encontró un modelo que mejore el estado del arte para esta especificación.

Evaluación 2: modelo multi clase

Los resultados para la base de datos en inglés, en donde se lematizó la base de datos para los algoritmos de RF, SVM, Naive Bayes, Redes Neuronales (completamente conectadas), así como un modelo de BERT con la base de datos original y uno lematizada, han sido los mostrados en la Tabla III.

Tabla III

Resultados para modelo multi clase (inglés)

Modelo	Exactitud	Sensibilidad	F1-score
TF-IDF + RF	0.84	0.84	0.84

TF-IDF + SVM	0.84	0.84	0.84
TF-IDF + Naive Bayes	0.74	0.74	0.72
TF-IDF + Redes Neuronales	0.82	0.82	0.82
TF-IDF + RF (# estimadores=, máx Depth, máx features=)	0.84	0.84	0.84
BERT (original)	0.87	0.87	0.87
BERT (lematizado)	0.86	0.86	0.86

Para este modelo se encontró que, de los primeros 4 modelos, se observa la simetría similar entre las 3 métricas, mostrando que el mejor de estos fue el RF y el SVM; sin embargo, dada las facilidades en calibración, se escogió el RF como modelo ganador, y de su calibración, se encontró que la mejor combinación fue con un número de estimadores de 1500, max depth de None y max features de "sqrt". Finalmente, así como en el modelo binario, el modelo de BERT con base de datos original presentó las mejores métricas, pero, ahora sí supera lo observado en el estado del arte, siendo 0,87 en las 3 métricas planteadas.

Para mostrar la clasificación por cada tipo, se muestra la matriz de confusión del modelo ganador en la figura 1,

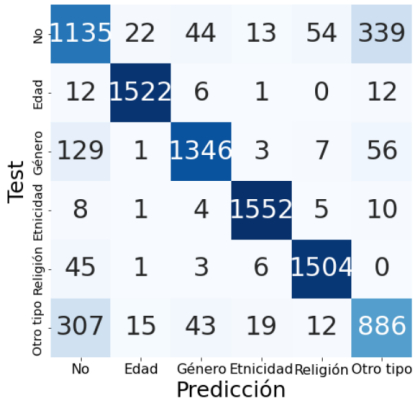

Fig. 1. Resultado de matriz de confusión para BERT, con base de datos original en inglés (todas las categorías)

Se observa que el menor rendimiento se encuentra en la categoría "No" y "Otro tipo", puesto que la mayor cantidad de errores se encuentran cuando el modelo predice "Otro tipo" cuando era "No" en 339 textos, y en viceversa, 307 textos.

TABLA IV

RESULTADOS PARA MODELO MULTI CLASE (ESPAÑOL)

Modelo	Exactitud	Sensibilidad	F1-score
TF-IDF + RF	0.81	0.81	0.81
TF-IDF + SVM	0.81	0.81	0.82
TF-IDF + Naive Bayes	0.73	0.73	0.71
TF-IDF + Redes Neuronales	0.79	0.79	0.79
TF-IDF + RF (# estimadores=, máx Depth, máx features=)	0.81	0.81	0.82
BETO (original)	0.85	0.85	0.85
BETO (lematizado)	0.83	0.83	0.83

Similarmente como en el idioma inglés, para el idioma español se encontraron buenos resultados, siendo ligeramente más bajos, con un valor de 0.85 en todas las métricas, aun así, alcanzando las metas impuestas según el estado del arte, por lo que se puede considerar satisfactorios estos resultados. Además, estos resultados son similares a los del inglés, solo siendo 0,02 menos en todas las métricas.

Para mostrar la clasificación por cada tipo, se muestra la matriz de confusión del modelo ganador en la figura 2.

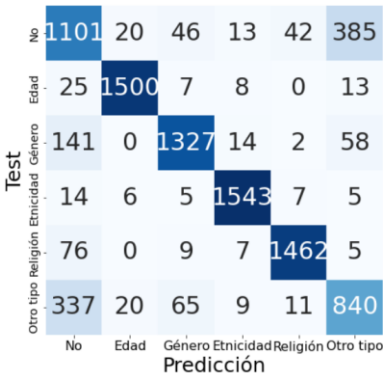

Fig. 2. Resultado de matriz de confusión para BETO, con base de datos original en español (todas las categorías)

De la misma forma, para el modelo es español existe una mayor clasificación equivocada entre "No" y "Otro tipo", se encuentra que cuando el modelo predice "Otro tipo" era "No" en 385 textos, y en viceversa, 337 textos.

VI. DISCUSIÓN

Para el modelo de clasificación binaria, se encontró para el idioma inglés que los modelos entrenados desde un inicio que son RF, SVM, Naive Bayes y Redes Neuronales, si bien mostraron un alto rendimiento en exactitud y F1-score, con resultados mayores a 0,8 para RF, SVM y Redes Neuronales, sus indicadores respecto a la clase negativa (no cyberbullying) salieron considerablemente bajos, teniendo desde un 0,4 hasta un 0,68 en especificidad y desde un 0,59 hasta un 0,75 en VPN. Por el lado del modelo BERT, se encontró una mejora considerable en la exactitud y F1-score, consiguiendo hasta un 0.88 para la base de datos original. Es relevante indicar que, si se hubiese empleado la sensibilidad en este caso, se hubiese concluido que es un modelo muy bueno por poseer valores superiores a 0,9; sin embargo, se encontró una especificidad de 0,69 y un VPN de 0,66, lo cual sigue siendo relativamente bajo, sin superar las métricas impuestas, aun aplicando la lematización y el TF-IDF como Leon y otros [14] para modelos de Machine Learning o aplicando modelos más avanzados como BERT, los cuales están inspirados en LSTM, como los modelos aplicados por Mamani [9].

En lo que respecta a la clasificación multi clase, se encontró que normalmente existían predicciones equivocadas entre las etiquetas "No" y "Otro tipo", por lo que se puede entender que estas dos clases son bastante similares, obteniendo en los modelos entrenados desde un inicio que el mejor modelo fue un RF con una exactitud, F1-score y promedio de sensibilidad de 0,81. Asimismo, para la estimación mediante BERT con la base de datos original, se consiguió para todas las métricas de validación un valor de 0,87, consiguiendo valores ligeramente más altos que los conseguidos por Van y otros [10] para el F1-score y sensibilidad, así como valores más altos en todas las métricas respecto a los alcances de Botti [8], aunque siendo conscientes de que realiza un menor rendimiento en las categorías "No" y "Otro tipo".

Respecto a los modelos estimados con la base traducida en el problema binario, el modelo de Hugging Face (Luna [7]) mostró resultados considerablemente bajos, con la base original y lematizada, por lo que no se puede concluir que esta versión del modelo lograría clasificar correctamente la intención o no de

cyberbullying. Por otro lado, de los modelos estimados, se han encontrado resultados bastante similares a los de la base en inglés, aunque ligeramente menores en las métricas, encontrando que el mejor modelo entrenado desde un inicio obtuvo una exactitud y F1-score de un valor de 0,85, pero, una especificidad de 0,59, y su modelo BETO tampoco mostró mayores mejoras, resultando en una exactitud de 0,88, pero, una especificidad con el mismo valor, por lo que el modelo de clasificación binaria en español de Mamani [9] consiguió un mejor resultado.

Sin embargo, para el problema multi clase, así como para la base en inglés, se consiguieron buenas métricas en general, aunque también con el inconveniente del conflicto entre la categoría "No" y "Otro tipo". El mejor resultado de los modelos entrenados desde un inicio que fue el RF, obtuvo una exactitud, F1-score y promedio de sensibilidad de 0,81; mientras que, para BETO con la base de datos original se consiguió 0,85 en todas las métricas, alcanzando los resultados del estado del arte impuesto, superando el resultado de Cuzcano & Ayma [13] en clasificación multi clase en español, y lo obtenido por Botti [8] para multi clase en inglés.

VII. Conclusiones

Se concluye que la predicción de cyberbullying en mensajes ha sido posible, mostrando resultados no muy favorables para el modelo como problema binario, pero, sí favorables para el problema multi clase, por lo que este último sería el que se validaría, tanto para el idioma español como para el inglés, llevando solo 0,03 puntos de diferencia en todas las métricas para su mejor especificación, que fue BERT para la base en inglés y BETO para la base en español, ambos estimados con la base de datos original (sin lematizar).

La no lematización de la base de datos se puede justificar debido a que estos modelos pre entrenados vienen pre determinados con la aplicación de un Word Embedding, por lo que pasar un vector lematizado disminuye la cantidad de información relevante que puede recoger el modelo.

Asimismo, se encontró que mientras más épocas, el sobreajuste se hacía mayor, por lo que 3 épocas resultaron ser la mejor opción de estimación, además, como la sugerencia de la documentación es aplicar los parámetros predeterminados, entonces no se modificaron otros términos como el ratio de aprendizaje.

Otro detalle relevante encontrado es que las categorías estimadas presentan problemas de diferenciar entre la categoría "No" y "Otro tipo", por lo que se puede especular que es un problema de datos que podría resolverse incrementando la cantidad de textos para estas categorías, con el fin de evitar contradicciones entre estas categorías, esto porque la cantidad de temáticas de "No cyberbullying" es amplia, puesto que se puede hablar de temas como finanzas, salud, entretenimiento, etc.

Finalmente, fue factible la traducción de la base de datos para ofrecer un modelo tanto en inglés como en español, puesto que, si bien se obtuvieron resultados ligeramente inferiores, siguen siendo valores altos. Asimismo, hay que considerar que la ganancia de este hallazgo es la reducción del costo de recolección de información y etiquetado, lo cual es una limitante para los proyectos de IA que requieren de bases de datos textuales, y también, puede aportar en la velocidad (al eliminar el tiempo de recolección) y eficiencia (al poseer etiquetas validadas por profesionales) en el despliegue de proyectos.

Como líneas de trabajo futuro, se sugerirían la aplicación de más variantes de los modelos pre entrenados BERT/BETO, porque no se pudieron realizar más pruebas debido a que cada modelo estimado mediante estas técnicas demoraba casi 1 hora, por lo que con mejores especificaciones técnicas del computador y tiempo disponible aportaría en mejorar más las métricas obtenidas.

Por otro lado, sí se consideraría importante agregar más textos referentes a no cyberbullying y otros tipos, puesto que estas fueron las categorías menos reconocidas, y potencialmente la razón de esto es que estas categorías son parecidas, puesto que hay veces en que el doble sentido es una línea ligera que podría interpretarse o no como una ofensa.

Asimismo, sería interesante poder calibrar los hiperparámetros de los modelos pre entrenados, como el ratio de aprendizaje, el optimizador, número de batches, más o menos capas intermedias, y otros aspectos que podrían afectar.

Finalmente, se consideraría un ejercicio interesante aplicar las técnicas de aumento sintético de datos, como la técnica de SMOTE (Synthetic minority over-sampling tecnique) probada por Rupapara y otros [21] para incrementar hasta en 0.07 las métricas para clasificación de textos con comentarios negativos y positivos. Esta técnica incrementa la cantidad de datos de la categoría con menos valores, eliminando el imbalanceo aunque promoviendo el sobre ajuste, aun así, sus resultados son favorables, por lo que sería un ejercicio interesante aplicarlo al problema binario e inclusive al multi clase. Esta herramienta no se pudo aplicar en este trabajo por la complicación de la programación en datos textuales con redes neuronales, así como el límite de tiempo.

Referencias

[1] Aquino, R. (2014). Cyberbullying: acoso utilizando medios electrónicos. *Revista Digital Universitaria, 15*(1). https://doi.org/http://www.revista.unam.mx/vol.15/num1/art04

[2] Fernández, A. (2015). Bullying y Cyberbullying: prevalencia en adolescentes y jóvenes de Cantabria. Universidad del País Vasco.

[3] Ochoa, L. (2013). *Consecuencias del Bullying en las adolescentes.* Universidad Autónoma del Estado de México.

[4] Wang, J., Fu, K., & Lu, C. (2020). SOSNet: A Graph Convolutional Network Approach to Fine-Grained Cyberbullying Detection. *Proceedings of the 2020 IEEE International Conference on Big Data*.

[5] Fatma, E. (2020). Cyberbullying datasets. *Mendeley Data V1*. https://doi.org/10.17632/jf4pzyvnpj

[6] Xu, J., Jun, K., Zhu, X., & Bellmore, A. (2012). Learning from Bullying Traces in Social Media. 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Lenguage Technologies, pages 656-666, 656-666.

[7] Luna, J. (s.f.). *Hugging Face*. https://huggingface.co/JonatanGk/roberta-base-bne-finetuned-cyberbullying-spanish

[8] Botti, V. (2020). *Detección de Cyberbullying en Redes Sociales.* Universitat Politècnica de Vàlencia.

[9] Mamani, M. (2020). Modelo Basado en Inteligencia Artificial para la detección de ciberacoso. Universidad Mayor de San Andrés.

[10] Van, C., Jacobs, G., Emmery, C., Desmet, B., Lefever, E., Verhoeven, G., . . . Hoste, V. (2018). Automatic Detection of Cyberbullying in Social Media Text.

[11] Ho, T. (1995). Random decision forests. *Proceedings of 3rd international conference on document analysis and recognition, 1*, 278-282. https://doi.org/10.1109/ICDAR.1995.598994

[12] Cortes, C., & Vapnik, V. (1995). Support-Vector networks. *Machine Learning, 20*, 273-297. https://doi.org/https://link.springer.com/article/10.1007/BF00994018

[13] Cuzcano, X., & Ayma, V. (2020). A Comparison of Classification Models to Detect. *International Journal of Advanced Computer Science and Applications, 11*(10), 132-138. https://doi.org/https://hdl.handle.net/20.500.12724/12718

[14] Leon, G., Gallegos, P., Vintimilla, P., Bravo, J., Barbosa, L., Palomeque, W., & Paredes, M. (Octubre de 2019). Presumptive Detection of Cyberbullying on Twitter through Natural Language Processing and Machine Learning in the Spanish Language. CHILECON.

[15] Rosa, H., Pereira, N., Ferreira, P., Carvalho, P., Oliveira, S., Coheur, L., . . . Trancoso, L. (2019). Automatic cyberbullying detection: A systematic review. *Computers in human behavior*. https://doi.org/10.1016/j.chb.2018.12.021

[16] McCulloch, W., & Pitts, W. (1990). A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY. *Bulletin of Mathematical Biology*, *52*(1/2), 99-115. https://doi.org/https://doi.org/10.1007/BF02478259

[17] Devlin, J., Chang, M., Lee, K., & Tooutanova, K. (2019). BERT: Pre-trained of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT 2019. https://aclanthology.org/N19-1423.pdf

[18] Cañete, J., Chaperon, G., Fuentes, R., Ho, J., Kang, H., & Pérez, J. (2020). SPANISH PRE-TRAINED BERT MODEL. PML4DC at ICLR 2020. https://doi.org/https://users.dcc.uchile.cl/~jperez/papers/pml4dc2020.pdf

[19] Hans, L. (1957). A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of research and development*. https://doi.org/10.1147/rd.14.0309

[20] Spärck, K. (1972). A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL. *Journal of Documentation*, *28*(1), 11-21. https://doi.org/10.1108/eb026526

[21] Rupapara, V., Rustam, F., Shahzad, H., Mehmood, A., Ashraf, I., & Sang, A. (2016). Vaibhav, R.; Furqan, R.; Hina, F.; Imran, A.; And, S. *IEEE Access*, *4*. https://doi.org/10.1109/ACCESS.2017.DOI