Integrantes

Nicolas Bedoya Figueroa

Daniel Escalante Pérez

Marilyn Stephany Joven Fonseca

Eder Leandro Carbonero Baquero

Bogotá DC

Marzo 2025

Proyecto de curso

Formato de Propuesta

Índice:

1. Título de la propuesta

Comparación de modelos de machine learning y deep learning al moderar mensajes inapropiados en redes sociales.

2. Integrantes

Nicolas Bedoya Figueroa

Daniel Escalante Perez

Marilyn Stephany Joven Fonseca

Eder Leandro Carbonero Baquero

3. Contexto, problema y Justificación. Impacto esperado

Contextualización detallada y clara del problema que se necesita resolver, especificando el área o sector en el que se enmarca (salud, educación, ambiente, etc.) y su relevancia dentro de ese contexto. Indicar el objetivo principal que se debe alcanzar.

Durante el último par de décadas, con el avance de la tecnología y el internet, las redes sociales han tomado cada vez más relevancia, puesto que estos medios han permitido la divulgación de información de forma más fácil, la democratización de esta, además de una conectividad global. (Amina Saleh Omar, 2024).

En el año del 2024, este crecimiento continuó. Facebook lideró el número de usuarios activos mensuales, llegando a acumular más de 3 mil millones de usuarios, algo que ninguna otra red social ha logrado hasta la fecha. YouTube, por otro lado, tiene 2 mil 500 millones de usuarios activos, mientras que Instagram y WhatsApp se ubican en el tercer puesto con 2 mil millones de usuarios. Lo anterior también genera que las redes sociales tengan una alta cantidad de ingresos, por ejemplo, Facebook genera más de 80 mil millones de dólares anualmente. De lo anterior es posible observar la relevancia que las plataformas de redes sociales poseen, estas tienen un gran número de usuarios, lo que a su vez contribuye a una alta generación de ingresos. (Geuens, 2025)

No obstante, las redes sociales también han traído nuevos problemas a considerar. Con la posibilidad de realizar publicaciones en cualquier momento sobre casi cualquier tema, el discurso de odio por parte de usuarios hacia otras personas ha tomado relevancia. El discurso de odio ha empezado a tener un efecto en la población, en enero del 2023 se realizaron ataques a los edificios del gobierno de Brasil, mientras que el 6 de enero del 2021 se llevó a cabo el asalto al capitolio de los Estados Unidos. Ambos eventos ocurrieron después de que ciertos grupos dirigieran una "retórica peligrosa y afirmaciones falsas contra otros" (United Nations, 2023).

Ahora, analizando a profundidad esta problemática en Facebook e Instagram, se encuentran las siguientes cifras:

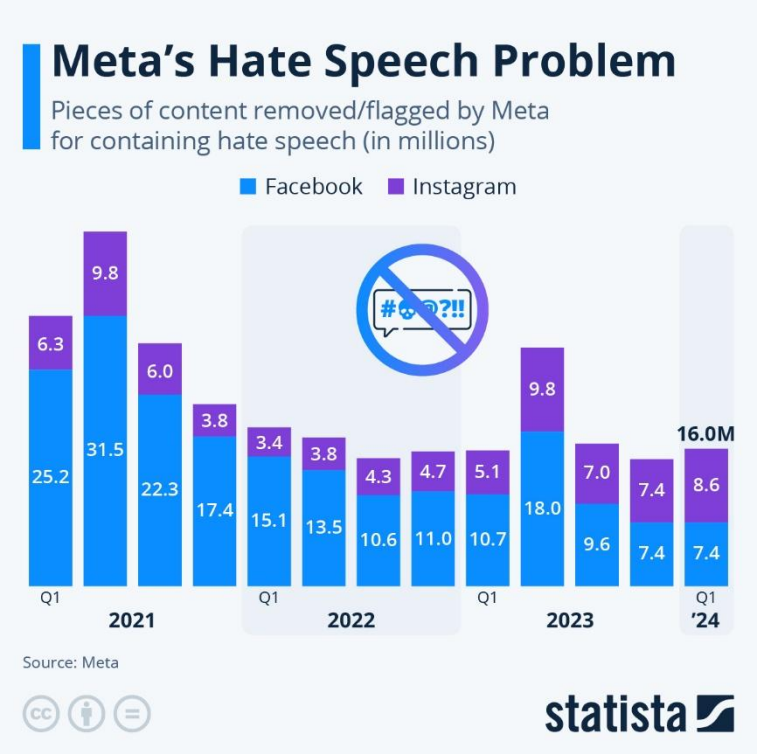


Imagen 1: Meta´s Hate Speech Problem. Tomado de: (Zandt, 2024)

Como se puede observar, cada año se eliminan millones de piezas de contenido ya que estas contienen discurso de odio de alguna forma.

Adicionalmente, en la plataforma X (previamente Twitter) se ha observado también un incremento de este tipo de comentarios. El discurso de odio ha aumentado en general en un 50%, comentarios transfóbicos, homofóbicos y racistas han aumentado en un 260%, 30% y 42% respectivamente. (Cohen, 2025).

Por lo anterior, es posible evidenciar la relevancia que tiene el discurso de odio en el ámbito social, específicamente en las redes sociales. Poder identificar de forma efectiva este tipo de comentarios para que sean eliminados es una tarea fundamental que las redes sociales tienen que llevar a cabo.

Así, lo que se buscará con este proyecto es desarrollar modelos de Machine Learning y Deep Learning para clasificar de forma efectiva comentarios en inglés, puesto que este es uno de los más usados, por ejemplo, para X, el 55% de tuits publicados en este lenguaje (Semiocast, 2024). Se buscará comparar los modelos implementados y observar cuál presenta los mejores resultados.

4. Descripción de los datos disponibles, sus restricciones y limitaciones (si las hay), así como la forma como se podrán acceder (fuente de los datos). (1 página)

Proporcionar una descripción detallada y clara sobre los datos que estarán disponibles para los estudiantes, incluyendo el formato (como bases de datos, archivos CSV, imágenes, texto, etc). Indicar si los datos contienen información sensible o confidencial. Explicar cómo los estudiantes podrán acceder a los datos y si existen restricciones adicionales para su uso, como la necesidad de firmar acuerdos de confidencialidad o solicitar permisos especiales.

Los datos son un factor de gran importancia en el desarrollo de este tipo de modelos, pues en muchos casos se puede enfrentar a la sensibilidad de estos, junto con las restricciones de su uso, que se deben tener en cuenta, especialmente porque la elección de estos puede afectar los resultados de los diferentes modelos. Luego de revisar diferentes estudios y modelos de detección de discurso de odio, se identificó que, aunque muchas redes sociales permiten la publicación de comentarios y texto en general, Twitter destaca como una de las plataformas más

relevantes para este tipo de análisis. Su principal propósito es fomentar la comunicación mediante textos cortos, lo que lo convierte en la fuente principal para este tipo de investigaciones.

Los datos disponibles para el desarrollo del modelo consisten en información basada en texto, que se ha organizado en csvs, relacionada con la detección de lenguaje ofensivo, comentarios tóxicos y discurso de odio. El acceso a estos datos se considera sensible por contener datos de usuarios en algunos casos o puede ser restringido debido a que es contenido que se quiere evitar. En investigaciones previas se han utilizado conjuntos de datos de acceso público que han sido recopilados y etiquetados manualmente para facilitar su uso en modelos de aprendizaje automático.

Uno de los conjuntos de datos más referenciados en estudios previos es el Hate Speech and Offensive Language Dataset de Davidson et al. (2017), el cual recopila tweets clasificados en tres categorías: *discurso de odio*, *lenguaje ofensivo pero no discurso de odio* y *contenido no ofensivo*. En otra investigación, se utilizó un conjunto de datos consolidado de nueve fuentes públicas de Twitter, compiladas a partir de múltiples estudios sobre detección de contenido tóxico en redes sociales. Estas fuentes incluyen el *Hate Speech Dataset*, *Offensive Content Dataset*, y colecciones de tweets etiquetados en categorías como sexismo y racismo. En total, el conjunto de datos final contenía 111,131 tweets en inglés, de los cuales 63,823 fueron etiquetados como *neutral* y 47,308 como *tóxico*.

Dado que estos datos provienen de fuentes públicas, se encontraron los siguientes conjuntos de datos ya utilizados en este tipo de estidos. A continuación, se listan algunos de los repositorios con los datos considerados:

- **Davidson et al.'s Hate Speech and Offensive Language Dataset**: Un dataset ampliamente utilizado que contiene tweets clasificados como discurso de odio, lenguaje ofensivo o ninguno. (Davidson et al., 2017).
- **NLP CSS 2017 Dataset**: Utilizado en aplicaciones de procesamiento del lenguaje natural (NLP) para la clasificación de contenido ofensivo. (Jha, 2017).

- **Zeerak Talat's Hate Speech Dataset**: Un repositorio que recopila múltiples conjuntos de datos relacionados con el discurso de odio. (Talat, n.d.).
- **HASOC Dataset**: Conjunto de datos desarrollado para el desafío de identificación de contenido ofensivo y discurso de odio. (HASOC, 2019).
- **Hate Speech ICWSM 2018 Dataset**: Utilizado en investigaciones sobre detección de discurso de odio en redes sociales. (Elsherief, 2018).
- **Hate Speech and Offensive Language Dataset (Kaggle)**: Un extenso dataset disponible en Kaggle para la clasificación de lenguaje ofensivo. (Morjaria, n.d.).

Por último, como todos los repositorios contienen datasets disponibles para el público, no se consideró necesaria la elaboración de algun acuerdo de confidencialidad o permisos especiales.

5. Criterios de éxito y rendimientos de referencia (si los hay).

Se establecen los siguientes criterios de clasificación que deben ser comparados contra modelos existentes provenientes de estudios previos.

1. **Determinar lo siguiente** en la clasificación de mensajes inapropiados y comparar contra los estudios de referencia resaltados (Al hacer uso de diferentes modelos).

   Accurancy: Su valor nominal debe estar entre: 92.5% <= x <= 97%

   Precision: Su valor nominal debe estar entre: 85% <= x <= 97%

   Recall: Su valor nominal debe estar entre: 92.5% <= x <= 90%

   F1-score: Su valor nominal debe estar entre: 88% <= x <= 93%
2. **Reducción de falsos positivos y falsos negativos** a menos del **5%**.
3. **Mejora en el manejo de mensajes complejos** (sarcasmo, ironía y lenguaje ambiguo).
4. Clasificación del mejor modelo de machine learning en la clasificación de mensajes inapropiados (De acuerdo con estudios similares Random Forest suele presentarse como el mejor modelo de clasificación)
5. Determinar si los datos encontrados son congruentes con estudios previos y que diferencias o similitudes podemos encontrar.

El objetivo es enfatizar en la comparativa de datos encontrados en los artículos mencionados en el punto número 7 de presente documento, donde se encontró las siguientes tablas de conclusión que serán el marco de referencia para realizar las comparaciones pertinentes de nuestro estudio.

Table 2: EDA Dataset – Machine Learning Perfor-mance.				
Model	**Accuracy**	**Precision**	**Recall**	**F1**
Random Forest (RF)	95.518	89.610	96.890	92.685
Decision Tree (DT)	95.357	89.385	96.545	92.414
Logistic Regression (LR)	94.422	87.696	95.949	91.050
AdaBoost (AB)	93.572	86.355	94.943	89.773
Na¨ıve Bayes (NB)	92.808	85.456	92.355	88.32
k-NN	89.494	80.515	90.566	84.030
Soft Voting – Top 3 (RF, DT, LR)	95.433	89.445	96.850	92.560
Soft Voting – Top 5 (RF, DT, LR, AB, NB)	95.571	89.743	96.820	**92.750**

Pagina 5 https://mendel-journal.org/index.php/mendel/article/view/211/194

Model	Dataset	Cleaning	Accuracy	F1-Score	Precision	Recall	AUC
LR	1	Lemma	0.9210	0.9073	0.9384	0.8783	0.9165
RF	2	Lemma	0.9188	0.9011	0.9578	0.8507	0.9110
LSA+SVM	1	Lemma	0.9248	0.9112	0.9486	0.8767	/*
BERTweet	1	Cleaned	0.9238	0.9140	0.9084	0.9197	0.9248
LDA+LR	1	Lemma	0.9190	0.9046	0.9391	0.8726	0.9140
LDA+RF	2	Lemma	0.9144	0.8973	0.9367	0.8611	0.9085

Pagina 9 - Table 6.Summary results. * The AUC value is missing due to computational capacity issue, but itmust be similar to the BERTweet's one (https://www.researchgate.net/publication/370792011_Comparison_between_Machine_Learning_and_Deep_Learning_Approaches_for_the_Detection_of_Toxic_Comments_on_Social_Netw

orks)

Dataset	Approach	Model	Accuracy	Precision	Recall	F-score	ROC
Hate Speech and Offensive Language	Machine Learning Models	SVM	0.873	0.852	0.862	0.851	0.78
		KNN	0.856	0.839	0.831	0.837	0.92
		NB	0.874	0.832	0.863	0.851	0.80
		DT	0.602	0.524	0.585	0.642	0.65
		RF	0.851	0.854	0.822	0.856	0.77
		LR	0.862	0.853	0.837	0.858	0.78
	Deep Learning Models	CNN	0.892	0.895	0.898	0.896	0.93
		LSTM	0.901	0.896	0.91	0.898	0.93
		BiLSTM	0.902	0.916	0.904	0.899	0.94
Twitter Hate Speech	Machine Learning Models	SVM	0.873	0.852	0.862	0.851	0.75
		KNN	0.856	0.839	0.831	0.837	0.90
		NB	0.874	0.832	0.863	0.851	0.76
		DT	0.602	0.524	0.585	0.642	0.68
		RF	0.851	0.854	0.822	0.856	0.77
		LR	0.862	0.853	0.837	0.858	0.78
	Deep Learning Models	CNN	0.892	0.895	0.898	0.896	0.92
		LSTM	0.901	0.896	0.91	0.898	0.92
		BiLSTM	0.902	0.916	0.904	0.899	0.93
Cyberbullying	Machine Learning Models	SVM	0.873	0.852	0.862	0.851	0.75
		KNN	0.856	0.839	0.831	0.837	0.80
		NB	0.874	0.832	0.863	0.851	0.79
		DT	0.602	0.524	0.585	0.642	0.67
		RF	0.851	0.854	0.822	0.856	0.78
		LR	0.862	0.853	0.837	0.858	0.78
	Deep Learning Models	CNN	0.892	0.895	0.898	0.896	0.91
		LSTM	0.901	0.896	0.92	0.898	0.91
		BiLSTM	0.902	0.916	0.904	0.899	0.93

Pagina 403 - TABLE I. COMPARISON OF THE OBTAINED RESULTS https://www.researchgate.net/publication/371334170_Hate_Speech_Detection_in_Social_Networks_using_Machine_Learning_and_Deep_Learning_Methods

Para ello se define que los datos serán evaluados haciendo uso de las siguientes técnicas

- Logistic Regression
- AdaBoost
- Random Forest
- Decision Tree
- Naive Bayes
- Métodos ensamblados
- Redes neuronales

Para poder determinar ¿Cuál es mejor? Se deben usar los siguientes criterios de evaluación haciendo análisis comparativo con estudios previo, para ello vamos a clasificar cual puede ser el mejor modelo con base en su precisión, que se divide en los tres subpuntos mencionados a continuación.

- Precisión de la clasificación

  F1-Score: Es una métrica crítica que balancea la precisión y la exhaustividad del modelo (recall), especialmente en escenarios con clases desbalanceadas (por ejemplo, más mensajes no inapropiados que inapropiados).

  Accuracy (Exactitud): Aunque no es ideal en clases desbalanceadas, puede servir como un punto de referencia general.

  Precision y Recall: Se debe medir la capacidad del modelo para identificar correctamente los mensajes inapropiados (precisión) y para detectar todos los mensajes inapropiados reales (recall).

Con base en lo anteriormente mencionado y comparando con algunos artículos que ya muestran resultados se concluye que el criterio de éxito definido esta argumentado por métodos de Machine learning aplicados al problema propuesto y su precisión.

6. Material de apoyo que pueda ser utilizado para darle contexto a los estudiantes, como artículos, libros, tutoriales u otros recursos que puedan ser útiles para abordar el problema.

Para este punto vamos a tacar los insumos desde el punto de vista de apoyo técnico, que pueden ser cursos o material didáctico que permita a los participantes del equipo reforzar conocimiento aplicados para desarrollar el ejercicio, a nivel teórico se proponen artículos o libros de referencia que abordar conceptos.

Cursos o recursos digitales técnicos.

Nombre del curso o material	**Enlace de acceso o lugar para acceder al material**
Decision Trees, Random Forests, AdaBoost & XGBoost in Python	https://www.udemy.com/course/machine-learning-advanced-decision-trees-in-python/?srsltid=AfmBOoo6OD4TladqPE7EiO_FruuZsQTrHYyaSBwDU7Pxr09EzvbkDoEm&couponCode=LETSLEARNNOW
Machine Learning Full Course 2025 | Machine Learning Tutorial | Machine Learning Roadmap | Edureka	https://www.youtube.com/watch?v=dQDoAmkrSQ8&t=13546s
Transfer learning for hate speech detection in social media	https://link.springer.com/article/10.1007/s42001-023-00224-9
How to build a hate speech detection in class	https://www.substring.ch/glossar/how-to-build-a-hate-speech-detection-in-class
Identification of Hate Speech on Social Media using LSTM	https://www.researchgate.net/publication/371911648_Identification_of_Hate_Speech_on_Social_Media_using_LSTM
Analysis of hate speech detection in social media	https://diposit.ub.edu/dspace/bitstream/2445/182589/2/tfg_ferran_sanchez_llado.pdf

Para justificar los criterios de éxito se toma información de investigaciones realizadas; para la propuesta actual se encontró que hay diversos estudios a replicar entre ellos podemos resaltar "Hate speech and offensive language" cuyo estudio replica anteriores investigaciones realizando comparación entre las diferentes técnicas de Machine Learning donde alguno de los criterios de exito son el F1-score. Para tal fin podrá ver los detalles en los siguientes notebooks que remarcan sus resultados

Notebook 1

Noteboot 2

Como en la mayoría de los casos que podemos encontrar de estudios similares, lo que se hace es una evaluación de la data a través de diferentes técnicas de Machine Learning. Y esto nos proporciona un marco de referencia para la ejecución del trabajo propuesto.

7. Resumen de tres artículos académicos directa y claramente relacionados a su propuesta

Offensive Language Detection Using Soft Voting Ensemble Model

- Introducción: ¿Cuál es el problema o tema que aborda el paper?

Con el crecimiento del internet, se han popularizado masivamente las redes sociales por dar la capacidad de compartir información y comunicarse libremente. No obstante, con esta popularización y la capacidad de comunicación libre, se ha expandido el lenguaje ofensivo. El lenguaje ofensivo consiste en insultos, discriminación, amenazas y en algunos casos puede llegar a ser discurso de odio, el cual consiste en atacar ciertos grupos sociales. Para combatir el lenguaje ofensivo se puede hacer moderación, no obstante, con la rápida generación de contenido en redes sociales esto se vuelve inviable si se hace manualmente. Por esto, con machine learning y deep learning se puede automatizar la moderación.

- Objetivo: ¿Qué se busca lograr con la investigación?

En investigaciones anteriores se han utilizado distintas aproximaciones para atacar el problema de clasificar el contenido ofensivo usando machine learning (ML), deep learning (DL), híbridos y soft voting. Hay demasiados algoritmos de ML y DL que se pueden usar para soft voting. Por esto anterior, el articulo buscaba mejorar el rendimiento de la clasificación de lenguaje ofensivo experimentando con combinaciones de ML y DL para modelos ensamblados.

- Metodología: ¿Cómo se llevó a cabo el estudio (enfoques, datos, herramientas, modelos, experimentos, etc.)? – acá se encuentra el corazón del resumen.

Para llevar a cabo el estudio, se tomó el dataset de un artículo en el cual se tomaron datos de twitter y se clasificaron como hate speech, offensive speech o ninguna de las anteriores. Las proporciones del dataset eran de 1430, 19190 y 4163 datos respectivamente. Una de las primeras transformaciones que se hicieron fue la fusión de las categorías de hate speech y ofensive speech en una sola para detectar hate speech en general. Las proporciones resultantes fueron 20620 para la clase de ofensivo y 4163 para la clase de no ofensivo.

Para el procesamiento, se tomó el dataset y se quitaron elementos como URLs, menciones de usuarios y puntuación. Posteriormente se hizo stemming sobre todas las palabras. Después de esto se hizo Easy Data Augmentation (EDA) para balancear el dataset porque (como se vio antes) la clase de lenguaje ofensivo era significativamente más grande que la de no lenguaje no ofensivo. Este proceso consistió en tomar la clase de menor tamaño y aplicar las técnicas de EDA aleatoriamente sobre los datos. Las técnicas usadas fueron synonym replacement, random insertion, random swap y random deletion. Estos métodos se aplicaron hasta que los tamaños de los datasets fueran iguales lo que resultó en 2 datasets, uno aumentado y el original.

Para la extracción de características, primero, se tomaron los tweets y se separaron las oraciones en palabras (tokenization). Posteriormente se hizo una extracción diferente para deep learning y machine learning. Para ML se uso TF-IDF para sacar la importancia de cada palabra en un documento y análisis de sentimiento para sacar información de sentimiento como puntajes positivos, negativos y neutros. Para el análisisg de sentimiento se usó el Valence Aware Dictionary and Sentiment Reasoner (VADER). Para DL se usó GloVe embedding para extraer características usando vectores de twitter pre entrenados con 27 millones de tokens, 1.2 millones de vocabularios y 200 vectores de dimensión.

Para el desarrollo del modelo se usó el voting classifier (el cual es un método ensamblado) el cual combina varios modelos/estimadores los cuales votan para predecir la clase de un dato. Existen 2 tipos de clasificadores de votación, en hard voting se selecciona la clase en base a la votación de la mayoría de los clasificadores, mientras que en soft voting se calcula la probabilidad de cada clase para cada estimador y con esto se determina la predicción final. El estudio se divide en 2,

ML y DL. Para machine learning se usaron modelos populares en investigaciones: Random forest, Naive Bayes, Decision Tree, Logistic Regression, AdaBoost y KNN. Cada uno de los modelos fue entrenado y evaluado con validación cruzada en 5 para tanto el dataset original como el aumentado. De este proceso se tomaron los modelos en el top 3 y top 5 f1-score de cada dataset para el modelo de votación. Para DL los modelos también se entrenaron con validación cruzada en 5 y se usaron 3 grupos de DL para ambos datasets (CNN, LSTM (LSTM y bidireccional) y Gated Recurrent (GRU y bidireccional)). De nuevo, se tomaron los modelos con los mejores f1-scores de cada grupo para el estimador de votación.

La arquitectura de los modelos de DL consistió en la capa de entrada, la capa de embedding usando GloVe con salida de 512, una capa de modelo (que depende del cual se estuviera entrenando) y una capa de dropout con una tasa de 0.2 para evitar el overfitting. La salida de la capa de dropout se enviaba a otra capa de modelo cuyo resultado se enviaba a otra capa de dropout con la misma tasa y salida de 256. Luego, siguió una capa densa con la función de activación Rectified Linear Unit (ReLU) que reduce el output a 64. Finalmente, se enviaron los datos a una capa densa con la función softmax para sacar las probabilidades de las 2 clases para la votación.

En general para la evaluación se uso macro average para medidas como accuracy, F1, precisión y recall.

- Resultados: ¿Cuáles son los hallazgos más relevantes?

Para machine learning, y usando el dataset original, el modelo con el mayor f1-score fue el random forest con 90.070%. El orden de los modelos de mayor f1-score a menor fue: Random Forest, Logistic Regression, AdaBoost, Decision Tree, Naive Bayes y KNN. Se tomó el top 3 y top 5 f1-scores para hacer modelos de votación y se obtuvo un f1 score de 91.370% para el top 3 y 91.509% para el top 5. Usando el dataset aumentado se obtuvo que el mejor modelo fue Random Forest con f1-score de 92.685%. Los modelos en orden de f1-score descendiente fueron Random

Forest, Decision Tree, Logistic Regression, AdaBoost, Naive Bayes y KNN. El top 5 logró un score de 92.750% siendo superior al top 3 de este dataset.

Para DL y el dataset original se tomaron los modelos con mejores métricas, siendo estos CNN, BI-LSTM y BI-GRU siendo el BI-LSTM el que mejor resultados obtuvo. El modelo de votación obtuvo un score de 91.864% lo cual es mejor que el que se obtuvo con el dataset original con machine learning. En general, todos los modelos mejoraron con respecto a machine learning. Con el dataset aumentado se usaron los mismos modelos ya que fueron los mejores de nuevo. El mejor de todos fue BI-LSTM con f1-score de 93.56%. El modelo de votación obtuvo 93.818% de f1-score superando por poco al BI-LSTM. Se puede decir que la mejora usando votación fue mayor en los datos originales que en los datos aumentados, aun así, el modelo de votación con el dataset aumentado y DL fue el mejor del estudio.

- Conclusión: ¿Qué implicaciones tienen los resultados y cuál es la contribución principal del trabajo? ¿Qué ideas puede tomar para su proyecto?

Para ML se encontró que la combinación de Random Forest, Decision Tree, Logistic Regression, Naive Bayes y AdaBoost supero a los demas modelos tanto en el dataset original como en el aumentado. Esto anterior tambien ocurrio en deep learning pero con los modelos de CNN, BI-LSTM y BI-GRU.

En general, para todas las pruebas, se puede decir que usar el dataset aumentado generó mejores resultados.

El trabajo contribuyó al problema principalmente mediante el logro de mejores metricas que en varias investigaciones anteriores y logrando buenos resultados tanto en machine learning como en deep learning mediante el uso de soft voting. Se concluye que soft voting con deep learning obtuvo mejores resultados que estimadores de machine learning. El estudio no descarta que en futuras investigaciones se puedan encontrar otras combinaciones con resultados interesantes para explorar.

Para nuestro proyecto del curso podemos tomar bastantes ideas de esta investigación. Primero que todo, se nos da un dataset que podriamos utilizar para entrenar los modelos del proyecto (datos de twitter clasificados como hate speech, offensive o ninguno). Por otra parte, el estudio compara machine learning con deep learning y nos da la idea de probar metodos ensamblados y varios algoritmos no ensamblados para encontrar la combinación que nos de los mejores resultados tanto para machine learning como para deep learning y compararlos con el otro y con los métodos sin ensamblar. Otras ideas que nos da el articulo son las formas en que podemos preparar, y extraer caracteristicas de los datos de texto para el uso en los modelos. Principalmente, el articulo nos inspira a usar EDA, analisis de sentimientos con VADER, TF-IDF y GloVe embedding para el preprocesamiento de los datos. Tambien, nos explica formas de limpiar datos como lo puede ser stemming, quitar URLs y quitar la puntuación. Por ultimo, se nos da un listado de modelos que podríamos probar o usar para nuestro proyecto (Random Forest, Logistic Regression, AdaBoost, Decision Tree, Naive Bayes, KNN, CNN, LSTM, BI-LSTM, GRU y BI-GRU)

Comparison between Machine Learning and Deep Learning Approaches for the Detection of Toxic Comments on Social Networks

- Introducción: ¿Cuál es el problema o tema que aborda el paper?

Debido a la revolución tecnológica de los últimos años el internet ha masificado su uso globalmente. Por esto anteriores nacieron las cibercomunicaciones que se ejecutan en el contexto de la ciber cultura. Estos conceptos nacen como producto del internet, las redes sociales, el uso de información nueva y las nuevas tecnologías. No obstante, este contexto permite falsificar la identidad o mantenerla secreta lo que le da a los individuos libertades que pueden resultar en mal comportamiento hacia los demás. Consecuencia de lo anterior y el gran volumen de mensajes diarios que se envían, los mensajes de odio, ofensivos y abusivos han visto un incremento y es imposible de moderarlos manualmente, cosa que ha vuelto esencial la investigación de métodos de detección para regulación. Este problema ha sido de tal importancia que hasta la Unión Europea ha dado ordenes de remover este tipo contenido y han expresado la relevancia de los métodos de detección automática.

- Objetivo: ¿Qué se busca lograr con la investigación?

La investigación busca comparar 3 métodos usados frecuentemente en retos de NLP (Regresión logística, RandomForest y Support Vector Machine - SVM) combinados con técnicas de modelamiento de tópicos (Latent Semantic Analysis - LSA y Latent Dirichlet Allocation - LDA) con la transformer architecture (que se menciona que es el estado del arte en análisis de texto con Deep learning) en la tarea de detección de mensajes tóxicos en las redes sociales usando métricas tradicionales de costo computacional y temporal.

- Metodología: ¿Cómo se llevó a cabo el estudio (enfoques, datos, herramientas, modelos, experimentos, etc.)? – acá se encuentra el corazón del resumen.

Para el estudio se tomaron 9 datasets públicos que contienen tweets marcados con diferentes categorías de toxicidad dependiendo del dataset (ej. Hate speech, racismo, entre otros).

https://github.com/AkshitaJha/NLP_CSS_2017

https://hasocfire.github.io/hasoc/2019/dataset.html

https://github.com/zeeraktalat/hatespeech

https://github.com/melsherief/hate_speech_icwsm18

https://ckan.hatespeechdata.com/dataset/founta-et-al-hate-and-abusive-speech-on-twitter/resource/9ccd6c1e-a7d2-4298-be72-fe78f529364e

https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset

Utilizando los 9 datasets, estos se integraron y se marcaron como neutral “0” y toxico “1” lo que resulto en 111,131 tweets únicos en ingles con 57.43% de los tweets siendo neutrales y 42.57% siendo tóxicos.

El siguiente paso fue limpiar los datos para solamente dejar la información relevante para detección de sentimentos. Para esto se quitaron las menciones, URLs, el símbolo # y las puntuaciones. Tambien, todo el texto se paso a minúsculas, y se quitaron las palabras de longitud menor a 2 caracteres.

Como los emojis pueden dar información importante del sentimiento, se realizaron 2 datasets para el estudio. El primer dataset tiene el texto limpio y lematizado resultando en 100,105 datos, y el segundo es igual pero sin stop words y traduciendo los emojis a strings resultando en 98,701 datos. Para ambos datasets se quitaron los duplicados.

Lo siguiente que se hizo fue extraer features mediante la transformación del texto en vectores numéricos para el uso en los modelos. Para lograr esto se usaron 2 metodos: El primero fue bag of words (BOW) en el que se hace una matriz n x m donde n son las palabras únicas en los datos y m el numero de datos. La matriz tiene un 1 si la palabra aparece y un 0 si no. El segundo método fue Term Frequency-Inverse Document Frequency (TF-IDF) que promueve los términos menos comunes sobre los mas comunes usando la frecuencia en que salen en un documento y la frecuencia en todos los documentos. No obstante, las 2 tecnicas pierden la relación entre las palabras y por ende el contexto. Por esto anterior se uso una tercera técnica “Word embeddings” que asigna vectores cercanos en un espacio continuo multidimensional a términos similares lo que permite calcular similitud entre palabras y oraciones con distancias o generalizar oraciones en otras similares.

Para los experimentos de machine learning se usó BOW y TF-IDF y, primero, se entrenó un modelo con regresión logisitica para ver el panorama del dataset, además, este algoritmo es eficiente, sencillo de entrenar, no requiere hiperparametros y se pueden interpretar los pesos. Para el segundo experimento se uso el método ensamblado de RandomForest que es más poderoso que otros algoritmos de clasificación no lineales. El tercer experimento fue con el LDA para reducir la dimensionalidad del dataset mediante el descubrimiento de temas en un set de documentos y SVM para separar las clases con un kernel radial. Finalmente, se uso LDA con el fin de dividir el dataset en subsets relacionados para hacer pruebas de forma separada sobre modelos.

Para deep learning se usó word embeddings y transformers para extraer caracteristicas que contengan información semántica. Para deep learning se utilizo el modelo “BERTweet” que es un LLM entrenado con 850 millones de tweets en ingles y basado en la misma arquitectura que la BERT-base. El modelo es entrenado usando el procedimiento de entrenamiento RoBERTa.

Los procedimientos fueron ejecutados usando scikit-learn y una maquina virtual de Google Colab Pro.

- Resultados: ¿Cuáles son los hallazgos más relevantes?

Para el experimento de regresión logística se tomaron 30% de los datos para pruebas y 50 seeds para tener resultados independientes en términos de factores de aleatoriedad de la partición. Posteriormente, se tomaron las métricas y se calculó la desviación estándar (la cual fue limitada a las milésimas). De este experimento se concluyó que bag of words fue la mejor opción para ambos datasets. Se obtuvo un f1-score de 0.9073 y un accuracy de 0.9210 para el primer dataset. Se obtuvo un f1-score de 0.9056 y un accuracy de 0.9206 para el segundo dataset.

Para Random Forest se tomó 15% para test y 15% para validación. Para encontrar los mejores hiperparametros se uso cross validation tanto para la configuración de bag of words como para el modelo. Una vez se eligió el mejor modelo con el conjunto de validación, se obtuvo 100 arboles de decisión con profundidad máxima de 80 y con el requerimiento de que se necesitaban 10 samples para dividir un nodo. Respecto a los parámetros de bag of words se ignoraron los términos con frecuencia de más de 0.2% y menores a 0.005% para reducir el vocabulario. Se obtuvo un f1-score de 0.9007 y un accuracy de 0.9178 para el primer dataset. Se obtuvo un f1-score de 0.9011 y un accuracy de 0.9188 para el segundo dataset.

Para SVM con LSA se realizó TF-IDF, se hizo singular value decomposition, se hizo normalización y se uso SVC buscando un C optimo. LSA resultó ser muy costoso computacionalmente entonces solo se hizo uso del primer dataset y se hizo la prueba con 500, 1000 y 5000 dimensiones. Con 5000 dimensiones y C = 4 se logró un f1-score de 0.9112 y un accuracy de 0.9248.

Para el transformer se utilizó tanta información como fuera posible entonces se usó el dataset 1 ya que BERTweet y su procedimiento de preentrenamiento incluye emojis. El primer paso para el transformer fue ajustar la capa de cabecera del transformer donde se entrenan los pesos. El segundo paso fue crear una red neuronal de clasificación binaria después del transformador entrenando todos los pesos. Por último, se congelaron los pesos del transformer y se entrenó solamente la red de clasificación binaria. El mejor f1-score (0.9140) se consiguió con 4 epochs y ajustando la capa de cabecera del transformador con una tasa de aprendizaje de 2×10^{-5} en el algoritmo de optimización Adam y un hidden dropout probability de 0.3.

Para LDA con modelos locales se tomaron 3 topicos del algoritmo no supervisado para los datos de entrenamiento, luego se hicieron inferencias para clasificar los tweets de validación y testeo en los 3 topicos y de esta manera tener 3 sets de entrenamiento, validación y testing. Lo siguiente fue aplicar regresión logística y random forest en cada partición y al tener una métrica para cada subset se concatenaban para tener una métrica global en el test set. El mejor f1-score fue con regresión logística usando el primer dataset, este obtuvo 0.9046 de f1-score y 0.9190 para accuracy.

- Conclusión: ¿Qué implicaciones tienen los resultados y cuál es la contribución principal del trabajo? ¿Qué ideas puede tomar para su proyecto?

Algo que se menciona en la investigación es que el lenguaje es muy versátil y ambiguo lo que incrementa las formas de ser ofensivo de manera sutil y que un algoritmo no podría detectar.

Los resultados implican que el modelo transformer tiene ligeramente un mejor rendimiento que los modelos de machine learning (considerando Accuracy, F1-score, Precision, Recall, AUC y curva ROC), no obstante, este modelo requiere más costo computacional para el entrenamiento y evaluación a diferencia de los demás modelos. Esto implica que un modelo más simple como la regresión logística podría ser preferible al tener un rendimiento similar y mayor capacidad de explicabilidad. En general, se podría decir que el modelo transformer no ofrece mucho para resolver el problema ya que las ganancias no son demasiadas en comparación con los altos costos.

De esta investigación se pueden tomar varias ideas para el proyecto. Primero que todo, se nos dan más fuentes de datos para tratar el problema nosotros mismos. Al igual que en el estudio anterior, se nos dan pasos específicos que podemos seguir para el preprocesamiento y extracción de características de los datos, entre los mencionados podemos destacar el cambio de emojis por palabras, los métodos BOW, Word embedding y TF-IDF, el uso de LDA para la extracción de tópicos para el entrenamiento de modelos, o la reducción de dimensionalidad con LSA. Tambien, se nos da a entender que podríamos evitar el uso de tranformer y BERTweet por su alto costo y poco beneficio. Otra idea que se puede sacar del artículo es que en nuestro proyecto podríamos evaluar el costo computacional y buscar el modelo con el mejor trade off de costo computacional y f1-score. Por ultimo, podemos tomar del artículo los múltiples modelos e hiperparametros usados que podríamos emplear o basarnos en para nuestras pruebas.

Hate Speech Detection in Social Networks using Machine Learning and Deep Learning Methods

- Introducción: ¿Cuál es el problema o tema que aborda el paper?

Las redes sociales han crecido y se han vuelto esenciales en la época moderna conectando y permitiendo a los usuarios compartir entre ellos. Este crecimiento a creado un incremento en el hate speech (forma de comunicación ofensiva y discriminatoria que toma de objetivo grupos sociales, étnicos, entre otros). El crecimiento del hate speech es un problema critico que lastima la libre expresión y el discurso respetuoso. Consecuentemente, se tiene la necesidad de tener herramientas efectivas de detección y monitoreo. Actualmente, las técnicas de machine learning se han presentado como una posible solución gracias su potencial en resolver tareas de NLP.

- Objetivo: ¿Qué se busca lograr con la investigación?

El articulo busca investigar y comparar el rendimiento de múltiples métodos de machine learning tradicional y de Deep learning en el contexto de la detección de hate speech en twitter. Con esto

anterior se quiere contribuir al desarrollo de herramientas y estrategias para combatir el hate speech en redes sociales y crear un ambiente sano.

- Metodología: ¿Cómo se llevó a cabo el estudio (enfoques, datos, herramientas, modelos, experimentos, etc.)? – acá se encuentra el corazón del resumen.

Los modelos fueron realizados y evaluados usando 3 datasets: Cyberbullying, Hate Speech and offensive language, y Twitter hate speech.

Para el desarrollo del proyecto se realizó una metodología de 4 pasos: preprocesamiento, extracción de características, clasificación y evaluación.

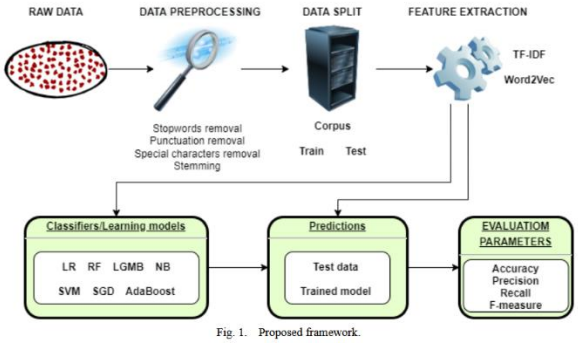


Fig. 1. Proposed framework.

Para la extracción de características se utilizó TF-IDF que refleja la importancia de un documento y en todo el corpus. Una calificación alta implica alta importancia dentro del documento y baja frecuencia en todo el corpus haciendo la característica valiosa. Con este método se puede transformar la información de texto a información estructurada con la importancia relativa de los términos. Los vectores de características resultantes se pueden utilizar para entrenar los modelos y detectar el hate speech. Este método se ha probado efectivo en varias tareas pero no captura el contexto ni la semántica compleja, por esto, el método puede ser reemplazado o complementado con técnicas más avanzadas como modelos pre-entrenados o Word embeddings y de esta forma crear modelos más sofisticados para la detección de hate speech.

Word2Vec es otra técnica significativa que sirve para crear Word embeddings. Esta técnica es no supervisada y convierte las palabras en representaciones continuas de vectores para capturar relaciones semánticas y sintácticas. Este método se ha vuelto popular en NLP para tareas como clasificaciones, análisis de sentimientos, entre otros. La técnica se puede utilizar para enriquecer features y mejorar el rendimiento de clasificación. Inclusive, se puede combinar con modelos pre entrenados y TF-IDF para mejorar la capacidad de clasificación aun más.

Bag of words es una tecnica para representar texto y es ampliamente utilizado en NLP al dar una representación simple y estructurada de los textos. Al igual que TF-IDF, carece capacidad de capturar el contexto, orden y semántica, por eso, tambien se puede combinar con técnicas como Word embeddings y modelos pre-entrenados.

Entre los metodos de machine learning usados esta el árbol de decisión. El árbol de decisión es un algoritmo supervisado que separa el input en regiones basándose en valores de las características. Este algoritmo permite manejar relaciones no lineales y es fácilmente interpretable. Otro metodo usado fue Naive bayes el cual es un clasificador probabilístico basado en el teorema de bayes que asume independencia de características. Este algoritmo se ha probado ser efectivo en clasificación de textos. La salida del método es la probabilidad de que un texto sea de una clase especifica. K-Nearest Neighbors es un algoritmo sin parámetros que clasifica basándose en la clase de los K puntos vecinos. Esta técnica es útil para clasificar por similaridad entre textos basándose en las características extraídas. Support vector machine es otro algoritmo usado, supervisado, que busca el hiperplano optimo que separe las clases en el espacio de características. Esta técnica es bastante efectiva al manejar datos de alta dimensionalidad y puede usar kernels para transformar el espacio. Esta técnica clasifica aprendiendo la frontera de decisión entre las características usadas.

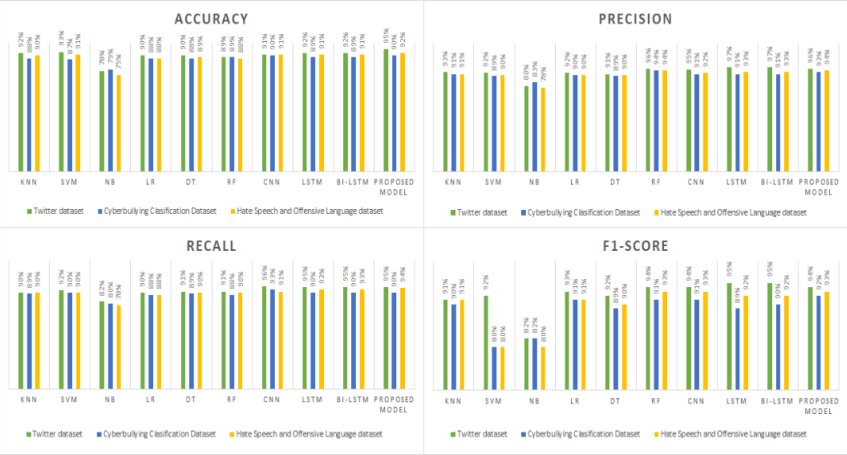
Para Deep learning, el primer método que se menciono que se usó fue LSTM. Long short-term memory (LSTM) resuelve el vanishing gradient problem y usa memoria para almacenar información sobre largas secuencias de información lo que permite identificar dependencias y contexto en la información. Gracias a lo anterior se puede mejorar la distinción entre clases.

BiLSTM (Bidirecctional long short-term memory) es una extensión de LSTM que procesa la información al derecho y al revés para capturar contexto tanto pasado como futuro. Gracias a lo anterior se puede sacar mejor información sobre dependencias y contexto y mejorar el rendimiento. Las BiLSTM se pueden combinar con otras arquitecturas como las redes neuronales convolucionales para mejorar la habilidad de detectar información local y global del contexto. Por ultimo, las CNN (Redes neuronales convolucionales) es una arquitectura usada para el procesamiento de imágenes pero se ha demostrado útil para tareas de NLP. Las CNN usan capas convolucionales para aprender patrones locales en la información de entrada usando filtros y kernels. Las CNN se pueden usar con texto, tratando este como una secuencia uni-dimensional de palabras o caracteres. Combinando CNN con otras arquitecturas como LSTM y BiLSTM se pueden sacar tantos patrones locales como lejanos, mejorando significativamente las clasificaciones.

Para la evaluación de los modelos se usó: Accuracy, Precision, Recall, F1-score, la matriz de confusión y la ROC curve. La ROC al graficar recall contra el false positive rate crea un área que se le llama AUC-ROC y entre más cercana a 1 mejor el modelo.

- Resultados: ¿Cuáles son los hallazgos más relevantes?

De los resultados se puede decir que los métodos de Deep learning probaron ser más valiosos que los de machine learning al lograr clasificar 3 clases: Cyberbullying, neutro y no cyberbullying. Los resultados se muestran a continuación:



Vale la pena resaltar que el modelo propuesto es un Bi-LSTM para el cual se uso una red neuronal para modificar los pesos y bias lo que resulto en esta logrando el mejor rendimiento entre todos los modelos. Adicionalmente, este modelo logró un menor tiempo de entrenamiento.

- Conclusión: ¿Qué implicaciones tienen los resultados y cuál es la contribución principal del trabajo? ¿Qué ideas puede tomar para su proyecto?

Algunas implicaciones que tienen los modelos utilizados es que, como ya dicho, los modelos básicos de ML pueden fallar capturando contexto. Por otra parte, los modelos de Deep learning pueden fácilmente hacer overfitting y necesitar grandes volúmenes de datos. Adicionalmente, son más costosos de entrenar y tienen menos capacidad de ser interpretados.

Otra implicación del estudio es que el hate speech evoluciona constantemente y cada día aparecen más code words y material no textual que podría ser considerado hate speech y que no puede ser capturado por los modelos o técnicas actuales de extracción de características. Tambien, los modelos pueden tener falsos positivos y falsos negativos que pueden conllevar a supresión injustificada de la libertad de expresión o permitir la existencia de contenido dañino.

Una de las contribuciones del artículo es que da una base para el desarrollo de herramientas más avanzadas para detección precisa de hate speech la cual puede ser aplicada en múltiples redes

sociales para ayudar a mitigar el hate speech y crear un ambiente seguro e inclusivo en el internet. El articulo tambien contribuye dando luz a la importancia de las técnicas de extracción de características para la extracción de la información necesaria para el entrenamiento y evaluación de modelos. Por ultimo, se puede decir que el articulo contribuyó demostrando la superioridad de las redes neuronales en la tarea de clasificación de hate speech, más específicament, la superioridad de Bi-LSTM.

Para nuestro proyecto del curso podemos tomar de esta investigación los datasets que usan para alimentar nuestros modelos. Otro elemento del articulo que podemos tomar son los resultados, estos nos sirven como una línea base o un objetivo para superar en los modelos que desarrollemos, tambien, los resultados son útiles para ver si vamos por un buen camino. Los resultados del articulo tambien nos llevan a considerar hacer énfasis o poner mayor empeño en modelos como Bi-LSTM ya que como visto, tiene bastante potencial y capacidad de resolver el problema. Es posible que, como mencionado en artículos previos, el hacer un modelo ensamblado con redes neuronales puede ayudarnos a potenciar los resultados que obtengamos. Se puede decir que este articulo, de manera similar a los demas, nos dan ideas y técnicas para la vectorización y extracción de características, por ejemplo, BOW, TF-IFD y Word-embeddings (Word2Vec y GloVe). Por último, vale la pena resaltar que el artículo nos da insights y mayor conocimiento sobre los modelos de redes neuronales que se pueden usar para el proyecto (BiLSTM).

Bibliography

B. Fieri and D. Suhartono, “Offensive Language Detection Using Soft Voting Ensemble Model,” *Mendel*, vol. 29, no. 1, 2023, doi: 10.13164/mendel.2023.1.001.

A. Bonetti, M. Martínez-Sober, J. C. Torres, J. M. Vega, S. Pellerin, and J. Vila-Francés, “Comparison between Machine Learning and Deep Learning Approaches for the Detection of Toxic Comments on Social Networks,” 2023. doi: 10.3390/app13106038.

A. Toktarova *et al.*, “Hate Speech Detection in Social Networks using Machine Learning and Deep Learning Methods,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 5, 2023, doi: 10.14569/IJACSA.2023.0140542.

Amina Saleh Omar, K. O. (2024). The Impact of Social Media on Society: A Systematic Literature Review. *The International Journal of Engineering and Science (IJES)*, 96-106.

Cohen, J. (12 de February de 2025). *A Platform Problem: Hate Speech and Bots Still Thriving on X*. Obtenido de USC Viterbi School of Engineering: https://viterbischool.usc.edu/news/2025/02/a-platform-problem-hate-speech-and-bots-still-thriving-on-x/

Geuens, R. (19 de Enero de 2025). *What are the top social media platforms in 2024?* Obtenido de SOAX: https://soax.com/research/top-social-media-platforms

United Nations. (28 de Enero de 2023). *Hate speech: A growing, international threat*. Obtenido de UN News: https://news.un.org/en/story/2023/01/1132597

Zandt, F. (18 de Junio de 2024). *Meta's Hate Speech Problem*. Obtenido de Statista: https://www.statista.com/chart/21704/hate-speech-content-removed-by-facebook/