Clasificador automático para la detección de lenguaje ofensivo en español en comentarios de redes sociales

Jassiel Luna Cantu, André Michel Marín Espinoza, and Marcos Rodolfo Macías Mondragón

Unidad Profesional Interdisciplinaria de Ingeniería, Campus Tlaxcala

1 Planteamiento del problema

En los últimos años se ha hecho presente como las redes sociales se han vuelto una parte más importante de nuestra vida, permitiendo la interacción y participación entre usuarios, pero de la misma manera, esa misma exposición da paso a los comentarios ofensivos o denigrantes dirigidos a individuos o grupos. Esto logrando crear en internet espacios de hostilidad digital, donde ataques personales y expresiones discriminatorias son observables sin filtro alguno.
En el idioma español, la detección automática de lenguaje ofensivo presenta barreras de falsos positivos en un entorno donde la riqueza y la variabilidad del español permiten una amplia capacidad de expresión. Consistiendo de modismos, expresiones coloquiales, ironía y sarcasmos en el lenguaje se dificulta la detección de comentarios genuinamente ofensivos y los que contengan lenguaje expletivo sin una intensión agresiva.
El corpus OffendES en el 2021 se dió a la tarea de recopilar comentarios de distintas redes sociales como Twitter, Instagram y Youtube, enfocado en su interacción con influencers. El corpus fue etiquetado manualmente en 4 categorías que distinguen entre comentarios no ofensivos, vocablo soez sin agredir y a que objetivos fue dirigido el ataque, asi como el genero del influencer al que fue dirigido.

Objetivo General: Desarrollar y evaluar un sistema automático de detección de comentarios ofensivos a partir de comentarios de redes sociales en español, tomando de base el corpus OffendES.

1.1 Objetivos específicos:

- Implementar un pipeline de preprocesamiento de texto adaptado al español utlizando en redes sociales.
- Explorar el dataset OffendES con el proposito de descubrir patrones comunes o desbalances y como puede afectar el modelo de clasificación

- Entrenar y evaluar modelos de clasificación binaria para la detección de lenguaje ofensivo, utilizando representaciones TF-IDF y algoritmos de aprendizaje supervisado como regresión logística y máquinas de soporte vectorial.
- Implementar un sistema de predicción que permita evaluar el comportamiento del modelo final sobre ejemplos no vistos, analizando cualitativamente sus resultados.
- Evaluar la capacidad de generalización del modelo en diferentes plataformas, evaluando como se comporta el modelo con comentarios provenientes de distintas fuentes a los consideradas.

2 Marco teórico

El corpus OffendES ha sido utilizado previamente para la detección de lenguaje ofensivo en español [1].

3 Metodología

3.1 Descripción de los datos

El corpus usado para este proyecto fue el corpus OffendES, el cual recopiló comentarios en español de diferentes redes sociales (Youtube, Instagram y Twitter) en el año 2021, principalmente dirigidos a influencers conocidos hispanohablantes. El archivo de entrenamiento cuenta con 16,710 registros contando con 6 atributos, siendo estos los siguientes:

- **Comment_id:** Identificador único del comentario
- **Comment:** El texto del comentario publicado en la red social.
- **Influencer**: Nombre del influencer al que el comentario va dirigido
- **Influencer_gender**: Género del influencer
- **Media**: La plataforma en donde se publicó el comentario
- **Label**: Clasificación de ofensa, siendo:
  - **OFP**: Ofensivo, atacando a una persona
  - **OFG**: Ofensivo, atacando a un grupo
  - **NOE**: No ofensivo, pero con lenguaje expletivo
  - **NO**: No ofensivo

En la revisión del corpues entre los parámetros que se observaron fueron la longitud del texto y la cantidad de emojis, donde los datos a primera vista se veian muy unidos hasta el tercer cuartil, donde para el valor encontrado de máximo se disparaba (1), para ello se decidió hacer una observación más profunda, analizando ambos datos con un boxplot y observar como se comportaban y si era necesario poner un límite, donde en ambos casos, el dominio que abarcaban outliers era muy por encima de el grueso de la población (2).

comment_length		n_emojis	
Min.	: 3.0	Min.	: 0.0000
1st Qu.	: 84.0	1st Qu.	: 0.0000
Median	: 134.0	Median	: 0.0000
Mean	: 172.3	Mean	: 0.5661
3rd Qu.	: 199.0	3rd Qu.	: 0.0000
Max.	:9887.0	Max.	:96.0000

Fig. 1. Estadísticos de comment_lenght y n_emojis

Revisando el dataset a más profundidad para encontrar estos outliers, en su mayoría se trataban de mensajes de spam o de textos copypaste, que a pesar de entrar en la categoría "NO", la grande cantidad de texto sumada al desapego al tema a tratar podría añadir ruido a nuestro modelo, por lo cual los mensajes fueron recortados a mil palabras (todavía manteniendo más del 75% de registros intactos).

La mayoría de los mensajes no contienen emojis, y durante el proceso de limpieza pudieron haber sido eliminados del mensaje, pero se decidió mantenerlos por un propósito de que integran parte de la intención del mensaje, siendo que fueron transcritos a sus descripciones durante la limpieza, como a *cara_sonriente*, o *cara_con_nauseas*, limitándolos a 45 emojis por mensaje.

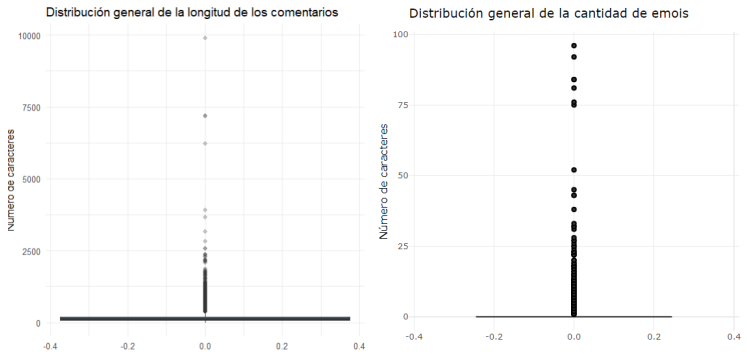

Fig. 2. Graficos de caja de comment_lenght y n_emojis

Para el entrenamiento del modelo fueron usados los registros siguieron la siguiente distribución, acumulando alrededor de 15% de los datos de comentarios ofensivos (3), encontrando un alto desbalance de clases que pudiera llevar a un sezgo durante el entrenamiento. Así mismo se pudo observar como en el caso cuando los influencers son mujeres, es mayor el porcentaje de comentarios ofensivos a comparación de cuando el influencer es hombre (4), esta observación hizo revisar los términos más usados por los comentarios ofensivos tanto en hombres y mujeres (5).

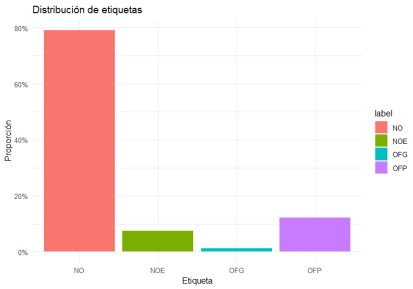

Fig. 3. Distribución de etiquetas

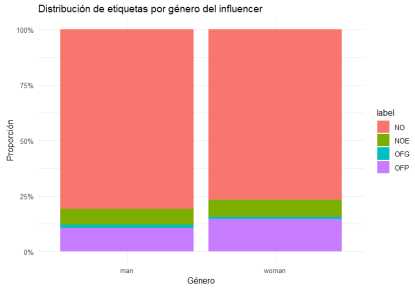

Fig. 4. Distribución de etiquetas por género

Fig. 5. Palabras mas relevantes en comentarios tóxicos dirigidos a influencers por genero

3.2 Pipeline

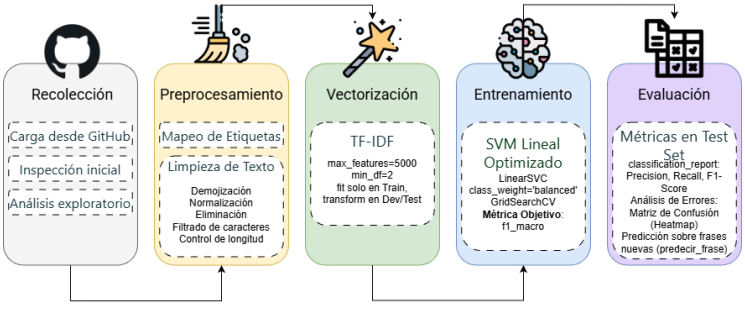

Fig. 6. Pipeline de procedimientos

Para la recopilación se usaron los conjuntos de entrenamiento, validación y prueba proporcionados por el corpus OffendES,

El preprocesamiento del texto fue diseñado específicamente para conservar información semántica relevante del español coloquial utilizado en redes sociales. A diferencia de enfoques tradicionales que eliminan emojis, se optó por convertirlos a descriptores textuales en español mediante el proceso de *demojización*, ya que los emojis pueden aportar señales importantes sobre la intención comunicativa del mensaje.

Adicionalmente, se aplicaron las siguientes técnicas:

- Normalización a minúsculas.
- Eliminación de URLs, menciones y hashtags.
- Filtrado de caracteres no alfanuméricos, preservando tildes y caracteres propios del español.
- Control de la longitud máxima del texto para evitar la dominancia de comentarios extremadamente largos.
- Limitación del número de tokens provenientes de emojis para reducir ruido.

Este preprocesamiento busca un equilibrio entre limpieza del texto y preservación del contenido expresivo característico de las interacciones en redes sociales.

Para la representación del texto se empleó el esquema TF-IDF, limitando el vocabulario a las 5,000 características más relevantes y descartando términos con muy baja frecuencia. Esta elección se justifica por su eficacia probada en tareas de clasificación de texto y su interpretabilidad, lo que facilita el análisis posterior del comportamiento de los modelos.

El vectorizador fue ajustado exclusivamente con el conjunto de entrenamiento y posteriormente aplicado a los conjuntos de validación y prueba, asegurando una evaluación adecuada del rendimiento.

El proceso de modelado se desarrolló en varias iteraciones, cada una motivada por observaciones obtenidas en la etapa anterior.

Iteración 1: Regresión Logística balanceada Se entrenó un modelo de Regresión Logística incorporando ponderación de clases para mitigar el efecto del desbalance. Este modelo sirvió como línea base debido a su simplicidad, estabilidad y buen desempeño en problemas de clasificación binaria de texto.

Iteración 2: SVM lineal sin balanceo Posteriormente, se evaluó un modelo SVM lineal sin estrategias de balanceo con el fin de analizar su comportamiento base frente al desbalance del corpus. Esta iteración permitió evidenciar una baja sensibilidad hacia la clase ofensiva, confirmando la necesidad de técnicas específicas para tratar este problema.

Iteración 3: SVM lineal con balanceo de clases A partir de los resultados anteriores, se incorporó la ponderación de clases en el modelo SVM. Esta modificación produjo una mejora significativa en métricas como el recall de la clase ofensiva, justificando su uso frente a la versión sin balanceo.

Iteración 4: Optimización de hiperparámetros Finalmente, se aplicó una búsqueda exhaustiva de hiperparámetros mediante Grid Search para optimizar el parámetro de regularización del SVM. El criterio de optimización seleccionado fue el F1-score macro, con el objetivo de lograr un equilibrio entre precisión y sensibilidad en ambas clases.

El desempeño de los modelos se evaluó utilizando métricas estándar de clasificación, incluyendo precisión, recall y F1-score, así como matrices de confusión para analizar los errores cometidos. Se dio especial énfasis al recall de la clase ofensiva, dado que los falsos negativos representan un riesgo mayor en aplicaciones de moderación automática de contenido.

Además, se realizaron pruebas cualitativas mediante la predicción de frases manuales, lo que permitió verificar el comportamiento del modelo en ejemplos no vistos y evaluar su coherencia semántica.

4 Resultados

4.1 Comparativa de Rendimiento (Línea Base y Modelo Final)

Para evaluar la efectividad de nuestra propuesta, comparamos el modelo inicial (Baseline) con la versión final optimizada. La métrica principal de decisión no fue la *Exactitud* (Accuracy) global, sino la *Sensibilidad* (Recall) de la clase `Ofensivo`, dado que en un sistema de detección de odio es más crítico no dejar pasar comentarios tóxicos (falsos negativos).

cc

Métrica	**Versión 1: SVM Base**	**Versión 3: SVM Optimizado**	**Interpretación**
Accuracy Global	88%	86%	Ligera disminución al reducir sesgo.
Recall (Ofensivo)	∼0.45	0.73	Mejora notable: 62% más de verdaderos positivos.
F1-Score (Ofensivo)	0.52	0.62	Mejor equilibrio entre precisión y recall.

Table 1. Comparación de métricas entre la primera iteración y la versión final.

En la primera versión, observamos que el modelo obtenía un 88% de exactitud simplemente prediciendo que casi todo era `No Ofensivo` debido al desbalance de clases (13k muestras negativas y 2k positivas). Cuando introducimos `class_weight='balanced'` y la traducción de emojis en la versión final, se logró mejorar al modelo, elevando su capacidad de detección de 0.45 a 0.73.

4.2 Elección de mantener los emojis

La decisión de implementar la librería `emoji` para convertir iconos gráficos en texto descriptivo resultó determinante.

Evidencia de Mejora: En las pruebas cualitativas, frases como "Vete a la 🍆" (7) eran clasificadas incorrectamente como neutras por el modelo base (al eliminar el carácter). El modelo final, al procesar la entrada como "vete a la berenjena", activó correctamente la clasificación de `ofensivo`. Esto confirma que una parte significativa del lenguaje de odio en el corpus OffendES es multimodal (texto + iconos).

```
Frase: 'Me gustó👍'
Limpieza: 'me gustó pulgar_hacia_arriba'
Resultado: No Ofensivo

Frase: 'Comes 💩'
Limpieza: 'comes caca_con_ojos'
Resultado: OFENSIVO

Frase: 'Vete a la 🍆'
Limpieza: 'vete a la berenjena'
Resultado: OFENSIVO
```

Fig. 7. Ejemplos de resultados obtenidos con textos con emojis

4.3 Análisis de Errores y Limitaciones

A pesar de la optimización de hiperparámetros (`C=0.1`) encontrada mediante `GridSearch`, identificamos limitaciones en la arquitectura SVM + TF-IDF que siguieron en la versión final. Mediante pruebas manuales, categorizamos los fallos en tres tipos:

A. Ceguera a la Negación (Falso Positivo)

- **Entrada:** "No eres tonta"
- **Predicción:** OFENSIVO (Incorrecta).
- **Análisis:** Aunque se configuró el vectorizador con n-gramas (`ngram_range=(1,2)`), el modelo asignó un peso matemático excesivo al término unigramas "tonta", opacando el efecto del bigrama "no eres". Esto evidencia una limitación de los modelos de *Bolsa de Palabras* (Bag of Words) frente a la escasez de datos: si la frase exacta de negación no es frecuente en el entrenamiento, el modelo prioriza la palabra clave ofensiva.

B. Ambigüedad Semántica y Contexto

- **Entrada:** "La perra de mi primo tuvo bebés"
- **Predicción:** OFENSIVO (Incorrecta).

– **Análisis:** El modelo no logró desambiguar el término polisémico “perra” (8). Al carecer de mecanismos de atención (como los presentes en Transformers), el SVM no puede relacionar que el contexto “tuvo bebés” modifica el sentido de la palabra hacia su acepción no peyorativa.

```
Frase: 'La perra de mi primo tuvo bebés'
Limpieza: 'la perra de mi primo tuvo bebés'
Resultado: OFENSIVO

Frase: 'No eres tonta'
Limpieza: 'no eres tonta'
Resultado: OFENSIVO
```

Fig. 8. Falso positivo por ambigüedad semántica = falta de contexto

C. Jerga

– **Caso de Éxito:** Sorprendentemente, el modelo clasificó correctamente “Me das cringe”, “jaja que manco” y “eres un niño 🐭” (intentando expresar: eres un niño rata), como `ofensivo`, lo que indica que el dataset de entrenamiento contiene jerga moderna de internet lo suficientemente representada para ser vectorizada.

```
Frase: 'Me das cringe'
Limpieza: 'me das cringe'
Resultado: OFENSIVO

Frase: 'Jaja que manco'
Limpieza: 'jaja que manco'
Resultado: No Ofensivo

Frase: 'Eres un niño 🐭'
Limpieza: 'eres un niño cara_de_ratón'
Resultado: OFENSIVO
```

Fig. 9. Detección correcta de neologismos y jerga digital

4.4 Matriz de confusión

Como se observa en la Matriz de Confusión, aunque el modelo aún comete errores (Falsos Negativos), la diagonal principal muestra una concentración alta de aciertos en la clase `Ofensivo`, validando que el balanceo de pesos funcionó.

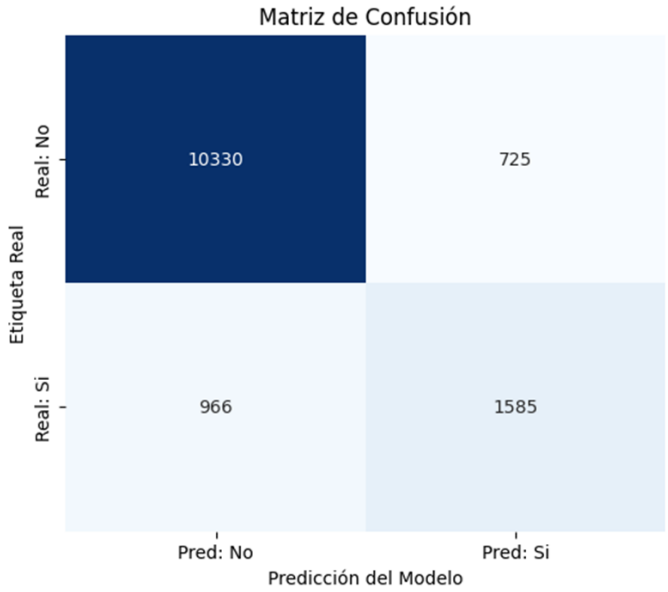

Fig. 10. Matriz de confusión del modelo final

Estos resultados nos permiten confirmar experimentalmente la teoría de la Hipótesis de Independencia de los modelos Naive Bayes y SVM lineales: al tratar las palabras como características independientes, se pierde la estructura sintáctica necesaria para detectar la negación y la ironía, como se mencionó en clase, una desventaja que teóricamente solo se resuelve con mecanismos de atención como los Transformers.

5 Apartado de Transparencia Tecnológica

Conforme a los lineamientos de ética académica del curso, el equipo declara el uso de herramientas de Inteligencia Artificial Generativa (ChatGPT-4o y Gemini) durante el desarrollo de este proyecto bajo las siguientes modalidades de asistencia:

1. **Consulta Conceptual y Técnica:** Se utilizaron modelos de lenguaje para profundizar en la comprensión teórica de algoritmos específicos (diferencias

entre *Bag-of-Words* y *Embeddings*) y para consultar documentación rápida sobre librerías de Python (`scikit-learn`, `emoji`, `pandas`).
2. **Depuración de Código:** La IA fue empleada como herramienta de apoyo para identificar y corregir errores de sintaxis y ejecución, específicamente durante la fase de carga de datos (problemas de *parseo* en archivos TSV) y en la optimización de expresiones regulares para la limpieza de texto.
3. **Refinamiento de Redacción:** Se utilizaron dichas herramientas de IA para revisar la gramática, ortografía y coherencia del presente reporte, y también para sugerir vocabulario técnico preciso en la redacción de los resultados.
4. **Generación de *Snippets* Auxiliares:** Se solicitó a la IA la generación de fragmentos de código aislados para tareas repetitivas (como la generación de gráficas con `seaborn`), los cuales fueron posteriormente revisados, integrados y validados por el equipo.

Declaración de Autoría: Nosotros pensamos que la concepción del problema, la toma de decisiones arquitectónicas, la curaduría del dataset, la ejecución de las pruebas manuales y el análisis crítico de los resultados presentados son de autoría humana propia. Las herramientas de IA no generaron el núcleo lógico del proyecto sin supervisión ni comprensión por nuestra parte.

6 Conclusiones

References

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: Mitkov, R., Angelova, G. (eds.) 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/