ISSN 1870-4069

Detección automática de palabras altisonantes en tweets utilizando redes neuronales

Ricardo Ismael Armas-Araujo[1], Yulia Ledeneva[2]

[1] Universidad Autónoma del Estado de México, México

[2] Instituto Literario, Unidad Académica Profesional Tianguistenco, México

armas5540@gmail.com, yledeneva@yahoo.com

Resumen. En la actualidad, las redes sociales y muchos medios de comunicación cuentan con problemas graves en la moderación de contenidos. La práctica más común en las redes sociales es una comunicación altisonante entre los miembros de las comunidades. Un buen manejo de y monitoreo de los comentarios en las redes sociales requiere de una herramienta actual y útil que nos permita identificar algún conjunto de las palabras más usadas en el lenguaje, para este trabajo será el lenguaje castellano y sus amplias palabras de esta índole. Para tener una gran herramienta de control de palabras altisonantes se explora como una opción viable el uso de las redes neuronales, ya que estas son capaces de aprender de un gran conjunto de datos y tener predicciones cada vez más precisas conforme avanza su entrenamiento. Esto implica mayor nivel de conocimiento de la misma red, que nos ayudara a encontrar este tipo de palabras en distintas frases. Por lo anterior mencionado, se propone en el presente trabajo una serie de pasos para la creación de una red neuronal y su entrenamiento, así como explorar las palabras utilizadas para la red neuronal.

Palabras clave: Redes neuronales, procesamiento de lenguaje natural, redes sociales, palabras altisonantes, análisis de texto.

Automatic Detection of Offensive Words in Tweets Using Neural Networks

Abstract. Currently, social media and many media outlets face serious issues in content moderation. The most common practice on social media is a profane communication among community members. Effective management and monitoring of comments on social media require a current and useful tool that allows us to identify a set of the most used words in the language; for this work, it will be the Spanish language and its extensive words of this nature. To have a great tool for controlling profane words, the use of neural networks is explored as a viable option, as they are capable of learning from a large dataset and making increasingly precise predictions as their training progresses. This implies a higher level of knowledge of the network itself, which will help us find this type of words in different phrases. Therefore, the present work proposes a series of steps

for the creation and training of a neural network, as well as exploring the words used for the neural network.

Keywords: Neural networks, natural language processing, social networks, profane words, text analysis.

1. Introducción

Hoy en día el área de Procesamiento de Lenguaje Natural (PNL) es de gran importancia ya que gracias a esta podemos interactuar entre las computadoras y el lenguaje humano de una manera más amigable. Actualmente, en las empresas se utilizan algunas de las técnicas de PLN en diferentes ámbitos que nos ofrecen una amplia forma de interacción humano-computadora a través del lenguaje, proporcionando así las bases para una mejor comprensión de este contexto. Uno de los puntos clave dentro del PLN son los modelos, como el modelo de representación de palabras, ya que estos nos ayudan a entender el significado de las palabras en su contexto.

Por lo tanto, es indispensable contar con un modelo de representación o detección de palabras, ya que gracias a este se pueden identificar palabras clave en textos largos. De igual manera se utiliza la tokenización para la separación de las oraciones. La aplicación directa de esta investigación pretende mejorar la moderación de las palabras altisonantes en contenido de redes sociales. Cuando se desarrolla un sistema robusto para el reconocimiento de este tipo de palabras se contribuye a la creación de entornos en línea más seguros y civilizados. Los comentarios dentro de la sociedad en muchas ocasiones son mal usados en diferentes contextos. En este artículo se pretende identificar de palabras altisonantes de un corpus de tweets, esto se realizará por el mal uso de estas palabras en la vida cotidiana de las personas.

Sin embargo, nosotros podemos detectarlas antes de que se hagan tweets o publicaciones de odio hacia cualquier persona, evitando de esta manera una discusión por estos puntos. Además, en [1] se menciona que en México el albur se utiliza en todas partes, ya que muchas palabras tienen doble sentido y la combinación de verbos sustantivos, como coincide [9] en la riqueza de este lenguaje. La organización de este trabajo se estructura en secciones que abarcan desde la primer sección de este trabajo, que es la introducción donde se explicó el objetivo de esta investigación; la segunda sección de trabajos relacionados es donde se mencionan trabajos similares que abordan esta misma problemática; en la tercer sección de artefactos propuestos se menciona como se abordó el problema y que metodologías se utilizaron para ello; la cuarta sección menciona los resultados de la investigación y conclusiones generales.

2. Trabajos relacionados

El PLN, como mencionan algunos autores en [10], involucra la detección del análisis semántico en textos, es decir, la interpretación del significado de estos textos.

Esta detección también se enfoca en la interacción entre el lenguaje humano y la computadora, ya que es la estrategia más básica para la detección de palabras en frases, como refiere [3]. La tokenización es un método que facilita la separación de las palabras

Fig. 1. Metodología de una red neuronal.

dentro de los textos [9]. Esta técnica, al hacer esta separación, permite la identificación efectiva y clara de las palabras en las oraciones, según lo mencionado en [5]. Necesitamos de técnicas como la tokenización porque no podemos manejar manualmente todo el contenido dentro de las redes sociales [6].

Teniendo en cuenta este notorio problema, debemos destacar que la tarea puede complicarse por los diferentes significados que algunas palabras clave pueden tener cuando se utilizan en distintos contextos [7], ya que, como se menciona en [1], en el contexto mexicano el albur siempre está presente con un doble sentido, añadiendo más complejidad a la detección de estos mensajes o palabras concretas. Existen muchos modelos para la identificación de palabras en conjuntos de datos.

En [4] se menciona que actualmente algunos autores han utilizado modelos de regresión logística para la detección de estas palabras, aunque con las nuevas tecnologías podemos optar por el uso de redes neuronales convolucionales (CNN). Por otro lado, como se menciona en [8], en la última década se han realizado campañas de evaluación del procesamiento de lenguaje natural y sus enfoques, donde destacan la importancia de la detección de mensajes de odio o altisonantes, ya que estos pueden ayudar a detectar problemas antes de que ocurran.

En el trabajo [2], para el conjunto de datos sobre misoginia que utilizaron, las redes neuronales convolucionales mostraron los mejores resultados en la detección de misoginia, superando a otros modelos de procesamiento de lenguaje en velocidad y eficacia en la detección del lenguaje ofensivo.

3. Metodología propuesta

Se proponen distintos modelos para la detección de las palabras clave en los enunciados que recibimos día con día en nuestras redes sociales. Sin embargo, a menudo no podemos centralizar de manera efectiva cómo nuestros mensajes pueden llegar a ser ofensivos. Por esta razón, se propone el siguiente diagrama de metodología para el desarrollo de una red neuronal convolucional (CNN) que constará de 4 capas.

La primera capa tendrá 1000 neuronas, la segunda no utiliza neuronas, ya que es una operación de reducción de dimensionalidad que calcula el promedio de los valores de características para cada ventana de la secuencia, la tercera contará con 24 neuronas y utilizará una función de activación ReLU, y la última capa estará compuesta por una única neurona, sumando un total de 1025 neuronas. El diseño a continuación (ver Figura 1) nos permitirá identificar eficazmente estas palabras clave.

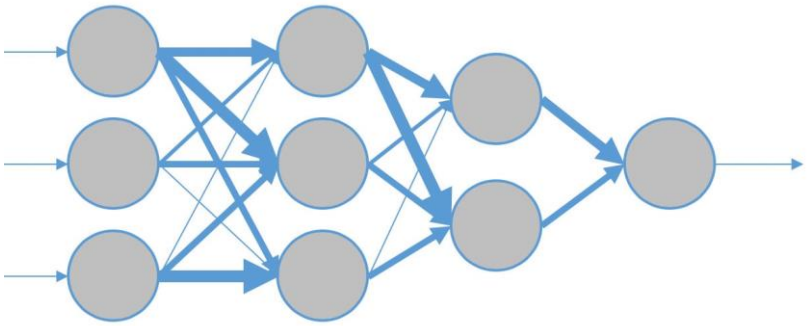

Fig. 2. Modelo de red neuronal.

3.1. Conjunto de datos

En nuestro conjunto de datos, disponemos de una lista de palabras altisonantes que ya han sido extraídas de textos y clasificadas como términos ofensivos o expresiones ofensivas. En este caso, cargaremos solo los términos ofensivos para analizar si la red neuronal puede identificar dichos términos en las frases que le proporcionemos. Además, crearemos algunos datos de ejemplo que contienen frases con términos ofensivos y otras sin ellos.

Clasificaremos las frases como 0 para no ofensivas y 1 para ofensivas. Esto nos permitirá tener un mejor rendimiento al momento de entrenar nuestra red neuronal. Para el análisis, debemos tener en cuenta cómo está estructurado el conjunto de palabras ofensivas en nuestro archivo, que se representa de la siguiente manera en un formato donde los datos se pueden extraer de mejor manera (ver Tabla 1).

3.2. Preprocesamiento de datos

Una vez cargados los datos, procederemos a su manipulación y limpieza. En primer lugar, realizaremos una tokenización. Configuraremos un tokenizador para convertir nuestro texto en secuencias de enteros, donde cada entero representa una palabra en un diccionario que definiremos con un máximo de 1000 palabras.

Finalmente, utilizaremos secuencias que se normalizarán a una longitud fija de 10, rellenando con ceros al final si son más cortas. La tokenización es una de las partes cruciales del procesamiento de datos [10] y para la red neuronal propuesta. Por lo tanto, un buen preprocesamiento de los datos nos permitirá tener una mayor precisión al momento de crear nuestro modelo de red neuronal.

3.3. Construcción del modelo de red neuronal

Una red neuronal tiene la característica de simular o imitar el proceso de aprendizaje del ser humano [11]. Este proceso se centra en la construcción de un modelo con neuronas unidas por una serie de caminos. Se seleccionó este modelo debido a su notable eficacia mencionada en trabajos relacionados.

Tabla 1. Formato de términos ofensivos.

Columna 1	Columna 2	Columna 3
T1	TERMINO_OFENSIVO	tonto
T2	TERMINO_OFENSIVO	hipócrita

Para la metodología propuesta, utilizaremos 4 capas en la red neuronal, como se muestra a continuación: Para la construcción de nuestro modelo, primeramente, debemos utilizar una capa de embedding, que se define para lograr un mejor entrenamiento de las redes neuronales [12]. En términos más explícitos, es una capa que convierte los índices de palabras en vectores densos de tamaño fijo.

De igual manera, utilizamos una capa de pooling para reducir la dimensionalidad promedio de las características. Al final, utilizamos capas densas (dense) para la clasificación y la salida binaria, donde 0 indica no ofensivo y 1 indica ofensivo. Para esta red neuronal, utilizaremos un total de 4 capas, como se mencionó anteriormente (ver Figura 2).

Como se observa en la imagen, las 4 capas nos permitirán manipular de mejor manera las palabras que se deben detectar. En la primera capa, se ingresan los textos tokenizados para que la capa de embedding convierta las palabras en vectores. Posteriormente, la segunda capa realizará el pooling para reducir la dimensionalidad de los vectores de características. La tercera capa transformará los vectores mediante una función de activación ReLU. Finalmente, la última capa determinará la clasificación como ofensiva o no ofensiva.

3.4. Entrenamiento del modelo

En este apartado, nuestro modelo comienza a utilizar todo lo realizado anteriormente para un buen entrenamiento. Un buen entrenamiento debe contar con una cantidad suficiente de datos que nos permita analizar varios puntos de vista. Existen distintos caminos para lograr el objetivo, que en este caso es la identificación de las palabras altisonantes.

En nuestro caso, pasamos los datos al modelo e indicamos cuántas iteraciones debe realizar. El objetivo es que en cada iteración la precisión en la identificación de las palabras mejore progresivamente. Además, definimos una función dentro de un condicional (if) con el conjunto de palabras altisonantes. Esta función se encargará de identificar si alguna palabra altisonante de la lista se encuentra dentro del mensaje enviado por el usuario.

3.5. Predicción del modelo

La última parte es la predicción, donde verificamos si se encuentra un término ofensivo. Mandamos un mensaje que teclea el usuario y este se compara con la lista de palabras altisonantes. Esto se logra convirtiendo el texto obtenido del usuario a una secuencia y luego a un vector, de la misma forma que los datos de entrenamiento. Finalmente, se realiza la predicción. Utilizaremos 10 épocas para la detección de las palabras altisonantes, como se mencionó anteriormente.

Tabla 2. Datos resultantes ofensivos obtenidos con la red neuronal.

Época	Pérdida	Exactitud
1	0.6935	0.5200
2	0.6923	0.6200
3	0.6912	0.6600
4	0.6903	0.6800
5	0.6893	0.7000
6	0.6883	0.7000
7	0.6872	0.7200
8	0.6960	0.7600
9	0.6847	0.8200
10	0.6833	0.8400

Esto se hace con la finalidad de que el modelo de red neuronal tenga más datos para reducir la pérdida y mejorar su exactitud al momento de predecir una oración con palabras altisonantes.

4. Experimentación y resultados

Una vez realizados todos los puntos de la metodología, se efectúan iteraciones en la red neuronal e identificamos problemas que surgen al aplicarla, especialmente cuando una palabra ingresada tiene algún carácter especial o alguna característica que modifica la palabra. Esto puede influir en la identificación de un término ofensivo, ya que las palabras usadas en el modelo no incluyen caracteres especiales. Otro problema es la clasificación de cada palabra; dado que el corpus para nuestro estudio también incluye expresiones ofensivas, es crucial dividir estas adecuadamente para evitar clasificaciones erróneas y, por ende, identificaciones incorrectas por parte del modelo.

Como se observa en la tabla anterior (Tabla 2), podemos denotar como en cada iteración la exactitud (accuracy) va en aumentos lo que significa que se está reconocimiento de mejor manera las palabras en una posible frase, en este caso en particular podemos observar cómo se manda una frase con algunos términos ofensivos y como se realiza el hallazgo de ciertos términos.

En este otro caso (Tabla 3), podemos observar cómo se le llega a mandar una frase que no resulta tener alguna de estos términos ofensivos, lo que se observa por las métricas de perdida y precisión del modelo, así como el resultado final que dentro de la ejecución del programa se manda un mensaje donde se menciona que no se llegan a determinar palabras obscenas dentro de nuestro mensaje.

Del conjunto de palabras altisonantes se determina cuáles son las principales palabras que se llegan a repetir o utilizar más en nuestro conjunto de datos: La gráfica anterior (Figura 3) muestra que la mayor palabra que se encontró es mierda con 1480 repeticiones, puto en segundo lugar con 804, puta con 706 repeticiones, pringada con 440 repeticiones, gorda con 336 repeticiones, coño con 331 repeticiones. Una vez vista

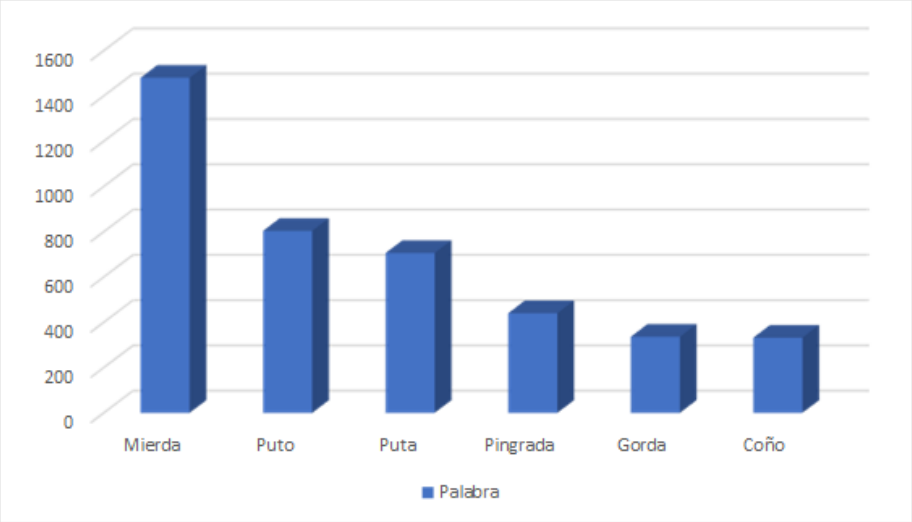

Fig. 3. Palabras más usadas en el conjunto de datos.

la forma en que nos arrojó las palabras podemos determinar que el uso de una red neuronal nos ayudó a realizar la detección de manera precisa y con un tiempo no tan excesivo; contamos con que la red neuronal tarde alrededor de 5 minutos en el entrenamiento de las épocas y notamos que los resultados fueron favorables, la red detecta de buena manera las palabras en el corpus, el único inconveniente presente es que si nosotros utilizamos palabras que se encuentran fuera del corpus o con modificaciones de doble sentido como puede ser mi3rda, utilizando un número en lugar de una letra, el modelo no lo llega a detectar; para mejores resultados podemos expandir el corpus y complementar el idioma castellano para la mejora de detección.

4.1. Parámetros de la red neuronal

Para la red neuronal propuesta se comenzó con un proceso de tokenización, que utiliza Keras como tokenizador, este a su vez fue configura con un máximo de 1000 palabras basadas en la frecuencia de aparición, lo que indica que solo las mil palabras más frecuentes se usaran para el entrenamiento de nuestra red, y cualquier otra palabra se marcara como "<OOV>" que representa las palabras fuera del vocabulario. Para las secuencias se hace un truncado a rellenado para asegurar que tengan longitud uniforme de 10 palabras, esto es necesario para que el modelo procese los datos de entrada de una manera más consistente, ya que las entradas de las redes neuronales deben ser del mismo tamaño.

Para la arquitectura de la red neuronal, se comienza con una capa Embedding que transforma los índices en las palabras de vectores densos de 16 dimensiones; la primera capa nos ayuda a que el modelo pueda aprender representaciones diversas y útiles basadas en su contexto; por otro lado, la segunda capa es una `GlobalAveragePooling1D´ que nos ayuda a reducir la dimensionalidad promedio de las incrustaciones dentro de la secuencia, para simplificar nuestra red. De igual manera,

Tabla 3. Datos resultantes no ofensivos obtenidos con la red neuronal.

Época	Pérdida	Exactitud
1	0.6919	0.5400
2	0.6899	0.6400
3	0.6881	0.8200
4	0.6863	0.8600
5	0.6842	0.9600
6	0.6819	0.9800
7	0.6798	0.9600
8	0.6772	0.9400
9	0.6747	0.9800
10	0.6719	0.9800

se introducen otras dos capas densas conectadas entre sí, la primera de 24 nodos con una función ReLU, que nos ayuda a introducir no linealidades en el modelo, aprendiendo así patrones más complejos, y la segunda capa que es densa con un solo nodo y una función de activación sigmoidea, comúnmente utilizada en problemas de clasificación binaria como en nuestro caso si un texto es ofensivo o no.

Para la compilación del modelo, este compila una función de perdida 'binary_crossentropy', la cual es eficiente para comparar las salidas de la función sigmoidea con las etiquetas binarias en los datos del entrenamiento; también se hace uso de un optimizador Adam, indispensable por su eficacia y ajuste automático del ritmo en que se da el aprendizaje, por último, la exactitud se usa para monitorear el rendimiento del modelo en el entrenamiento.

Finalmente, nuestro modelo se entrena por medio de 10 épocas, haciendo que las representaciones aprendidas por las iteraciones minimicen la perdida y de una mejora en la precisión de la detección de las palabras altisonantes según su contenido ofensivo. El enfoque que se presenta combina técnicas de procesamiento de lenguaje natural con aprendizaje profundo, ofreciendo un modelo que sea capaz de identificar y detectar palabras ofensivas basadas en características lingüísticas especiales.

4.2. Descripción de corpus

OffendES_spans es un corpus en español creado a partir del corpus OffendES, con la identificación automática de términos ofensivos utilizando el lexicón SHARE [13]. El corpus consta de 47.128 comentarios anotados con términos y expresiones ofensivos. Los comentarios fueron anotados de manera manual utilizando un esquema de anotación detallado. Los comentarios fueron recopilados de diferentes redes sociales: Twitter, Instagram y YouTube. Los comentarios publicados fueron ofensivos o hirieron los sentimientos de otros usuarios según su género, raza, religión, ideología u otras características personales.

Tabla 4. Presencia de términos ofensivos de léxicos en los comentarios recuperados.

Red Social	Término ofensivo	Término no ofensivo	Total
YouTube	19,449	184,414	203,863
Instagram	3,142	58,209	61,351
Twitter	1,197	18,728	19,925
Total	23,788	259,865	283,622

Primero, se recolectaron un total de 283.622 comentarios (ver la Tabla 4). Luego, los comentarios se filtraron según dos principales limitaciones: la presencia de lenguaje potencialmente ofensivo y diversidad léxica. Para evitar la creación de un corpus con pocos o ningún comentario ofensivo, se etiquetaron todos los comentarios con banderas que determinaban si el comentario contenía alguna de las palabras encontradas en cinco léxicos controlados diferentes [14]. Se seleccionaron todos los comentarios con lenguaje potencialmente ofensivo (23.788 comentarios).

5. Conclusiones y trabajo futuro

Como llegamos a ver los entrenamientos de redes neuronales no son una tarea fácil, ya que implica un buen entrenamiento, creación y carga de la misma red neuronal, de igual manera debemos considerar nuestros datos, ya que por medio de estos nuestra red tendrá un cierto porcentaje de error o asertividad.

Como todo lo nuevo si no llegamos a profundizar en el tema tendremos problemas al momento de realizar un programa de esta índole, una vez conociendo el entorno y los métodos básicos utilizados como la tokenización, podemos avanzar en los modelos de redes sin mayor complejidad, lo que de igual manera tendrá un impacto en los resultados.

La propuesta de red neuronal explorada nos ayuda a comprender la identificación de las palabras altisonantes castellanas en una frase enviada por el usuario, algunos aspectos de las palabras pueden hacer que el modelo las identifique o no, sin embargo, con un conjunto de palabras con acentos o algún otro carácter especial contenida dentro de esta no tendrá mayor problema de identificación.

Cada una de las fases exploradas corresponde a puntos cruciales en el diseño de una red neuronal, la fase donde procesamos los datos es una de las más importante de todas las vistas, ya que en ella se puede visualizar la subida de los datos para nuestra red neuronal y si se llegan a presentar fallos puede comprometer nuestro modelo y sus parámetros.

Como trabajo a futuro podemos destacar lo anteriormente mencionado un conjunto de datos con distintos tipos de caracteres especiales puede ser un reto para identificar más palabras, ya que muchas veces la gente cambia incluso una letra por un número para que las palabras sean difíciles de identificar para los algoritmos como el presentado en este trabajo. Si contemplamos otros tipos de modelos y datos especiales para la red neuronal propuesta podremos mejorar algoritmos como estos para que identifiquen palabras como las ya mencionadas.

Referencias

1. Guzmán, E., Beltrán, B., Tovar, M.: Clasificación de frases obscenas o vulgares dentro de tweets. Research in Computing Science, vol. 85, pp. 65–74 (2014)
2. De la Peña, G.: Análisis y detección de odio en mensajes de Tweeter. Universitat Politècnica de València, pp. 11–14 (2019)
3. De la Peña, G.: Deep Analyzer at SemEval-2019 Task 6: A deep learning-based ensemble method for identifying offensive tweets. In: Proceedings of the 13th International Workshop on Semantic Evaluation, pp. 582–586 (2019). DOI: 10.18653/v1/S19-2104.
4. Shushkevich, E.: Automatic misogyny detection in social media: A survey. Computación y sistemas, vol. 23, no. 4, pp. 1159–1164 (2021). DOI: 10.13053/cys-23-4-3299.
5. Clarke, I., Grieve, D.J.: Dimensions of abusive language on twitter. In: Proceedings of the First Workshop on Abusive Language Online, pp. 1–10 (2017). DOI: 10.18653/v1/w17-3001.
6. Ahluwalia, R. Shcherbinina, E., Callow, E., Nascimento, A.C., De-Cock, M: Detecting misogynous Tweets. University of Washington, pp. 1–7 (2018)
7. Canós, J.S.: Misogyny identification through SVM at IberEval 2018. Universidad Politécnica de Valencia, pp. 229–233 (2018)
8. Frenda, S., Ghanem, B., Montes-y-Gómez, M.: Exploration of Misogyny in Spanish and English tweets. In: Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval´18), colocated with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN´18), vol. 2150, pp. 260–267 (2018)
9. Ramos, O.: Análisis sobre el idioma español en México, con base en la frecuencia de palabras azules rojas y obscenas y vulgares en Twitter. Universidad de Puebla, pp. 1–8 (2016)
10. Pérez, A.: Detección del discurso de odio de twitter. Universidad Politécnica de Valencia, pp. 27–33 (2020)
11. Castaneda-Sanchez, W.A., Polo-Escobar, B.R., Vega-Huincho, F.: Artificial neural networks: A measurement of forecast learnings as potential demand. Universidad Ciencia y Tecnología, vol. 27, no. 118, pp. 51–60 (2023). DOI: 10.47460/uct.v27i118.686.
12. López, D.: Aprendizaje profundo para la extracción de aspectos en opiniones textuales. Revista Cubana de Ciencias Informáticas, vol. 13, no. 2, pp. 105–145 (2019)
13. Plaza-del-Arco, F.M., Montejo-Ráez, A., Ureña-López, L.A., Martín-Valdivia, M.T.: OffendES: A new corpus in spanish for offensive language research. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), pp. 1096–1108 (2021)
14. Plaza-del-Arco, F.M., Molina-González, M.D., Ureña-López, A., Martin, M.: Sinai at semeval-2020 task 12: Offensive language identification exploring transfer learning models. In: Proceedings of the Fourteenth Workshop on Semantic Evaluation, pp. 1622–1627 (2020). DOI: 10.18653/v1/2020.semeval-1.211.