Análisis de Sentimiento aplicado a la detección de Ciberacoso en YouTube como Red Social

Álvaro F. Llampa[1], José H. Farfán[1], Mariela E. Rodríguez[1]

[1] Facultad de Ingeniería - UNJu,
San Salvador de Jujuy, Jujuy 4600, Argentina
{a.facundollampa, jhfarfan, maru972}@gmail.com

Resumen. En un mundo en el que las personas están constantemente en comunicación sin importar las barreras de distancia o idioma gracias al internet, las redes sociales han significado una gran oportunidad para transformar la interacción humana posibilitando la creación de comunidades en línea donde se puede compartir conocimiento o simplemente generar contenido para entretenimiento. YouTube es un atractivo para las personas, más aún los jóvenes, por el sueño de poder generar ingresos monetarios al realizar videos que atraigan visitas. Sin embargo, esta situación también da lugar a personas con malas intenciones. El ciberacoso es una preocupación constante debido a los efectos negativos que pueden causar en los afectados como depresión, baja autoestima, dificultad para relacionarse, etc. Afortunadamente existen herramientas con las que se puede colaborar a su detección para poder tomar acción. El trabajo realizado en el campo del procesamiento del lenguaje natural es amplio en el idioma inglés y no tanto en español, por esto, se busca que con el análisis de sentimientos junto a un algoritmo de *machine learning* trabajar en un proceso para obtener un modelo automático que dé resultados efectivos en español. Se diseñarán etapas de tratamiento de texto ajustadas al contexto, se seleccionará y entrenará el modelo de aprendizaje supervisado *Support Vector Machine* y que el resultado final pueda proveer datos valiosos para el análisis de cualquier video de la plataforma YouTube.

Palabras Clave: Cyberbullying, Text Mining, Sentiment Analysis, Support Vector Machine.

1 Introducción

Las redes sociales van más allá de los métodos digitales tradicionales de comunicación (correo electrónico, blogs, foros, etc.) en los cuales las personas interactúan y comparten experiencias y opiniones. Los beneficios son muchos, pero también es importante contemplar los riesgos, principalmente en adolescentes, que hacen uso de estas plataformas para entretenimiento y sociabilización. Un peligro que lamentablemente está presente es el ciberacoso o *cyberbullying* en inglés, una actividad dañina en la cual usuarios realizan acciones contra otros, tales como como amenazas, acoso, chantaje, etc. Los casos más comunes se dan en adolescentes y jóvenes adultos. De acuerdo con una investigación llevada a cabo por Unicef llamada “Global Kids Online 2015-2016”

[1] se indica que el 78% de los adolescentes sufren estas experiencias negativas y que en el 68% de los casos la familia conoce poco o nada de la situación, destacándose que los adolescentes priorizan buscar ayuda entre pares (otros adolescentes) en lugar de comunicarse con adultos.

Entonces, este trabajo se centra en el estudio e implementación de un conjunto de procesos que permitan determinar el sentimiento o estado del ánimo extraído de los comentarios de cualquier video que se encuentre publicado en la red social YouTube.

1.1 Contexto

Este trabajo forma parte de la etapa final del proyecto de investigación titulado "Análisis Inteligente de Técnicas de Minería de Texto e Implementación para la Detección de Ciberacoso en Redes Sociales", comprendido en la Beca EVC-CIN 2.018 otorgada por el Consejo Interuniversitario Nacional al estudiante Álvaro Llampa (quien actualmente es ingeniero en informática). Además, el presente trabajo se encuentra también incluido en el Grupo de Investigación del proyecto denominado "Minería de Textos aplicada a problemas de ingeniería" de la Facultad de Ingeniería de la Universidad Nacional de Jujuy, donde es importante mencionar, que los autores del presente paper forman parte de dicho grupo de investigación.

2 Minería de Texto y sus aplicaciones

La minería de texto o *Text Mining* es un campo amplio con aplicaciones como Detección de Correos Spam (*Spam Detection*), Síntesis de Documentos (*Text Summarizer*), Generación de Texto *(Text Generation*) o Análisis de Sentimiento (*Sentiment Analysis*) entre otros [2]. En el caso particular de este trabajo, el foco de atención estará en el análisis de sentimiento y cómo esta puede ayudar a la detección de ciberacoso o *cyberbullying.*

El análisis de sentimiento se basa en la extracción y clasificación del sentimiento comúnmente positivo o negativo sobre un tema en particular. Su uso es común por ejemplo en empresas donde buscan descubrir cual es la opinión de las personas sobre su producto o servicio a través de encuestas. Inicialmente es una forma controlada de obtener respuestas claras y directas. También existe otra forma muy utilizada de saber la opinión de las personas y es aplicarlo en las redes sociales. Hoy en día estas plataformas de comunicación con alcance mundial son consideradas las fuentes de opinión más extensas y variadas. Sin embargo, también son fuente de mucho contenido de poco valor o irrelevante por lo que se debe tratar con cuidado todo dato extraído de dicha fuente.

En el caso particular de este trabajo, el enfoque se aplica en analizar los comentarios de usuarios de la plataforma de YouTube. Inicialmente se elabora un dataset para el entrenamiento y evaluación de un algoritmo de aprendizaje supervisado y posteriormente se incluye un video al cual se le extraen los comentarios para su tratamiento,

clasificación y análisis. Los videos en esta plataforma son publicados constantemente y lamentablemente se ha vuelto moneda corriente la actividad de realizar comentarios con mensajes violentos, amenazadores, etc. Un nombre común para referirse a las personas que realizan esta actividad es el de "*hater*" que proviene de la palabra inglesa "*hate*" que se traduce como "odio". Actualmente las redes sociales buscan mitigar este comportamiento ofensivo, pero no se logra en un 100%, surgiendo en consecuencia la necesidad de conocer hasta donde llegaran los usuarios con sus opiniones, para que sus comentarios, sean positivos o negativos, sean escuchados.

2.1 Objetivo

El fin principal es identificar y analizar el sentimiento plasmado en comentarios a través de su clasificación y análisis utilizando un algoritmo de aprendizaje supervisado.

2.2 Objetivos Específicos

- Obtener y Procesar comentarios para concluir con un dataset balanceado.
- Determinar una secuencia de etapas que cumplan con la limpieza y normalización de comentarios.
- Seleccionar, entrenar y evaluar un modelo de aprendizaje supervisado.
- Obtener y analizar el resultado al procesamiento de los comentarios de un video publicado en la plataforma YouTube.

3 Obtención y Preprocesamiento de Comentarios

Este trabajo es implementado con el lenguaje de programación Python junto a librerías como ***spaCy***, ***scikit-learn*** y ***matplotlib***, que son conocidas por su gran uso en el tratamiento, análisis y visualización de datos. Además, los comentarios de la plataforma YouTube se obtienen a través de *YouTube Data API*, la cual Google provee como servicio y así, mediante peticiones *http* recuperar comentarios de un determinado video subido en dicha plataforma e identificado a través de su *id* (identificador).

3.1 Preparación del DataSet

Para determinar la polaridad o sentimiento de los comentarios se hace uso de un modelo de aprendizaje supervisado, entonces es necesario que se prepare un conjunto de datos inicial etiquetados (dataset) y que se separen en conjuntos de entrenamiento y posteriormente pruebas.

Al solicitar datos a la API de YouTube, las respuestas se realizan en formato JSON[1], las cuales deben ser *parseadas*[2] para luego recuperar los comentarios, los cuales son los elementos de interés. Estos textos pueden contener *emojis*, *timestamps* del propio video, enlaces a páginas externas u otros videos, errores de ortografía; todo esto debe ser tratado para que de esta forma se obtenga un conjunto de datos aptos para ser trabajados en el procesamiento de comentarios.

Una cuestión delicada respecto a la preparación de un dataset es que, para el contexto en el cual se trabaja, el fin perseguido tiene gran peso y por ello no es posible, en este caso, que se eliminen o corrijan todo tipo de anomalías de un comentario porque existen detalles o expresiones que son característicos de la forma de expresarse de las personas, por ejemplo, términos como "groso" , "capo" o "copado" son modismos muy utilizados en Argentina y por lo tanto son elementos de valor al momento de analizar los comentarios.

Se realizó una primera etapa de limpieza donde se involucran procedimientos implementados en código para remover elementos mencionados anteriormente y un tratamiento manual para tratar con errores de ortografía y eliminación de todos signos de puntuación enfocándonos solamente en las palabras, además todo el texto fue llevado a minúsculas, lo cual es un paso importante para homogeneizar el conjunto de comentarios. Finalmente, se culmina con un dataset que se compone de 1000 comentarios etiquetados como positivos y 1000 comentarios etiquetados como negativos, todos almacenados en un documento con extensión *csv* para su posterior tratamiento.

3.2 Vectorización de Palabras

Para poder realizar cualquier tipo de análisis estadístico o entrenar un modelo es necesario transformar los datos representados en textos a un equivalente numérico que pueda ser entendido o tratado por una máquina, este procedimiento es comúnmente conocido como *feature extraction*. En este caso la representación intermedia es el proceso de **vectorización de palabras** (*Word Vectorization* o *Word Embeddings*) en donde cada comentario es representado por un vector cuyo estado es apto para poder alimentar cualquier modelo de *machine learning* [3]. Para aplicar esta transformación existen distintas técnicas.

BoW (*Bag of Words*) o **Bolsa de Palabras**. Con esta técnica se construye un diccionario que contiene todas las palabras individuales del conjunto de documentos original [4]. Los llamados documentos son los comentarios dentro del contexto de este trabajo. Se ilustra de forma sencilla un ejemplo que tiene los siguientes comentarios:

"yo pienso que eres una persona maravillosa"
"me gusta el video y eres una buena persona"

[1] JSON (*JavaScript Object Notation*) es un formato basado en texto estándar para representar datos estructurados en la sintaxis de objetos de JavaScript.

[2] Parsear es la acción de analizar un texto y determinar si cumple con una estructura o sintaxis.

"me encanta tu actitud"

El diccionario resultante es:

{yo, pienso, que, eres, una, persona, maravillosa, me, gusta, el, video, y, buena, encanta, tu, actitud}

Un comentario como "*me encanta el video y me gusta tu actitud*" transformado a vector seria:

[0, 0, 0, 0, 0, 0, 0, 2, 1, 1, 1, 1, 0, 1, 1, 1]

Como se puede ver el peso de cada palabra es más alto mientras más se repita, con esta técnica las palabras con un mayor peso serían artículos, preposiciones o pronombres, lo cual supone una desventaja importante debido a que son elementos poco relevantes al fin perseguido. Por esto, la alternativa elegida para este trabajo es una técnica llamada **TF-IDF** (*Term Frequency – Inverse Document Frequency*), una característica que destaca a este método es que incrementa el peso de las palabras que son poco utilizadas y disminuye el peso de las palabras que ocurren con mucha frecuencia [5].

Dentro de la vectorización se encuentra un proceso de *Tokenization* o **Tokenización** que consiste en separar cadenas de texto en elementos individuales llamados *tokens*, por ejemplo, cada palabra de un comentario es un *token*. También es posible definir como *tokens* a oraciones dentro de un párrafo.
Para realizar esta transformación a vectores, existe una librería llamada ***scikit-learn*** que provee una función llamada *TfidfVectorizer*, más adelante se detalla cómo se la utiliza. Así, es que con esta etapa es posible pasar a entrenar un modelo, pero aún es posible realizar otros tratamientos con el fin de mejorar los resultados del entrenamiento.

3.3 Proceso de Refinamiento

Como se mencionó antes, existen ciertas palabras que son muy frecuentes en un texto como pronombres, preposiciones o artículos, adverbios que toman el nombre de *Stop Words* o **Palabras Vacías** debido a que se considera que no aportan significado por sí solas o en su procesamiento [6], además, al remover este tipo de elementos se colabora en que la etapa de entrenamiento se enfoque solo en palabras que se consideren importantes.

Otro proceso recomendado es aplicar *Lemmatization* o **Lematización** y consiste en reducir la palabra a su lema, forma canónica o también llamada forma de diccionario [7]. En otras palabras, esto quiere decir que se las lleva a su unidad semántica y que es posible encontrarla en el diccionario. Por ejemplo, tenemos la siguiente secuencia de palabras: "*cantas, canta, cantamos, cantaré, cantan*". Todas son conjugaciones del verbo cantar, se separa al texto en tokens y se obtiene el lema de cada uno dando como resultado:

['*cantar*', '*cantar*', '*cantar*', '*cantar*', '*cantar*']

Nótese que las palabras encerradas entre corchetes y separadas por coma forman una lista. El beneficio principal es que al vectorizar los comentarios con los lemas de los tokens y no sus conjugaciones o variaciones, se reduce la dimensionalidad del vector resultante. Siguiendo el ejemplo anterior, la cantidad de palabras a procesar pasan de ser 5 (conjugaciones del verbo *cantar*) a una sola, *cantar*.

Una vez descritos estos procesos adicionales es recomendado optar por remover *Palabras Vacías* en casos de clasificación de textos [8] y ***spaCy***, una librería ampliamente utilizada en *NLP* (Procesamiento Natural del Lenguaje) tiene incorporada una lista de ellas. Esta lista es modificable eliminando o agregando palabras según el contexto lo requiera.

Al trabajar con ***spaCy*** se descubrieron errores al encontrar el lema de algunas palabras en español por lo que se buscaron alternativas. La universidad de Stanford tiene un grupo llamado *The Standford NLP Group* [9] donde desarrollaron una librería llamada ***Stanza*** para el procesamiento de lenguaje natural en Python. ***spaCy*** permite acoplarse con esta librería a través de un paquete llamado ***spacy-stanza***. De esta forma es que al utilizar el modelo de ***Stanza*** en español se logran mejoras en cuanto a la etapa de lematización.

Además, se hace uso de la lista de palabras vacías que provee. En la figura 1 se muestran los primeros 10 elementos de un conjunto en total de 551 palabras.

```
{'actualmente',
 'acuerdo',
 'adelante',
 'ademas',
 'además',
 'adrede',
 'afirmó',
 'agregó',
 'ahi',
 'ahora',
```

Fig. 1. Primeras 10 Palabras Vacías de spacy-stanza.

Al definir la función *TfidfVectorizer* se puede modificar un parámetro de configuración llamado *tokenizer* [10]. En este parámetro se indica la función que realiza este procesamiento de refinamiento descrito anteriormente. De esta forma, con estas técnicas aplicadas como parte del preprocesamiento es que se puede abordar el estudio del modelo a entrenar.

4 Clasificador y Analizador de Sentimientos

Es una práctica común separar el conjunto de datos respetando una proporción del 70% al 80% para el subconjunto de entrenamiento y del 20% al 30% para evaluación [11]. Para este trabajo, el dataset se separa en una proporción de 70% para entrenamiento y

30% para evaluación, es importante recordar que existe el mismo número de observaciones para comentarios positivos y negativos, por lo tanto, es un conjunto balanceado y se tuvo especial cuidado en que las proporciones también lo sean.

4.1 Entrenamiento y Análisis del Modelo de Aprendizaje Supervisado

Existe una gran variedad de algoritmos de aprendizaje supervisado, para este caso en particular que se trata de un problema de clasificación binaria (comentarios positivos y negativos) y se escogió *Support Vector Machine* (SVM) como algoritmo clasificador. Considera características particulares de los textos como Espacio de características de alta dimensionalidad (*High Dimensionality Feature Space*) que viene dado por la gran cantidad de palabras que se ven involucradas entre en el total de cuerpos de texto o Alto Grado de Redundancia (*High Leve lof Redundancy*) que indica que existen muchas características relevantes y frecuentemente en del mismo documento aun después de haber realizado tratamientos previos como lematización, y eliminación de palabras vacías [12]. Una implementación de SVM se encuentra en la librería ***scikit-learn*** y se llama *Support Vector Classifier* (SVC) que es el modelo a entrenar con los comentarios.

En la figura 2 se muestra un flujo de procesos que se realiza desde la obtención de los comentarios hasta el entrenamiento y evaluación del modelo de aprendizaje supervisado.

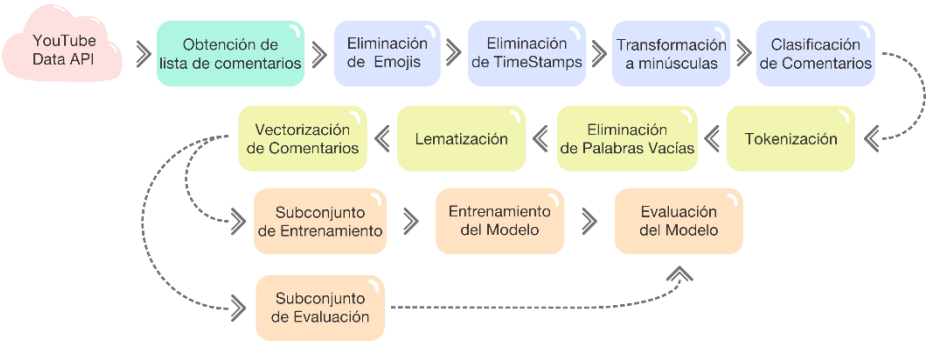

Fig. 2. Flujo de procesamiento de comentarios hasta el entrenamiento y evaluación del modelo SVC seleccionado.

Con el modelo entrenado se introduce el subconjunto de comentarios de pruebas para obtener las predicciones y evaluar el rendimiento del modelo. Para esto existen distintas métricas [13] y están estrechamente relacionadas con la Matriz de Confusión (*Confusion Matrix*), la cual es una herramienta que permite visualizar el desempeño de un algoritmo de aprendizaje supervisado [14].

En la figura 3 se muestra la matriz de confusión que se aplica por cada etiqueta del dataset.

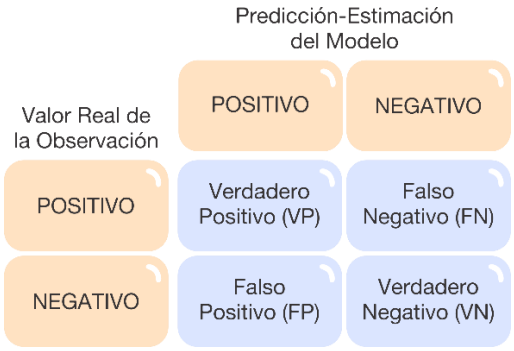

Fig. 3. Matriz de Confusión general.

Exactitud (*Accuracy*): Esta medida es bastante intuitiva debido a que se calcula como el cociente entre las predicciones correctas y el total de las observaciones. No es una medida adecuada cuando el conjunto de observaciones es desbalanceado.

Expresado en términos de la matriz de confusión la fórmula seria:

$$Exactitud = \frac{Verdadero\ Positivo + Verdadero\ Negativo}{Total\ de\ Observaciones} \quad (1)$$

Precisión (*Precision*): hace referencia a la capacidad del modelo de identificar solo los casos que son relevantes (verdaderos positivos), por ejemplo, de un grupo de comentarios etiquetados como negativo (sentimiento), ¿cuántos de ellos transmiten ese sentimiento en realidad? A mayor valor de precisión es menor la tasa de falsos positivos. En fórmula 2 queda expresado su cálculo.

$$Precisión = \frac{Verdaderos\ Positivos}{Verdaderos\ Positivos + Falsos\ Positivos} \quad (2)$$

Sensibilidad (*Recall*): hace referencia a la capacidad del modelo de identificar a todos los casos relevantes dentro del conjunto de observaciones. Es decir, siguiendo el ejemplo anterior, de todos los comentarios que realmente son negativos, ¿el modelo cuántos etiqueto como tal? En la fórmula 3 se presenta su cálculo.

$$Sensibilidad = \frac{Verdaderos\ Positivos}{Verdaderos\ Positivos + Falsos\ Negativos} \quad (3)$$

F1-Score: es una función de Precisión y Sensibilidad que se define como la media armónica entre estas dos medidas. Es útil considerarla para conjuntos de observaciones desbalanceados o cuando los valores de las métricas anteriores varían demasiado. En la fórmula 4 se presenta su cálculo.

$$F1 - Score = 2 * \frac{Precisión * Sensibilidad}{Precisión + Sensibilidad} \quad (4)$$

Desde la librería ***scikit-learn*** se pueden llamar a funciones que devuelven los valores para las métricas mencionadas. En la tabla 1 se presentan los valores obtenidos para el modelo entrenado.

Tabla 1. Resultado de Métricas para el modelo entrenado.

Etiquetas	Precisión	Sensibilidad	F1-Score	Exactitud
pos	0.96	0.95	0.95	0.955
neg	0.95	0.96	0.96	

4.2 Aplicación y Análisis de los Resultados.

Para esta etapa, se selecciona un video de un canal de YouTube en español. Este video presenta a una persona explicando una situación personal delicada y en respuesta existen comentarios positivos donde la gente expresa su empatía y apoyo y por otro lado existen comentarios negativos donde la gente ofende y desacredita que los hechos sean reales.

Se extraen los comentarios y con el modelo entrenado se procede a obtener la clasificación en comentario "*pos*" por positivo o "*neg*" por negativo. Para poder ver los resultados de una forma más apropiada que una tabla de comentarios con etiquetas es que se realiza una nube de palabras por cada grupo: comentarios positivos y comentarios negativos como se muestra en las figuras 4 y 5.

Fig. 4. Nube de palabras para comentarios positivos.

Fig. 5. Nube de palabras para comentarios negativos.

El fin es visualizar cuales son las palabras más comunes o utilizadas en los comentarios para describir el sentimiento de los usuarios. Además, al analizar de estos resultados se aprecia que el conjunto de palabras que resaltan en ambas nubes es consistente con el sentimiento que buscan representar. Cabe destacar que en el caso de la nube de comentarios negativos se realizó un desenfoque a palabras groseras.

Como elemento adicional, se realiza un gráfico de tortas presentando el porcentaje de comentarios positivos versus comentarios negativos presentado en la figura 6.

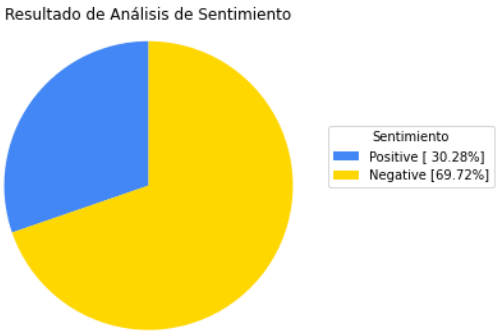

Fig. 6. Gráfico de torta cono porcentajes de comentarios positivos y negativos.

Como se ha mencionado, el video analizado es sobre una *youtuber*, palabra con la que se denominan a los usuarios de esta plataforma, explicando situaciones personales pero que también comprometen a otra persona, también *youtuber*. En esta plataforma existen enfrentamientos igual que en cualquier otra red social. Además, es común que suscriptores también se involucren tomando parte a través de los comentarios de apoyo o agresivos que se analizan en este trabajo. En este caso en particular se aprecia que los comentarios negativos están cerca del 70% del total indicando que muchos usuarios están en desacuerdo con el contenido del video ya sea porque el protagonista carece de credibilidad o estos usuarios simpatizan con el *youtuber* al cual se involucra.

Existen personas que se ven muy afectadas por el acoso y agresiones, y si además es continuo puede llevar a resultados trágicos, más aún si los involucrados son adolescentes. Justamente hoy en día son ellos quienes participan activamente en las redes sociales buscando reconocimiento y tomando riesgos, llevando sus acciones al extremo por conseguir más seguidores, suscriptores o visitas.

5 Conclusión

¿Es posible utilizar este modelo entrenado para análisis en otras redes sociales? Si, es posible, en sí, el modelo fue entrenado con mensajes cuyo tratamiento hizo posible que sus predicciones no estén ligadas a una red social sino al sentimiento expresado en un

texto. Por eso es que también se puede usar la API de la red social *Twitter* para hacer el proceso de obtención de tweets respecto a un tema o cuenta en particular, realizar el proceso de limpieza, obtener las predicciones y ejecutar un análisis a partir de la visualización de datos.

Como mejoras a este trabajo, es posible agregar un paso más en el análisis final con ayuda de la API de YouTube donde también se puede extraer el nombre de las cuentas que realizan comentarios y ver su frecuencia de participación en mensajes clasificados como negativos. Además, también es posible saber la fecha de creación de la cuenta y poder determinar si fue creada recientemente indicando que son cuentas desechables ya que es una práctica común que las personas no utilicen su propia cuenta para realizar actividades dañinas.

Extraer automáticamente información considerada relevante de textos puede parecer una tarea trivial para el ser humano, pero para una maquina es un verdadero reto. Por esto es que es importante una etapa de preprocesamiento que posibilite el análisis de textos. Sin embargo, existe una situación particularmente difícil en las que este modelo y muchos otros no tratan correctamente y es el sarcasmo o doble intención ya que incluso pueden ser difíciles de comprender para un humano.

En este trabajo se utiliza un algoritmo de aprendizaje supervisado, pero dependiendo de la complejidad a la que se requiera llegar, existen mejores opciones, por ejemplo, el uso de redes neuronales. Un ejemplo de esto es Bert, un modelo de código abierto de procesamiento del lenguaje natural y es posible hacer uso de él, adicionándolo como una capa más si se trabaja con redes neuronales. Al final, se resume en poder detectar eficazmente casos de violencia verbal, las víctimas y sus agresores para así poder tomar decisiones, mientras más circunstancias sean tomadas en cuenta, se obtienen mejores resultados.

Referencias

1. Ravalli, M. J., Argentina es líder en ‘cyberbulling’ según un estudio global. Obtenido de https://www.perfil.com/noticias/elobservador/argentina-es-lider-en-cyberbulling-segun-un-estudio-global.phtml en marzo de 2020.
2. Justicia de la Torre M.C., Nuevas Técnicas de Minería de Textos: Aplicaciones. Universidad de Granada. España. Ed Universidad de Granada. Tesis Doctorales (2017)
3. Prabhu, Understanding NLP Word Embeddings — Text Vectorization. Obtenido de https://towardsdatascience.com/understanding-nlp-word-embeddings-text-vectorization-1a23744f7223 en marzo de 2020.
4. Jocelyn D'Souza, An Introduction to Bag-of-Words in NLP. Obtenido de https://medium.com/greyatom/an-introduction-to-bag-of-words-in-nlp-ac967d43b428 en febrero de 2020.
5. Chris Nicholson, A Beginner's Guide to Bag of Words & TF-IDF. Obtenido de https://pathmind.com/wiki/bagofwords-tf-idf en marzo de 2020.
6. Chopra D., Joshi N., Mathur I., Mastering Natural Language Processing with Python. Packt Publishing (2016)
7. Erick Fonseca, State-of-the-art Multilingual Lemmatization. Obtenido de https://towardsdatascience.com/state-of-the-art-multilingual-lemmatization-f303e8ff1a8 en abril de 2020.

8. Shubham Singh, NLP Essentials: Removing Stopwords and Performing Text Normalization using NLTK and spaCy in Python. Obtenido de https://www.analytics-vidhya.com/blog/2019/08/how-to-remove-stopwords-text-normalization-nltk-spacy-gen-sim-python/ en abril de 2020.
9. The Stanford Natural Language Processing Group. Obtenido de https://nlp.stanford.edu/ en abril de 2020.
10. David Batista, Applying scikit-learn TfidfVectorizer on tokenized text. Obtenido de http://www.davidsbatista.net/blog/2018/02/28/TfidfVectorizer/ en abril de 2020.
11. Documentación de Amazon Web Services, Dividir los datos en datos de formación y evaluación. Obtenido de https://docs.aws.amazon.com/es_es/machine-learning/latest/dg/splitting-the-data-into-training-and-evaluation-data.html en febrero de 2020.
12. Joachims Thorsten, Text categorization with Support Vector Machines: Learning with many relevant features. In: Nédellec C., Rouveirol C. (eds) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1398, pp 137-142. Springer, Berlin, Heidelberg (2005)
13. Grus J., Data Science from Scratch. 1st Ed. O'Reilly (2015)
14. Gerón A., Hands-On Machine Learning with Scikit-Learn & TensorFlow. 1st Ed. O'Reilly (2017)