Clasificación de "toxicidad" de comentarios en Wikipedia usando técnicas de procesamiento de texto

Danahí Ayzailadema Ramos Martínez[a], **Yalidt Díaz Vázquez**[a], **and César Zamora Martínez**[a]

[a]Alumnos de Maestría en Ciencias de Datos (ITAM)

En fechas recientes el uso de plataformas digitales como un medio de comunicación se ha vuelto muy común en el mundo. Sin embargo, a través de dicha herramientas en línea se observan comportamientos de molestia, acoso y violencia de usuarios a otros por motivos diversos que los hacen difíciles de identificar, de modo que puedan ser aprovechados posteriormente para crear sistemas que prevengan tales conductas y sirvan para regularlos. Motivado por ello, en este trabajo se plantea un análisis basado en técnicas de procesamiento de texto en conjunto con modelos de aprendizaje de máquina con el propósito de poder detectar este tipo de comentarios según su nivel de "toxicidad".

El "comportamiento tóxico" se refiere a una serie de acciones o expresiones de personas que se dan en el seno de una comunidad y que potencialmente pueden dañar a otros individuos o grupos dentro la misma. Tales circunstancias apuntan, desde luego, a que bajo diferentes contextos y criterios cierta conducta pueda considerarse o no como un "comportamiento tóxico", sin embargo, las acciones o expresiones que suelen etiquetarse como "tóxicas" reflejan matices de discurso potencialmente agresivo, hostil o discriminatorio, además de que llevan consigo una carga de insulto, acoso y otras suertes de abuso verbal dirigido hacia las personas, típicamente como ataques personales (Dunn (2020)).

Por otro lado, resulta notable que durante las últimas tres décadas las comunicaciones digitales han tenido un avance sin precedentes en el mundo, posicionándose como herramientas que impulsan el desarrollo económico y social, pues permiten crear oportunidades, facilitar la interacción entre individuos e impulsar el progreso económico y social para el bienestar de la población. En tal contexto, han cobrado relevancia las plataformas digitales como canales que permiten la comunicación de personas en diferentes lugares del mundo, no solo con niveles asequibles de inversión, sino también desde una manera ágil e interactiva.

Dada libertad que se otorga a los usuarios en dichos entornos de comunicación, se ha observado la proliferación de comportamientos de ataque personales a usuarios por motivos muy diversos, lo que ha llevado a muchas comunidades a limitar o cerrar por completo los comentarios de los usuarios. Es así que los comportamiento tóxicos pueden afectar del desenvolvimiento de una comunidad y a sus usuarios de manera perjudicial, ya que la amenaza de abuso y acoso en línea significa que las personas dejan de expresarse y a su vez pueden cesar de buscar opiniones diferentes. Es por ello que comprender la prevalencia y el impacto de los ataques personales en las plataformas a través del discurso es un problema complejo que es de interés para muchos actores, ya que puede facilitar la implementación de intervenciones que prevengan este tipo de conductas y ayuden a que las conversaciones se den de manera efectiva.

Por otro lado, de entre todas las plataformas en línea destaca Wikipedia, la cual ha cobrado enorme popularidad desde su creación en 2001. Este sitio funciona como una enciclopedia de acceso libre donde existen más de 500 millones de artículos que abarcan una variedad de temas desde científicos, biográficos, literarios y hasta de cultura popular, los cuales han sido redactados en 300 idiomas (Wikipedia (2020)). Actualmente es administrada por la Fundación Wikimedia, la cual es una organización sin fines de lucro, que se auto-define como "un movimiento global cuya misión es llevar contenido educativo gratuito al mundo" (Wikimedia (2020)).

Cabe destacar que la creación del contenido de los artículos de este Wikipedia se basa en la edición colaborativa, realizada en línea. Bajo dichas circunstancias es natural que se lleven a cabo intercambios de opinión y discusiones como mecanismos relevantes para los editores y su trabajo creativo. Lamentablemente, al igual que dentro de otras comunidades en línea, en Wikipedia las discusiones no son sólo el lugar de coordinación y cooperación; también son una vía importante por la cual las personas involucradas en dichos proyectos experimenten conductas de acoso y toxicidad.

Para ilustrar como estos comportamientos afectan a la comunidad y a sus usuarios en un sentido perjudicial se debe mencionar que en Noviembre de 2015 se llevo a cabo una encuesta denominada "Harassment survey" la cual englobó al personal que laboraba en los diversos proyectos de la Fundación Wikimedia (incluyéndose a Wikipedia), obteniéndose que un 54% de las víctimas de acoso informaron una menor participación en los proyectos en los que estaban trabajando (Wikimedia (2016)).

Sobre este tema específico, cabe destacar que a raíz de la colaboración entre la Fundación Wikimedia y Jigsaw [*], a través de su iniciativa de investigación Conversation AI[†], en 2018 se hicieron disponibles un conjunto de comentarios nacidos del entorno de crowdsourcing de Wikipedia Talk, mediante el sitio electrónico de Kaggle[‡] (Kaggle (2018)).

[*] Jigsaw es una empresa que forma parte de Alphabet (filial de Google), y que se define a si misma como una "incubadora de tecnología" y que entre sus objetivos relacionados la censura en Internet, reducción de las amenazas de ataques digitales, contrarrestar la violencia del extremismo y proteger del ciber-acoso a las personas. https://jigsaw.google.com/vision/

[†] Conversation AI se refiere a si misma como un "esfuerzo de investigación colaborativo que explora técnicas de aprendizaje de maquina como una herramienta para mejores discusiones en línea", teniendo como objetivo ayudar a aumentar la participación, la calidad y la empatía en la conversación en línea a gran escala. https://conversationai.github.io

[‡] Kaggle, una subsidiaria de Google LLC, que funge como una comunidad en línea de científicos de datos y profesionales del aprendizaje automático. Una de sus características más llamativas

En líneas generales, tales datos de Wikipedia incluyen el texto de los comentarios expresados por usuarios exclusivamente en idioma inglés, sobre los que un equipo de 5,000 personas asoció un conjunto de etiqueta para calificarlos, de ser el caso, de acuerdo con su toxicidad. Ello en consideración de que no todos los comentarios de los usuarios reflejan este tipo de conductas; sin embargo, en caso afirmativo, para distinguir el tipo de toxicidad del contenido de los mismos, se emplearon una serie de etiquetas, dadas por: toxico (*toxic*), severamente tóxico (*severe toxic*), obsceno (*obscene*), de amenaza (*threat*), insulto (*insult*), de odio a la identidad (*identity hate*).

Un punto que no debe perderse de vista es que, como se ha dicho antes, bajo diferentes contextos y criterios una cierta conducta pueda considerarse como tóxica, por lo que, particularmente, en razón de que el proceso de etiquetado de los comentarios se realizó con intervención humana, los criterios bajo los cuales un comentario se le asoció o no una determinada etiqueta de "comportamiento tóxico" podrían no ser necesariamente uniformes y presentar variabilidad, lo que su vez representa un reto a la hora de construir modelos predictivos que internalicen apropiadamente dicho fenómeno.

Teniendo presente todo lo anterior, el objetivo de este documento es explorar diversos conceptos surgidos en el análisis de texto y técnicas de aprendizaje de máquina para poder predecir a partir del cuerpo de los comentarios vertidos por un usuario, si este presenta un "comportamiento tóxico".

De esta manera, la idea será realizar un análisis desde dichas perspectivas para 1) procesar los comentarios de los usuarios, intentando preservar el sentido original de sus mensajes, 2) incorporar el contexto de los comentarios que rodean al discurso tóxico de los usuarios y 3) diseñar modelos de aprendizaje de máquina que incorporen estos elementos para mejorar el desempeño en la detección de este tipo de conductas verbales.

Esto se propone como punto de partida para el diseño de modelos que puedan ser empleados para detectar comportamiento tóxicos de manera automatizada y, posiblemente, auxiliar el diseño de sistemas que puedan intervenir oportunamente para frenar tales conductas.

En adelante, se abordará la información consultada junto con la metodología propuesta para tal efecto.

1. Descripción de información y análisis exploratorio

A continuación se describirán los datos para el análisis de este procesamiento de texto y modelos de clasificación, junto con las consideraciones particulares derivadas de su exploración[§].

Se debe precisar que, toda vez que las acciones o expresiones que suelen etiquetarse como "tóxicas" reflejan matices de discurso potencialmente agresivo, hostil o discriminatorio y por ende el lenguaje que se mostrará a continuación puede contener expresiones con una fuerte carga de insulto y acoso que podrían considerarse ofensivo para ciertos grupos, las cuales deben involucrarse en el análisis en el sentido de que reflejan el *status quo* del abuso verbal presente en los usuarios de Wikipedia y no así de los autores del presente documento, por lo que se aconseja discreción.

es que permite encontrar y publicar conjuntos de datos, explorar y construir modelos en un entorno de ciencia de datos basado en la web, trabajar con otros científicos de datos e ingenieros de aprendizaje automático, y participar en concursos para resolver desafíos de ciencia de datos. https://www.kaggle.com

[§]El procesamiento de la información se llevó a cabo a través de Python, empleando entre otras librerías de scikitlearn, tensorflow y keras

A. Información de comentarios. En términos generales, la revisión de la información abarcó los datos publicados por la Fundación Wikimedia y Jigsaw a través de Kaggle, donde se pone a disposición del público la información de comentarios realizada por diferentes usuarios junto con la clasificación del contenido de los mensajes de texto que realizaron en Wikipedia Talks. En concreto, la información de comentarios que realizaron los usuarios de dicha plataforma se provee con el siguiente nivel de detalle:

- El código identificador del usuario que realizó el comentario (valor alfanumérico),
- Comentario realizado por el usuario (texto),
- Variable indicadora de si el comentario se considera "toxic" (valor booleano),
- Variable indicadora relativa a si el comentario se considera "severe toxic" (valor booleano),
- Variable indicadora correspondiente a si el comentario se considera "obscene" (valor booleano),
- Variable indicadora de si el comentario se considera "threat" (valor booleano),
- Variable indicadora relativa a si el comentario se considera "insult" (valor booleano),
- Variable indicadora correspondiente si el comentario se considera "identity hate" (valor booleano).

Cabe destacar que los datos publicados para el concurso de Kaggle, se proveen a través de dos conjuntos denominados *train* y *test*. Sin embargo, únicamente el conjunto *train* se conforma de comentarios en texto y etiquetas de toxicidad. Debido a ello, se decidió avanzar en el presente trabajo considerando únicamente al conjunto *train*, pero generando subdivisión aleatoria sobre el mismo para probar el desempeño de los modelos planteados más adelante. Esta separación conjunto *train* se realizó considerando una proporción de .20 y .80; dicha consideración será adoptada a lo largo del presente trabajo, y los conjuntos en comento se denominarán como conjuntos de "entrenamiento" y "prueba", respectivamente.

Sin embargo, para el análisis exploratorio que se presenta a continuación, se remite al conjunto de "entrenamiento", sin pre-procesamiento del texto ni censura.

A.1. Análisis con conteos. En tal sentido, el primer punto explorado fue realizar conteos de los comentarios, desagregados de acuerdo a la etiqueta de toxicidad asociada. Aquellas que no fueron calificados con alguna etiqueta de toxicidad se especifican como "Clean". La siguiente tabla resume los resultados obtenidos:

Table 1. Tabla de conteos incluyendo los comentarios no catalogados como tóxicos

Clasificación	Frecuencia absoluta	% Frecuencia relativa
Clean	114,740	80.46
Toxic	12,171	8.53
Obscene	6,696	4.69
Insult	6,254	4.38
Severe Toxic	1,264	0.88
Identity hater	1,088	0.76
Threat	384	0.26

De lo anterior se desprende que, en cerca de 80% de los comentarios, no se presentó ningún tipo de toxicidad. Sin embargo, en el 20% donde si existe tal fenómeno, hay diversidad en el tipo de toxicidad detectada. Para ilustrar este punto, se presenta un gráfico de barras, focalizado hacia comentarios tóxicos:

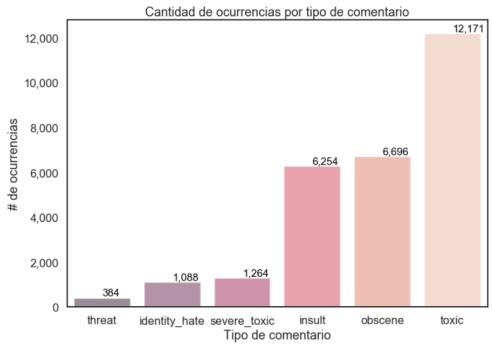

Fig. 1. Gráfica de conteos por tipo de comentario. Fuente: Adaptación de*adityakarampudi*, "Toxic comment classification"

De acuerdo a la figura 1, se identificó que dentro de las clasificaciones de toxicidad la que tiene mayor concurrencia es *toxic* (44% de observaciones), seguida de comentarios de tipo *obscene* e *insult* (respectivamente, 24% y 22% de observaciones). En contraste, la categoría con menor aparición fue *threat* con apenas 1% de observaciones.

B. Múltiples etiquetas en comentarios y tablas de contingencia. Otro punto de interés para el análisis es notar que los comentarios pueden pertenecer a varias categorías de toxicidad simultáneamente. Ello en razón de que un comentario puede ser usado para atacar a una persona en su individualidad desde varios ejes. Es así que, es de esperarse que las categorías de toxicidad para los comentarios no sean exclusivas.

Para ello se ha construido un gráfico que muestra los conteos de números de etiquetas por comentarios de Wikipedia Talks.

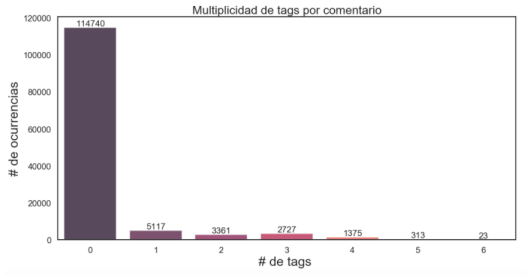

Fig. 2. Gráfica de conteos por número de etiqueta. Fuente: Adaptación de*adityakarampudi*, "Toxic comment classification"

Al respecto, tras analizar la información de la figura 2 se encontraron que de entre los comentarios que poseen algún nivel de toxicidad, aquellos que tienen sólo una etiqueta representa el 40%, seguidos de los que poseen dos o tres etiquetas simultáneamente (mismos que representan 26% y 21% de los comentarios con alguna etiqueta de toxicidad) . Sin embargo, sólo en el *set* de entrenamiento existen 23 comentarios que tienen 6 categorías al mismo tiempo, por lo que se realizará una análisis con tablas de contingencia para tener mayor información sobre estas características.

Table 2. Tabla de contingencia en porcentajes sobre el total de comentarios con etiqueta como referencia: *toxic*

	Severe Toxic		Obscene		Threat		Insult	
Toxic	0	1	0	1	0	1	0	1
0	90.46	0	90.14	0.32	90.44	.018	90.12	0.33
1	8.54	.99	4.61	4.92	9.25	0.28	4.97	4.56

En la tabla (2) se refleja un dato curioso acerca de los comentarios tóxicos y severamente tóxicos, ya que el 99% se traslapan, es decir que casi todos los comentarios que son severamente tóxicos son tóxicos por default.

De igual manera, existe un 4.92% de los comentarios que son obscenos y al mismo tiempo tóxicos, lo cual en esta tabla de referencia es la combinación a parte de la anteriormente mencionada que presenta mayor número de traslapes.

Para abundar en el análisis, también se presenta la tabla de contigencia de los comentario que tienen la etiqueta *Obscene*, contra el resto de etiquetas de toxicidad:

Table 3. Tabla de contingencia en porcentajes sobre el total de comentarios con etiqueta como referencia: *obscene*

	Toxic		Severe toxic		Threat		Insult	
Obscene	0	1	0	1	0	1	0	1
0	90.14	4.6	94.71	0.044	94.64	0.11	93.68	1.07
1	0.32	4.92	4.29	0.94	5.05	0.18	1.41	3.82

Respecto a la tabla 3, se encontró que sólo el .044% de los comentarios es clasificado como severamente tóxico y no obsceno, en contraste existe un 94% que se considera severamente tóxico y obsceno a la vez.

Este análisis con tablas de contingencia ayudó a determinar que prácticamente todas las etiquetas son subconjunto de otras etiquetas, unas en mayor proporción que otras a excepción del caso mencionado en la tabla 2, donde se concluyó que casi todos los comentarios que son severamente tóxicos serán tóxicos.

De igual manera, se logra tener mayor conocimiento sobre las clasificaciones que se quieren predecir en la parte del modelado, debido a que se desconoce el criterio que se tomó para realizar el etiquetado.

C. Análisis de palabras frecuentes por categorías. Para mostrar la variedad de léxico que se emplea en los comentarios según su etiqueta de toxicidad, se construyeron nubes de palabras[¶], donde el tamaño y el color de los términos es proporcional a la frecuencia con que fueron empleados sobre dichos grupos. Los resultados obtenidos se aprecian en las figuras (3) y (4).

Al respecto, de acuerdo con la figura (3), en la cual se hace una comparación de nubes de palabras entre las categorías

[¶] Empleando la librería *WordCloud* de Python

Fig. 3. Nube de palabras de la categoría Toxic y Severe Toxic. Fuente:Adaptación de*aditya-karampudi*, "Toxic comment classification"

Toxic y *Severe Toxic* se encontró que tienen varias palabras similares entre sus textos como por ejemplo : *suck* y *you*, aunque resulta notable la palabra *you* es utilizada para complementar una insulto cuando se aplica directamente a otra persona.

Fig. 4. Nube de palabras de la categoría Threat y Obscene Toxic. Fuente: Adaptación de*aditya-karampudi*, "Toxic comment classification"

Del mismo modo, en la gráfica (4) se comparan las nubes de palabras de las categorías *Threat* y *Obscene*, respectivamente. Del análisis se desprende que una de las palabras que tienen en común estas categorías es *Die*. Por otra parte, en la clasificación *Threat* se encuentran palabras como *Kill* en varios conjuntos de frases, que es comunmente empleanda en discursos de tono amenaza a la integridad de una persona.

D. Procesamiento del texto de los en comentarios. Con miras hacia la etapa de modelado, tras analizar el texto presente en los comentarios, se decidieron ejecutar, principalmente las siguientes acciones de limpieza:

- Transformar el texto a minúsculas,
- Se decidió eliminar las mención "User" que se incluía residualmente en algunos mensajes, como preámbulo al mensaje del usuario,
- Quitar espacios y caracteres de salto de línea,
- Sustituir, en la medida de lo posible, contracciones usadas en el idioma inglés,
- Remover direcciones IP del texto[‖],
- Eliminar enlaces de páginas de internet[**]„

E. Unigramas y Bigramas principales. Para complementar el análisis, también se estudiaron los unigramas y bigramas de acuerdo a la métrica *Tf-idf*.

A este respecto, por cada categoría se buscaron las 10 palabras con valor más alto de td-idf tal como se muestra en la figura (5) en las categorías *Toxic*, *Severe toxic*, *Insult* y *Obscene* la mayoría de las palabras más utilizadas coinciden, pero en diferente orden. Por otro lado las categorías *Identity hate* y *Threat* además de tener algunas palabras más relevantes según la métrica *Tf-idf* respecto a las demás categorías, se diferencian por tener palabras que resaltan su clasificación.

[‖] Se refiere a un número que identifica de forma única a una interfaz en red de cualquier dispositivo conectado a ella que utilice el protocolo IP (Internet Protocol), que corresponde al nivel de red del modelo TCP/IP para transmisión de datos en internet.

[**] En concreto se eliminaron los enlaces de tipo *https://...*

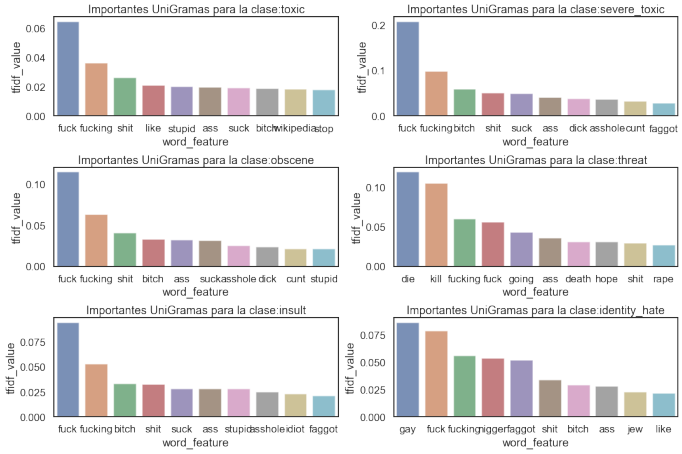

Fig. 5. Top 10 de Unigramas más importantes según medida tf-idf por clase. Fuente: Adaptación de*aditya-karampudi*, "Toxic comment classification"

Dando continuidad al análisis de texto, se buscaron los bigramas más relevantes según la métrica *Tf-idf*, tal como se muestra en la figura (6), dándose a notar que si bien todas las categorías comparten la mayoría de sus bigramas notables, se encuentran en diferente orden de mayor métrica *Tf-idf*.

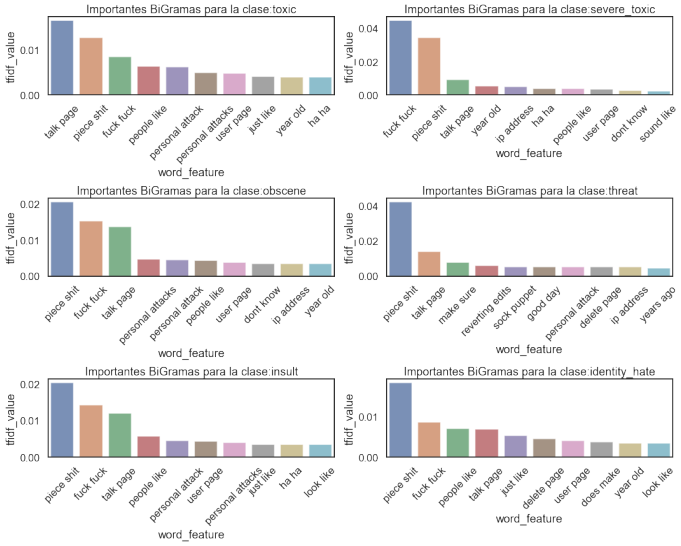

Fig. 6. Top 10 de Bigramas más importantes según la métrica *Tf-idf* por clase. Fuente: Adaptación de*aditya-karampudi*, "Toxic comment classification"

2. Modelado e implementación

Tal como hemos comentado en las secciones previas, los datos de Wikipedia Talks guardan una serie de campos que describen el texto de los comentarios expresados por los usuarios, así como variables binarias que indican las diversas etiquetas referentes a toxicidad que les fueron asociadas en un proceso colaborativos, empleando 5,000 personas para dicha tarea.

Al respecto, como se se ha evidenciado en el análisis exploratorio de la sección previa, existen varios puntos que deben ser considerados para desarrollar modelos que intenten inferir:

- Existe léxico que puede denotar toxicidad pero que es

empleado en comentarios que pertenecen diferentes etiquetas,
- Existen comentarios puede tener más de una etiqueta de toxicidad,
- Las categorías de toxicidad se presentan en el conjunto de datos con proporciones heterogéneas,
- De los puntos anteriores, se desprende que tanto las categorías de toxicidad como el léxico dentro de estas no es mutuamente excluyente a nivel de comentarios; existe un traslape entre estos y algunas comentarios pueden pertenecer o no a varias categorías simultáneamente

En línea con ello, a continuación se realizará una exposición de los conceptos relativos al análisis de texto y de aprendizaje de máquina que fueron revisados para plantear modelos dirigidos al problema de predicción una-a-una de las etiquetas que pueden tener los comentarios y, en segunda, del problema de predicción de todas las etiquetas de toxicidad a las que puede pertenecer, empleando herramientas de procesamiento de texto, así como modelos de regresión logísticas y redes recurrentes.

Cabe destacar que las implementaciones se realizaron a través de la plataforma Google Colab.

Bajo este enfoque, las fases de la metodología para abordar el problema son las siguientes:

A. Conceptos teóricos.

A.1. N-gramas y N-gramas de caracteres. Dentro del contexto de procesamiento de lenguaje natural, se conoce como corpus es una colección de textos (o habla) del lenguaje de interés. El vocabulario es una colección de palabras que ocurren al interior del lenguaje, pero que en específico se materializan a través corpus. Desde luego, para representar un texto a partir de su contenido es necesario definir que se considera como una palabra (o token). Se conoce como vocabulario al conjunto de todas las palabras posibles en nuestros textos.

Dado un corpus escrito, se conoce como n-grama de nuestro lenguaje a una sucesión de longitud n de palabras adyacentes $w_1 w_2 \dots w_n$. La idea detrás de este tipo de representaciones es que al considerar con qué medida de frecuencia la palabra X es seguida por la palabra Y, se pueda construir un modelo que captura las relaciones entre estas.

Una idea similar se puede aplicar sobre los caracteres que conforman las palabras, en este caso un n-grama de caracteres de longitud n es una sucesión de caracteres que aparecen de manera adyacentes en palabras $c_1 c_2 \dots c_n$, también conocidos como n-tejas.

A.2. Frecuencia inversa en documentos ($tf - idf$). Considerando lo anterior, una pregunta relevante es cómo cuantificar de qué trata un documento, naturalmente la idea es poder inferirlo a partir de las palabras que componen el documento. En este sentido, una medida que pretende representar cuán importante puede ser una palabra es su frecuencia de término (tf), es decir, la frecuencia con que aparece una palabra en un documento. Sin embargo, hay palabras en un documento que ocurren muchas veces pero pueden no ser tan relevantes para el problema en estudio[††], por lo que se deben hacer algún tipo de ajuste a los conteos de frecuencia.

Otro enfoque es observar la frecuencia de documentos inversa de un término(idf), que disminuye el peso de las palabras de uso común y aumenta el peso de las palabras que no se usan mucho en una colección de documentos. Esto se puede combinar con la frecuencia de término para calcular la medida conocida como $tf - idf$ de un término (las dos cantidades multiplicadas juntas), la frecuencia de un término ajustada por la poca frecuencia con la que se usa. Dicho concepto pretende medir la importancia de una palabra para un documento en una colección (corpus) de documentos.

A.3. Representación de palabras y encajes. Una de las ideas fundamentales de este enfoque es representar a cada palabra como un vector numérico de dimensión d, es decir como parte de una abstracción que forma parte de un espacio vectorial, de forma que las palabras similares o relacionadas se encuentren representadas por puntos cercanos. Esto se llama una representación vectorial distribuida, o también un *embedding* de palabras, en la que cada entrada tiene asociada una representación numérica con alguna codificación relevante para el problema que se puede entender como una descripción numérica de cómo funciona una palabra en el contexto de su n-grama (por ejemplo, a través de codificaciones que se calibren usando modelos de aprendizaje de máquina o que aproveche medidas como $tf - idf$).

Este paso funciona como una especie de traducción de lenguaje hacia un espacio de representaciones que pueden ser procesadas usando modelos de aprendizaje de máquina.

A.4. Representación word2vec-GoogleNews-vectors. En el contexto de procesamiento de lenguaje natural, una idea recientemente explorada es obtener la representación vectorial de palabras a entrenadas previamente en un corpus suficientemente amplio para abarcar al problema de nuestro interés.

En la actualidad, existen encajes que organizaciones han entrenado basados en algún corpus y que se ponen a disposición del público. Entre ellas, destaca la herramienta word2vec preentrenado de Google, en Google News[‡‡], el cual es un corpus consistente en 3 mil millones de palabras que cuenta con 3 millones de vectores de palabras en inglés de 300 dimensiones, que ha sido empleando *negative sampling*.

Este encaje es de potencial interés para el problema que nos ocupa, pues al ser entrenado con un corpus tan amplio, es probable que se adapte mejor que otros encajes nuestro contexto de detección de "toxicidad".

A.5. Redes recurrentes y LSTM. Dentro del contexto de aprendizaje profundo, las redes recurrentes son una familia de modelos diseñados para lidiar con datos que presentan estructura secuencial, de manera que su diseño ayuda a que en un determinado punto las redes "recuerden" estados previos y puedan incorporar tal información en la predicción del siguiente estado. Esta idea de retroalimentación a partir del pasado, corresponde a introduciendo bucles en el diagrama de la red.

Dentro de dicha familia de modelos destaca las redes *Long Short Term Memory* (LSTM, por sus siglas en inglés), introducidas en Hochreiter and Schmidhuber (1997), que pueden aprender dependencias largas, por lo que se puede pensar que tienen una "memoria" a largo plazo. Como se ha dicho antes, en esencia, todas las redes neuronales recurrentes tienen la forma de una cadena de módulos repetitivos de red neuronal, pero el módulo de repetición de la LSTM tiene una estructura con una sola capa de red neuronal, en conjunto hay

[††] Podemos pensar, por ejemplo, en artículos, conectores de discurso o preposiciones

[‡‡] https://code.google.com/archive/p/word2vec/

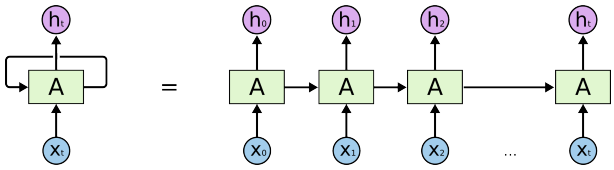

Fig. 7. Esquema de una red recurrente, tomado de http://colah.github.io/posts/2015-08-Understanding-LSTMs/

cuatro celdas que interactúan de una manera muy especial para incorporar la información de estados presentes y pasados.

De manera esquemática se tratan de 1) *forget gate:* decide que, considerando h_{t-1} y x_t, estima usando una función de activación (sigmoide) que información se debe desechar o guardar para considerarse con el estado de la celda C_{t-1}, 2) *input gate:* este paso permite decidir que valores del estado se actualizarán, 3) *cell state:* permite incorporar la información de la puertas previas para dedicir que informacion se almacenara en la celda y 4) *output gate:* decide cuál debe ser el siguiente estado oculto (dicho estado oculto también se usa para predicciones).

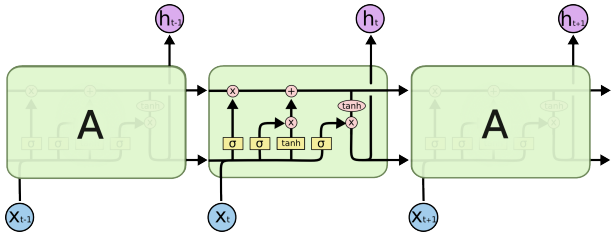

Fig. 8. Esquema de una red LSTM, tomado de http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Tales redes son particularmente útiles en el procesamiento de texto, dado que la estructura semántica del lenguaje otorga cierto comportamiento secuencial que puede ser incorporado con tales modelos.

B. Modelos y variables a considerar.

B.1. Regresión Logística con N-gramas. En este primer modelo, el cual sigue la línea de uno de los mejores punteados en el contexto del concurso de *Kaggle* (Tunguz (2018)) donde se tiene como objetivo realizar las predicciones para una etiqueta a la vez, a diferencia del resto de modelos planteados en secciones posteriores los cuales abordan el problema de predicción de toxicidad en comentarios desde un punto de multi-etiquetas. El algoritmo de este modelo se basa principalmente en tres fases:

En primer lugar, calcula la métrica *Tf-idf* con una función ya implementada en Python, "TfidfVectorizer" del paquete "sklearn". Esta función realiza la vectorización de las palabras del corpus para determinar su importancia y toma en cuenta dos factores (la frecuencia del término y la frecuencia inversa del documento), con lo cual se logra dar mayor ponderación a las palabras que son frecuentes en el documento pero poco frecuentes en el corpus. De esta manera se tiene un mayor acercamiento de aquellas palabras que son tipo *key words*, ya que por ejemplo la palabra *you* es muy probable que aparezca con alta frecuencia en un texto, pero si se comparan varios textos tendría poco peso de importancia porque no aporta mucho al conjunto del corpus. Es posible que esta métrica sea un poco confusa de entender, por lo cual se expondrá un ejemplo análogo: si imagináramos que salimos de viaje a un lugar que jamás hemos visitado y se desea elegir algún restaurante para cenar, pero se tienen dos objetivos: cenar un platillo rico y al mismo tiempo probar algo típico de esa cuidad. Eso significaría que no probaríamos el platillo que cualquier lugar ofrezca, más bien buscaríamos alguna recomendación para elegir el sitio adecuado, con tal de obtener un platillo **distintivamente bueno.**[§§]

En segundo lugar, para el cálculo de vector "TfidfVectorizer", se puede realizar la vectorización con n-gramas por **palabra o por caracteres**, dado que en la función se puede ingresar un argumento llamado *analyzer* con el cual se puede indicar como se desea hacer el análisis, un ejemplo de esta representación es la siguiente:

corpus=['Machine is my favorite subject', 'This example is useful']

Table 4. *Features names* con TfidfVectorizer por palabra

'example'	'example is'	'favorite'	'favorite subject'	'is'
'is my'	'is useful'	'machine'	'machine is'	'my'
'my favorite'	'subject'	'this'	'this example'	'useful'

Table 5. Algunos *Features names* con TfidfVectorizer por caractér

'_'	'_e'	'_ex'	'_exa'	'_f'	'_fa'	'_fav'
'_i'	'_is'	'_is_'	'_m'	'_my'	'_my_'	'_s'
'_su'	'_sub'	'_u'	'_us'	'_use'	'a'	'ac'

De acuerdo con la tabla (4) se obtiene que los resultados por **palabra** con la función de "TfidfVectorizer" implica una combinación de n-gramas por palabras, en este caso se utilizó un rango $(1, 2)$, por lo cual primero se forma un unigrama y posteriormente se calcula el bigrama con la palabra consecutiva. En contraste, la tabla (5) muestra la creación de n-gramas por **caracteres** en un rango de $(1, 4)$ por lo que forman progresivamente los conjuntos de 1 hasta llegar al rango máximo para posteriormente elegir otra posición y crear nuevamente los n-gramas hasta completar el corpus.

En tercer lugar, se plantearon 6 modelos de regresión logística para realizar las predicciones asociadas a cada categoría de toxicidad. La función para aplicar el modelo de regresión es "LogisticRegression" del paquete de "sklearn" y se emplea como parámetro solver **sag** para realizar la optimización por medio "Descenso de gradiente medio estocástico", el cual es utilizado generalmente cuando el conjunto de datos es grande. Adiconalmente, se utiliza un **cross validation** de 3 y un **score** a optimizar **roc_auc.**

Para determinar que tipo de **split** por Ngramas era el apropiado se decidió evaluar el modelo en 3 escenarios de **split** (por caracteres, por palabra y en conjunto), y para determinar que escenario arrojaba mejores predicciones se comparó el promedio de la métrica **roc_auc**, la cual representa el área bajo la curva **ROC** y ayuda a determinar si el modelo tiene buena precisión.

[§§] Matthew J. Lavin, Analyzing Documents with TF-IDF (Mayo 2019)

Table 6. CV Scores para las predicciones de toxicidad por etiqueta

Etiqueta	Ngrams/char	Ngrams/word	Ngrams/char&word
Toxic	0.9728	0.9705	0.9780
Severe Toxic	0.9880	0.9849	0.9885
Obscene	0.9865	0.9851	0.9903
Threat	0.9835	0.9850	0.9885
Insult	0.9799	0.9779	0.9825
Identity hate	0.9806	0.9748	0.9825
Total score	**0.9819**	**0.9797**	**0.9850**

Note: Para la determinación de los CV scores no se realizó el pre-procesamiento con *stop words*.

Una vez evaluados los tres escenarios de la tabla (6) se decidió escoger la metodología que contempla ambos tipos de vectorización(por palabra y por caracteres al mismo tiempo), debido a que todas las predicciones resultan tener un mejor **CV score** y por lo tanto una mejor aproximación.

Otro punto importante a destacar en el pre-procesamiento del texto son las **stop words** que serían removidas del corpus para realizar el análisis, las cuales son un **set** de palabras más comunes que se utiliza para limpiar los textos, algunos ejemplos de estas palabras comunes del ingles son *{"the", "a", "an", "in"}*.

Por un lado se realizó el análisis sin remover **stop words** y se comparó con el modelo que remueve las palabras. En este punto se realizó una comparación de las curvas ROC para determinar si era un punto relevante para el análisis el quitar o no las **stop words** del corpus, ya que existen comentarios tóxicos que son compuestos de algunas de estas palabras, por ejemplo el comentario *"Fuck you"* incluye una palabra que seria removida para el análisis.

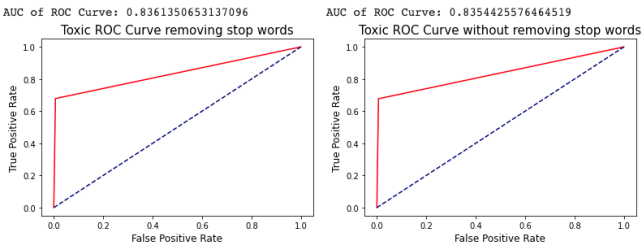

Fig. 9. Comparación gráficas curvas ROC para la etiqueta "Toxic" sin remover stop words y removiendo stop words

De acuerdo a la figura (9) se puede concluir que el modelo tiene buen desempeño en ambos escenarios, ya que su **AUC** es cercano a 1. Sin embargo, el modelo que remueve **stop words** tiene mejores mediciones porque su **AUC** es mayor en comparación con el otro escenario.

B.2. Modelo basado en redes recurrentes LSTM con entrenamiento de capa embedding. Para atacar el problema de predecir simultáneamente si un comentario emitido por los usuarios de Wikipedia Talks pertenece a la 6 etiquetas empleadas para denotar el nivel de toxicidad, siguiendo la línea de https://www.kaggle.com/sbongo/do-pretrained-embeddings-give-you-the-extra-edge, se propuso un modelo basado en la red recurrente de tipo LSTM.

Ello en razón de que, como se ha mencionado previamente, tales modelos son útiles en el procesamiento de texto, dado que la estructura semántica del lenguaje otorga cierto comportamiento secuencial que puede ser incorporado que puede ser procesar para considerar en la predicción de palabras.

Capa	Parámetros
Encaje	2,560,000
LSTM	45,360
Global MaxPooling	-
Densa 1	3,050
Densa 2	306
Total	2,608,716

Table 7. Cantidad de parametros por capa del modelo basado en redes recurrentes LSTM con entrenamiento de capa de encaje

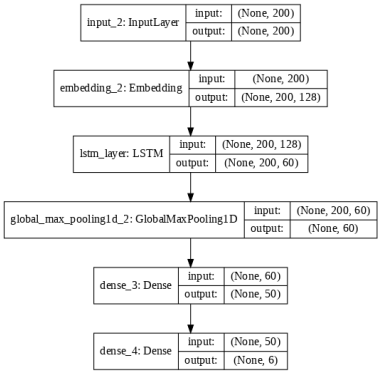

Fig. 10. Diagrama de la red propuesta para predecir simultáneamente las categorías a partir del texto, obtenido con la libreria $plot\_model$ de $keras.utils$

Para tal efecto, el texto de los comentarios fue procesado considerando los siguientes principios:

- El texto fue transformado en una representación vectorial del corpus de texto que usa conteos de palabras para reportar, convirtiendo cada texto en una secuencia de enteros (donde cada entero es el índice de un token en un diccionario) un vector donde el coeficiente para cada token es binario, basado en el conteo de palabras, empleando la librería $Tokenizer$ de $keras.preprocessing.text$.
- Del vocabulario, se consideraron únicamente a las 20,000 palabras más frecuentes para conformar la representación vectorial del punto anterior, ,
- La longitud máxima de las secuencias de texto a considerar se fijo en un valor de 200.

En línea con lo anterior, la arquitectura consideró en primera una capa de encaje de dimensión 200×128, la cual se encuentra conectada a una red LSTM[¶¶]. Posteriormente, sigue una capa que realiza un *max pooling global*[***], a la que le siguen, respectivamente un capa denso de 50 salidas (con función de activación relu) y otra de 6 salidas (con función de activación sigmoide).

Además para ajusta el modelo y entrenarlo se hicieron las siguientes consideraciones:

[¶¶] En este caso, a través del parámetro $return\_sequences$ se especificó que este modelo debería regresar un predicción para secuencia recibida

[***] Se refiere al valor máxima de la secuencia obtenida en el paso previo

Modelo	Con stopwords	Sin stopwords
Pérdida entrenamiento	0.0053	0.1416
Accuracy entrenamiento	0.9982	0.9636
Pérdida prueba	0.1235	0.1455
Accuracy prueba	0.9777	0.9622
Tiempo promedio por época	$\sim$ 6 min	$\sim$ 6 min

Table 8. Resultados del modelo basado en redes recurrentes LSTM con entrenamiento de capa de encaje

- Uso de una función de pérdida de entropía cruzada binaria, dado que las etiquetas que denotan toxicidad usan una codificación binaria,
- Como método de optimización se empleó el método de descenso de gradiente estocástico,
- Se realizó un entrenamiento del modelo considerando lotes (batches) de tamaño 256, a través de 5 épocas;,
- Se evaluó su desempeñeo contra un split de los datos de entrenamiento (en una relación de 80% y 20%),
- Se registro el valor de la métrica *accuracy* y la pérdida en dichos conjuntos.

A continuación se presentan los resultados obtenidos con la arquitectura y consideraciones metodológicas recién descritas. Cabe destacar que para probar el efecto del procesamiento de texto, se decidió ajustar el modelo manteniendo los stopwords y también en el caso en que tales se desecharon.

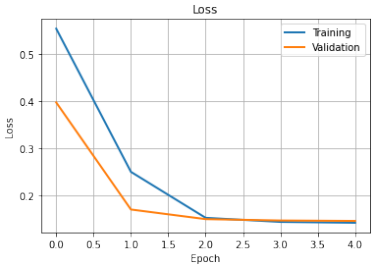

Fig. 11. Curvas de pérdida en el conjunto de entrenamiento y validancián, en el caso en que los mensajes desecharon los stopwords

Al respecto se destacan los siguientes: i) en ambos casos se obtuvo un desempeño menor en el conjunto de prueba que en el conjunto de entrenamiento, ii) ambos modelos obtienen desempeño con accuracy superiores a 0.96, por lo que se estima que son buenos indicios respecto a que este enfoque puede servir para predecir si un comentario de un usuario de Wikipedia Talks ejerce un comportamineto tóxico hacia otros; iii) al comparar el desempeño del modelo que fue entrenado considerando el procesamiento del texto para mantener o quitar los stopwords, se aprecia que al conversar estos últimos se observó un valor de accuracy, en 10^{-2} órdenes de magnitud más alto [†††], iv) el tiempo promedio de entrenamiento por época fue cercano a los 6 minutos empleando la plataforma de Google Colab.

[†††] Este punto llama la atención, dado que, como es bien sabido, algunos de los discursos de amenaza, abuso o insulto se acompañan de stopwords como ataques hacia una persona. Dicha observación sugiere que se deben hacer consideraciones del procesamiento de texto para atacar el problema de identificación de comentarios "toxicos".

B.3. Modelo basado en redes recurrentes LSTM con capa embedding pre-entrenada. Este modelo se basa en la arquitectura de la sección anterior, sin embargo se utiliza una capa de embedding pre-entrada utilizando el corpus de texto de Google Negative News con el objetivo de que estas palabras negativas puedan ser de ayuda para contextualizar el problema en el ambiente "tóxico".

La librería utilizada para esta sección fue gensim.models.keyedvectors o mejor conocida como word2vec (de la cual se ha hablado en secciones previas), la cual se refiere al modelo de redes neuronales superficiales que está capacitado para reconstruir contextos lingüísticos de palabras ya que toma como entrada un corpus de texto y a partir de ahí produce un espacio vectorial, en este caso el corpus de texto esta relacionado con Google Negative News y los comentarios.

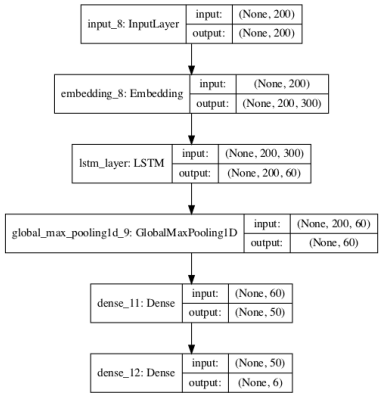

Fig. 12. Diagrama de la red propuesta para predecir simultáneamente las categorías a partir del texto, obtenido con la libreria $plot\_model$ de $keras.utils$

La idea principal es entrenar un modelo en el contexto de cada palabra, por lo que palabras similares tendrán representaciones numéricas similares. Lo anterior se logra dividiendo las oraciones en palabras(tokenizar) y crear varios pares de palabras que en este caso son 300, los pares se refieren a una variable dependiente y otra independiente, alimentamos la palabra independiente en la red neuronal a través de una capa de inclusión inicializada con pesos aleatorios, a continuación una vez que se obtuvieron los pesos del embedding procedemos con la construcción de la red neuronal LSTM de la sección anterior, cuyos detalles se muestran en la Figura 12.

El modelo se ajusto tanto para el caso en que se decidió mantener las stopwords como el caso en estas se desecharon.

La figura 13 muestra los resultados obtenidos con la arquitectura y desechando stopwords.

Por otro lado la tabla 9 muestra los resultados dependiendo si se desecharon o no las stopwords, teniendo mejor desempaño el caso en que las stopwords se retiran ya que la precisión tanto de entrenamiento como de prueba es mayor, por otro lado las pérdidas son menores.

Al respecto se destacan lo siguiente: i) en ambos casos se obtuvo un desempeño menor en el conjunto de entrenamiento que en el conjunto de prueba, ii) ambos modelos obtienen desempeño con accuracy superiores a 0.97, por lo que se observa que este enfoque puede servir para predecir si el comentario de un usuario de Wikipedia Talks ejerce un comportamineto tóxico hacia otros, además es ligeramente mejor el desempeño a comparación del modelo de la sección anterior, lo que indica

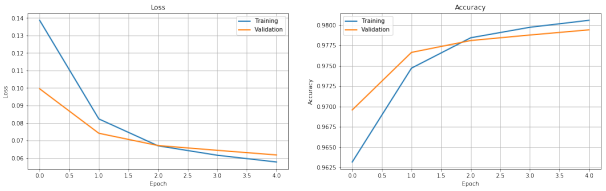

Fig. 13. Curvas de pérdida y precisión en el conjunto de entrenamiento y validanción, en el caso en que los mensajes desecharon los stopwords

Modelo	Con stopwords	Sin stopwords
Pérdida entrenamiento	0.0612	0.0578
Accuracy entrenamiento	0.9796	0.9806
Pérdida prueba	0.0633	0.06178
Accuracy prueba	0.9787	0.9794
Tiempo promedio por época	$\sim$ 6 min	$\sim$ 16 min

Table 9. Resultados del modelo basado en redes recurrentes LSTM con capa embedding pre-entrenada

que agregar un capa de encaje pre-entrenada es de utilidad ; iii) al comparar el desempeño del modelo que fue entrenado y considerando el procesamiento del texto para mantener o quitar los stopwords, se aprecia que al quitar estos últimos se obtuvo un valor de accuracy, en 10^{-2} órdenes de magnitud más alto en *train* y 10^{-3} órdenes de magnitud más alto en *test*. Dicha observación sugiere que se deben hacer consideraciones del procesamiento de texto para atacar el problema de identificación de comentarios "toxicos", iv) el tiempo promedio de entrenamiento por época fue cercano a los 12 minutos empleando recursos de una computadora con características: MacBook Pro, Procesadros 2.4 GHz Intel Core i5, Memoria 8 GB.

3. Conclusiones

A. Generales.

- En este documento se presentó una propuesta de modelos basados en métodos de procesamiento de texto y aprendizaje de máquina para atacar el problema de predicción de etiquetas de toxicidad a partir del texto vertido por usuarios de Wikipedia Talks.

- A partir de dicha información se realizó un análisis exploratorio tanto del léxico empleando en los diferentes mensajes, segmentado de acuerdo a la etiqueta de toxicidad. Asimismo,

- Se presentaron esencialmente dos enfoques 1) basado en regresiones logísticas, mediante el uso de n-gramas de caracteres y la medida $td - idf$, así como 2) el uso de modelos de aprendizaje profundo que emplean un encaje para traducir la codificación las palabras de los diferentes mensajes para que fueran procesados oportunamente por modelos matemáticos. En el segundo caso, se probó un encaje pre-entrenado word2vec-GoogleNews-vectors.

- Se evaluó el desempeño de modelos propuestos, obteniéndose resultados con un desempeño aceptable empleando ambos enfoques.

- Finalmente, los resultados obtenidos muestran que modelos bajo tales consideraciones podrían dar la pauta para implementar sistemas que permitan intervenir antes la presencia de mensajes que perjudiquen el entorno de una comunidad.

B. Lecciones aprendidas.

- Aunque la teoría puede darnos indicios de como resolver un problema, se requiere tiempo y discusión para aterrizar el proceso creativo en código realista. Sin embargo, una buena base teórica y visión para el proceso de programación son claves para el éxito de los proyectos

- No se puede pensar solo en el diseño de la implementación, sin tener en cuenta la infraestructura necesaria para lograrlo.

- Las herramientas de cómputo en la nube (Google Colab) se pueden aprovechar en nuestro beneficio para resolver problemas complejos, con soluciones que pueden ser compartidas y replicadas fácilmente a través de estos medios.

References

Dunn, R. (2020). *Multidisciplinary Perspectives on Media Fandom*. Advances in Religious and Cultural Studies. IGI Global.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Kaggle (2018). Toxic Comment Classification Challenge. Identify and classify toxic online comments . *https://tinyurl.com/y7qmd8lm*.

Tunguz, B. (2018). Toxic Comment Classification Challenge. Identify and classify toxic online comments . *https://www.kaggle.com/tunguz/logistic-regression-with-words-and-char-n-grams*.

Wikimedia (2016). Harassment Survey. Results Report. Support and Safety Team of Wikimedia Foundation. *https://tinyurl.com/y7ev4z6g*.

Wikimedia (2020). https://www.wikimedia.org. *Sitio electronico*.

Wikipedia (2020). https://es.wikipedia.org/wiki/Wikipedia. *Sitio electronico*.