CIMAT

Centro de Investigación en Matemáticas, A.C.

# Detección de Agresividad en tuits escritos por mexicanos en español

**T E S I S**

Que para obtener el grado de

**Maestra en Cómputo Estadístico**

**Presenta**

María Guadalupe Garrido Espinosa

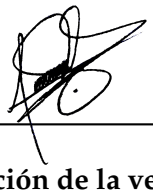
**Director de Tesis:**

Dr. Adrián Pastor López Monroy

**Autorización de la versión final**

**Co-director de Tesis:**

Dr. Alejandro Rosales Pérez



**Autorización de la versión final**

Monterrey, Nuevo León, Julio de 2021

*"¡Den gracias al Señor porque es bueno,*

*porque es eterno su amor!*

*Diga el pueblo de Israel: es eterno su amor.*

*Diga la descendencia de Aarón: es eterno su amor.*

*Digan los que respetan al Señor: es eterno su amor.*

*En la angustia clamé al Señor;*

*él me atendió y me sacó de apuros ..."*

Sal 118,1-5

# Agradecimientos

A Agus, Ale, Emmy y Alejandro, por el soporte que me dieron, por su paciencia y sobre todo, por su amor, sin el cual este proyecto no se habría llevado a cabo. Gracias a Emily y Emiliano, con su cariño y energía cargaron mis pilas cuando estaban agotadas.

A Jorge y Ana, mis palabras se quedan cortas para expresar el agradecimiento que les tengo, ¡gracias por brindarme su amistad!

A la Srita. Blanca Garza, por cuidar de mi y ser mi compañía especialmente en tiempos de pandemia.

Al Dr. Adrián Pastor López Monroy y al Dr. Alejandro Rosales Pérez, por sus recomendaciones, orientación y paciencia durante este trabajo de tesis.

A los Dres. José Ulises Márquez y José Jaime Hérnandez, por sus consejos y su voto de confianza en mi persona.

Finalmente, quiero agradecer al Centro de Investigación en Matemáticas A.C. (CIMAT) unidad Monterrey y al Consejo Nacional de Ciencia y Tecnología (CONACYT), por la beca número 718246 y el apoyo que me otorgaron, ambos me permitieron hacer más que sólo una maestría.

# Resumen

Las redes sociales muestran cada día formas más innovadoras de comunicación y se han convertido en un canal donde las personas expresan su opinión, se difunde información y se discuten temas relevantes para la sociedad.

Al mismo tiempo, al permitir el anonimato, existen facilidades para que los usuarios de las redes sociales ataquen y ofendan a otras personas. Este hecho se vuelve relevante por el impacto que tiene la propagación de mensajes agresivos, delitos violentos, el acoso en línea, etc. Por lo anterior, tanto los propietarios de las redes sociales como académicos han trabajado en detectar mensajes agresivos.

En esta tesis se abordará la detección de tuits agresivos, para ello se evaluarán características como los N-gramas con Máquinas de Soporte Vectorial, Redes Convolucionales, Recurrentes y además, se incluyen características del perfilado de autor. De igual forma, se profundiza en la forma en cómo se ingresa esta información a los métodos desarrollados para probar si las personas agreden de forma distinta dependiendo de su perfil. En el desarrollo de este trabajo se verá que agregar el género, la localización y la ocupación permite mejorar la discriminación entre mensajes no agresivos y agresivos.

# Abstract

Social media show more innovative ways of communication every day and have become in a channel where people express their own opinion, their information is disseminated and where the society discusses relevant topics.

At the same time, by allowing anonymity, social media allows users to attack and offend other people. This fact becomes relevant due to the impact of the propagation of aggressive messages, violent crimes, online harassment, etc. Therefore, both the owners of social networks and academics have worked to detect aggressive messages.

This thesis will addresses the detection of aggressive tweets using characteristics such as N-grams with Machine Learning and Deep Learning methods. In addition, characteristics of author profiling are included and we explore the best way to aggregate this information into the methods developed to test if people attack differently depending on their profile. In the development of this work, it will be seen that adding gender, location and occupation allows improving the discrimination between non-aggressive and aggressive messages.

# Índice general

# Índice de figuras

# Capítulo 1

# Introducción

El internet y la tecnología actual ofrecen innovadoras y variadas formas de participación dentro de la sociedad. Las redes sociales, al hacer uso de ambos, se han convertido en un canal de comunicación, así como un espacio donde las personas pueden discutir, difundir información y expresar su opinión.

A la par, las redes sociales, al permitir el anonimato, hacen que sea más sencillo el atacar u ofender a otras personas. Quienes realizan este tipo de actos carecen del contacto visual cara a cara o del riesgo de ser desprestigiados porque fácilmente se puede fingir un nombre. A pesar que el contenido ofensivo puede eliminarse de forma rápida y sencilla, éste es transmitido a una amplia audiencia en segundos.

Se estima que empresas de redes sociales como Facebook, Twitter y YouTube invierten cientos de millones de euros cada año en acciones para combatir una versión extrema de agresividad: el discurso de odio, sin que todas estas acciones resulten suficientes (Gambäck y Sikdar, 2017).

Uno de los aspectos más importantes para detectar la agresividad en las redes sociales está en el efecto que los mensajes agresivos en estas plataformas pueden llegar a tener sobre las personas. Por un lado, Müller y Schwarz (2019) sugieren que las redes sociales pueden actuar como un mecanismo de propagación entre

los mensajes en línea y los delitos violentos. Por otro lado, el Departamento de Salud y Servicios Humanos de Estados Unidos de América, mediante su página stopbullying[1], indica que las redes sociales son uno de los lugares más comunes para realizar ciberacoso. En consecuencia, se vuelve relevante detectar mensajes agresivos de forma automática.

Sin embargo, detectar la agresividad en redes sociales puede llegar a ser una tarea difícil, no sólo por la cantidad de información que se comparte minuto a minuto -en Twitter, por ejemplo, se comparten 350 mil tuits en promedio, por minuto[2]-, sino también porque no existe una definición formal del lenguaje agresivo, ni es sencillo diferenciar esta tarea de otras con mayor complejidad como el discurso de odio y el ciberacoso. El lenguaje agresivo puede definirse como aquel que "busca dañar o lastimar a un grupo o individuo al referirse o incitar a la violencia"[3]. Por otra parte, de acuerdo con Nockleby (2000), en el discurso de odio se "ataca a una persona o grupo sobre la base de atributos tales como raza, religión, origen étnico, origen nacional, sexo, discapacidad, orientación sexual o identidad de género".

El interés en este tema ha ido en aumento y ha suscitado la organización de talleres y foros para encontrar soluciones. Algunos de los talleres y foros se enfocan en el idioma inglés, alemán o griego, por ejemplo:

- El tercer taller sobre lenguaje abusivo en línea (ALW3)[4]. Incentiva el presentar artículos donde se muestren modelos y métodos de Procesamiento de Lenguaje Natural (NLP, por sus siglas en inglés) para detectar lenguaje abusivo en línea, incluidos, entre otros, discurso de odio, acoso cibernético, etc. En el contexto de este taller también se alienta a los participantes a probar sus

[1] https://www.stopbullying.gov/cyberbullying/what-is-it

[2] De acuerdo con *Internet live Stats*, https://www.internetlivestats.com/twitter-statistics/

[3] Definición usada en la edición 2020 del MEX-A3T https://sites.google.com/view/mex-a3t/tracks/AI?authuser=0

[4] https://sites.google.com/view/alw3/

experimentos en mensajes en inglés y alemán.

- El Segundo Taller sobre *trolling*, agresión y *cyberbullying*. Este taller promueve conversaciones dedicadas a la detección automática de la agresión, tanto en el habla como en el texto[5].

En español, se pueden encontrar algunos otros foros:

- Identificación automática de misoginia (AMI, por sus siglas en inglés)[6]. Propone la identificación de la misoginia, la categorización del comportamiento misógino y la clasificación de objetivos tanto en español como en inglés.

- Taller internacional sobre evaluación semántica 2019 (SemEva)[7]. Dentro del taller se realizó la competencia llamada Detección multilingüe de discurso de odio contra inmigrantes y mujeres en Twitter (hatEval, por sus siglas en inglés)[8], la cual consiste en la detección de odio en Twitter presentado por dos objetivos diferentes, inmigrantes y mujeres, en una perspectiva multilingüe, para español e inglés.

- El Análisis de autoría y agresividad (MEX-A3T)[9]. El objetivo es avanzar en el estado del arte en el análisis no temático de textos breves escritos en español mexicano. Una de las tareas propuestas es discriminar entre tuits agresivos y no agresivos escritos en español mexicano.

El MEX-A3T resulta de particular interés respecto de las demás competencias por varias razones: 1) los tuits están escritos en español, 2) la variante de español es mexicano y, 3) dado el poco estudio para el español, existe espacio suficiente para mejorar la detección de agresividad en esta variante mediante el uso de técnicas

[5] https://sites.google.com/view/trac2/home?authuser=0
[6] https://amievalita2020.github.io/
[7] http://alt.qcri.org/semeval2019/index.php?id=tasks
[8] https://competitions.codalab.org/competitions/19935
[9] https://sites.google.com/view/mex-a3t/home?authuser=0

de procesamiento de lenguaje natural. Aragón y cols. (2019), en su revisión de las propuestas para detectar la agresividad en el MEX-A3T 2019, describen el por qué esta labor es compleja: existen mensajes que son agresivos sin palabras vulgares, otros son irónicos y otros más, usan palabras que están fuera del vocabulario de entrenamiento. Por ejemplo:

> *Y hablando de cosas feas, ¿cómo está tu novia?*. Este tuit es irónico.

> *Ponte a correr gorda, está bien que las puertas del gimnasio de abren*. Y este tuit no contiene palabras vulgares, pero es agresivo.

Algo interesante de la edición 2019 del MEX-A3T es que, el equipo ganador, Casavantes, López, y González (2019), incluyó la ocupación y la localización dentro de su sistema para explorar si existen diferencias en el vocabulario dependiendo del perfil del autor del tuit. Intuitivamente, si se sabe que, por ejemplo, una persona tiene un trabajo relacionado con las ciencias exactas, ¿esto resulta de ayuda para saber si los tuits que escribe son agresivos? Los resultados de éste mostraron que, en el mejor de los casos, los cambios son casi imperceptibles.

En el presente trabajo se abordará la detección de tuits agresivos en el marco de la competencia MEX-A3T 2020, para ello se emplearán características como los n-gramas con aprendizaje máquina y métodos de aprendizaje profundo. Además, se retoma la idea propuesta por Casavantes y cols. (2019) de incluir características del perfilado de autor, pero variando la forma en cómo se ingresa esta información a los métodos desarrollados y más adelante se verá que incorporar información adicional como el género, la localización y la ocupación permite mejorar la discriminación entre mensajes no agresivos y agresivos.

## 1.1. Objetivo

Proponer un método que, tomando ventaja del aprendizaje profundo, características tradicionales de texto y características de perfilado de autor como el género, la localización y la ocupación, pueda aprender un modelo que supere métodos de referencia en la detección de agresividad en el corpus del MEX-A3T.

### 1.1.1. Objetivos específicos

I) Generar un modelo con un algoritmo de línea base de aprendizaje máquina que tenga como entrada las características tradicionales para clasificación de texto.

II) Adaptar métodos de aprendizaje profundo que puedan capturar la agresividad en tuits escritos en español.

III) Replicar modelos de la literatura para pronosticar características de perfilado de autor como género, localización y ocupación de los autores de los tuits para posteriormente usarlos como características en los métodos descritos en los puntos uno y dos.

IV) Evaluar la utilidad las características de perfilado de autor en los métodos de aprendizaje profundo y aprendizaje máquina para evaluar su utilidad en la detección de agresividad.

## 1.2. Organización de la tesis

La tesis está organizada de la siguiente manera: en el Capítulo 2 se presenta el marco teórico requerido para que este trabajo sea lo más autocontenido posible. Luego, dentro del Capítulo 3 se presenta el trabajo relacionado con la detección de

agresividad que se ha realizado con anterioridad. Enseguida, en el Capítulo 4 se describe el método de aprendizaje máquina, aprendizaje profundo y de perfilado de autor usado en esta tesis. En el Capítulo 6 muestran los experimentos más relevantes y sus resultados. Finalmente, en el Capítulo 7 se dan las conclusiones y el trabajo futuro.

# Capítulo 2

# Marco Teórico

El NLP es un conjunto de técnicas computacionales para analizar y representar texto producido de forma natural. Su propósito es lograr un procesamiento del lenguaje similar al humano para una variedad de tareas (Liddy, 2001).

Dentro de las tareas abordadas por el NLP está la clasificación de texto. Ésta consiste en que, dado un texto o documento de cierto tipo, se pueda decidir a qué clase -dada con antelación- pertenece. Algunos ejemplos de esta tarea son la identificación del lenguaje, la determinación del autor de un texto (*atribución de autoría*), o incluso características específicas, como por ejemplo el género, la edad, la localización, ocupación, etc., (*perfilado de autor*). Este trabajo de tesis se centra en determinar si un texto es agresivo o no.

En este capítulo se introducen los conceptos básicos que sustentan el trabajo realizado para la detección de agresividad. En esencia, se divide en tres partes: la descripción del funcionamiento de una Máquina de Soporte Vectorial (SVM, por sus siglas en inglés) lineal, la exposición de los métodos de aprendizaje profundo empleados y, finalmente, en la Sección 2.5 se describe la definición de la métrica $F_1$, usada para evaluar el desempeño de los modelos propuestos.

## 2.1. El problema de clasificación binaria

Dado un conjunto de entrenamiento:

$$\mathcal{L} = \{(x_i, y_i) : i = 1, 2, \ldots, L\} \tag{2.1}$$

donde $x_i \in \mathcal{R}^k$ y $y \in \{-1, +1\}$. El problema de clasificación binaria es usar $\mathcal{L}$ para construir una función $f : \mathcal{R}^k \to \mathcal{R}$ tal que:

$$\phi(x) = signo(\gamma(x)), x \in \mathcal{R}^k \tag{2.2}$$

donde $\phi$ es un clasificador. Si $\gamma$ tiene valor positivo, $\gamma(x) > 0$, entonces $x$ se clasifica como perteneciente a la clase positiva ($y = +1$). Mientras que si $\gamma(x) < 0$, se clasifica como perteneciente a la clase negativa ($y = -1$).

## 2.2. Máquina de Soporte Vectorial

La SVM, introducida por Vapnik (1996), es un clasificador lineal entre dos clases que busca encontrar el hiperplano óptimo que maximice el margen de separación entre las clases.

### El caso linealmente separable

Supóngase que los datos disponibles del conjunto $\mathcal{L}$ pueden ser separados por un hiperplano,

$$\Gamma = \{x : \gamma(x) = w^T x + b = 0\} \tag{2.3}$$

si el hiperplano separa el conjunto de datos de entrenamiento en las dos clases sin error, al hiperplano se le conoce como separador. Si además maximiza la distancia

a los puntos más cercanos de ambas clases, es decir, si maximiza

$$\frac{2}{\|w\|} \tag{2.4}$$

se dice que es de margen máximo.

Ahora bien, si los datos del conjunto de entrenamiento son linealmente separables, entonces existe $w$ y $b$ tales que

$$y_i(w^T x_i + b) \geq 1, \;\; i = 1, \ldots, L \tag{2.5}$$

De esta forma, el problema se convierte en encontrar el hiperplano que maximice el margen, $\frac{2}{\|w\|}$ , sujeto a la condición de la ecuación 2.5. De forma equivalente, se desea encontrar $w$ y $b$ tales que

$$\min_{w,b} \qquad \frac{1}{2}\|w\|^2, \tag{2.6a}$$

$$\text{tal que} \qquad y_i(w^T x_i + b) \geq 1, \;\; i = 1, \ldots, L \tag{2.6b}$$

Esta formulación es un problema de optimización cuadrática puesto que la función objetivo es cuadrática y la restricción es lineal. La solución a este problema es única siempre que el hiperplano exista.

Cabe señalar que, para el par óptimo, $\{w^\star, b^\star\}$, algunos puntos del conjunto de entrenamiento cumplirán

$$y_i(w^{\star T} x_i + b^\star) = 1 \tag{2.7}$$

a estos puntos se les conoce como los vectores de soporte y son importantes porque definen al hiperplano.

El problema de la ecuación 2.6 puede ser resuelto mediante el uso de los multiplicadores de Lagrange,

$$F(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{L} \alpha_i \{y_i(w^T x_i + b) - 1\} \tag{2.8}$$

donde

$$\alpha = (\alpha_1, \ldots, \alpha_L)^T \geq 0 \tag{2.9}$$

es el vector de los coeficiente no negativos de Lagrange. Una vez que se minimiza $F$ respecto de las variables primales $w$ y $b$, se requiere maximizar el mínimo resultante respecto de las variables duales $\alpha$.

Con las condiciones de Karush-Kuhn-Tucker se obtienen las condiciones necesarias y suficientes para encontrar la respuesta a este problema de optimización[1]. El valor máximo de F está dado por:

$$\sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i \alpha_j y_i y_j (x_i^T x_j) \tag{2.10}$$

El siguiente paso es encontrar los multiplicadores $\alpha$ maximizando la ecuación 2.10 sujeto a la restricción mostrada en la ecuación 2.9 y a que

$$\sum_{i=1}^{L} \alpha_i y_i = 0 \tag{2.11}$$

A este problema también se le conoce como la dualidad de Wolfe. Si $\alpha^\star$ es solución a este problema, entonces el óptimo de $w^\star$ es

$$w^\star = \sum_{i \in sv} \alpha_i^\star y_i x_i \tag{2.12}$$

donde $sv$ es el conjunto de todos los vectores soporte, es decir, $w^\star$ es una función lineal de estos vectores. En tanto, el óptimo de $b^\star$ está dado por

$$b^\star = -\frac{1}{2}(w^{T\star} x_+ + w^{T\star} x_-) \tag{2.13}$$

donde $x_+$ es cualquier vector soporte de la case positiva y $x_-$ es cualquier vector soporte de la clase negativa.

[1] Para mayores detalles ver el capítulo 11 de Izenman (2008),

**SVM con restricciones suaves**

Puede darse el caso en el que cualquiera de las dos clases sean separables, pero no linealmente, por tanto, no existe un hiperplano separador de las clases y por ende, no hay solución al problema de optimización. Sin embargo, este problema se puede solucionar relajando las restricciones y con la introducción de variables de holgura:

$$\min_{w,b} \quad \frac{1}{2} w^T w + C \sum_{i=1}^{L} \xi_i \tag{2.14a}$$

$$\text{tal que} \quad \forall i \;\; y_i(w^T x_i + b) \geq 1 - \xi_i \tag{2.14b}$$

$$\forall i \;\; \xi_i \geq 0 \tag{2.14c}$$

La variable de holgura $\xi_i$ permite que la entrada $x_i$ esté más cerca del hiperplano o incluso que esté mal clasificada, pero aplica una penalización en la función objetivo. Si el valor de $C$ es muy grande, la SVM es estricta y fuerza a que todos los puntos estén en el lado correcto del hiperplano. Por el contrario, si $C$ tiene un valor muy pequeño, la SVM se relaja y algunos puntos los clasifica erróneamente con tal de obtener una $\|w\|$ más baja.

## 2.3. Métodos de aprendizaje profundo

En esta sección se abordarán los métodos de aprendizaje profundo empleados en la tesis. Comienza explicando cómo se representan las palabras por medio de vectores en la subsección 2.3.1, continúa explicando qué son y cómo funcionan las redes convolucionales para textos en la subsección 2.3.2 y en la subsección 2.3.3 se explica brevemente qué son las redes recurrentes y se aborda la RNN empleada en este documento, las *Gated Recurrent Units*.

### 2.3.1. Representación de palabras como vectores.

Cuando se trabaja en NLP, una de las las preguntas más comunes que surgen es, ¿cómo representar las palabras en un modelo? Idealmente, esta representación debe reflejar qué tan similares son unas palabras de otras. Una forma simple de hacerlo sería, dado un corpus con $V$ palabras únicas, éstas puede ordenarse y cada palabra se representa como un vector de dimensión $\mathbf{R}^{|V|\times 1}$ con todas sus entradas iguales a cero excepto en el índice correspondiente a la palabra, donde tiene un valor iguala 1, esto es a lo que se le conoce como *one hot encoding*. Sin embargo, esta representación es poco útil pues no da una noción de qué tan similares son las palabras: el producto punto de vectores *one hot encoding* para dos palabras distintas sería cero aún cuando éstas fueran, por ejemplo, "estudiante" y "alumno".

Por lo anteriormente descrito, se han propuesto otros modelos que representan a las palabras como vectores donde el significado de la palabra se distribuye a lo largo del vector. Bajo este supuesto, el producto punto de dos palabras distintas es diferente de cero y optimizado para que, palabras similares con contextos (conjuntos de palabras que rodean a determinada palabra) similares puedan localizarse como dos puntos cercanos. Uno de los métodos más conocidos que cae en esta rama es ***word2vec***, propuesto por Mikolov, Chen, Corrado, y Dean (2013), y consiste en dos algoritmos:

1. *Continuos Bag of Words*, tiene el objetivo de predecir una palabra dado un contexto,

2. *Skipgram*, realiza lo opuesto, predice las palabras circundantes dada una palabra.

Los vectores de palabras producidos por ***word2vec*** se entrenan con corpus de millones de palabras y suelen ser la base sobre la cual se componen modelos más

complejos. En las siguientes secciones se verá que a partir de los vectores de palabras se construyen redes convolucionales y redes recurrentes.

### 2.3.2. Redes neuronales convolucionales para textos

Las redes neuronales convolucionales inicialmente fueron usadas para procesamiento de imágenes donde mostraron ser exitosas para detectar objetos o bien, para reconocerlos de forma supervisada (Krizhevsky, Sutskever, y Hinton, 2012). Su uso en NLP fue introducido por Collobert y cols. (2011), quienes las usaron para tareas semánticas como el reconocimiento de entidades nombradas (Named-Entity Recognition) o para identificar a las palabras con una etiqueta que indique su función sintáctica (Part-Of-Speech Tagging, POS por sus siglas en inglés). Posteriormente, las CNN fueron empleadas por Kim (2014) y por Kalchbrenner, Grefenstette, y Blunsom (2014) para clasificación de texto.

En este trabajo se tomó como referencia la arquitectura propuesta por Kim (2014) debido a su simplicidad y su buen desempeño en ciertas tareas de clasificación (Gehrmann y cols., 2017; Qiu, Yoon, Fearn, y Tourassi, 2017), gráficamente, se puede ver en la Figura 2.1 . A continuación, se describe matemáticamente cómo se aplica la convolución para clasificar textos en categorías, tomando las definiciones del trabajo de Kim (2014).

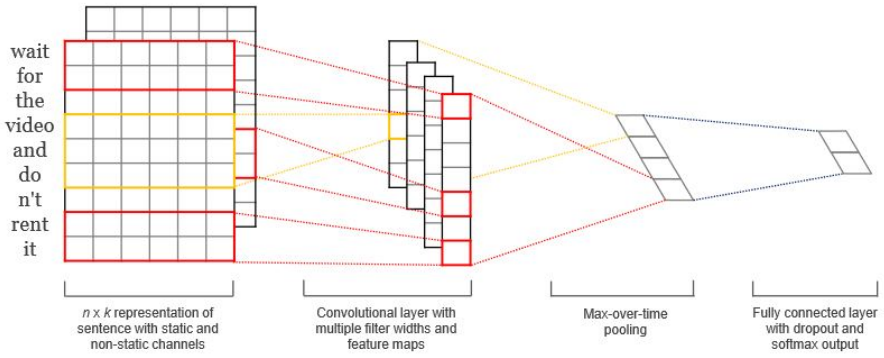



**Figura 2.1:** Arquitectura de Kim (2014) con dos canales para una oración. Comienza con una representación de los textos de tamaño $n \times k$ con canales estáticos y no estáticos. Posteriormente, se sigue con una capa convolucional con filtros de tamaños múltiples para producir diversos mapeos de características, enseguida se aplica la operación *max-overtime pooling* para finalmente realizar *dropout* y hacer las predicciones con una capa con una función de activación *softmax*.

Supóngase que se tiene un corpus con $L$ documentos en el conjunto de entrenamiento. Sea $x_i \in \mathbf{R}^k$ un vector de dimensión $k$ que representa a la i-ésima palabra en una oración de tamaño n, es decir, la oración que cuenta con n palabras. De esta forma, Kim (2014) representa la oración como

$$\mathbf{x_{1:n}} = \mathbf{x_1} \oplus \mathbf{x_2} \oplus \cdots \oplus \mathbf{x_n} \tag{2.15}$$

donde el operador $\oplus$ se entiende como la operación de concatenar los vectores asociados a las palabras de la oración. Cuando se vea la expresión $x_{i:i+j}$ se está haciendo referencia a la concatenación de los vectores $x_i, x_{i+1}, \ldots, x_{i+j}$.

Ahora, la convolución involucra un filtro $v \in \mathbf{R}^{lk}$ que es aplicado en una ventana de $l$ palabras para producir una nueva característica, $\mathbf{c_i}$ :

$$\mathbf{c_i} = f(v \cdot \mathbf{x}_{i:i+l-1} + b) \tag{2.16}$$

El término $b$ es el sesgo y f es una función no lineal como, por ejemplo, la tangente hiperbólica. Este filtro es aplicado a cada ventana de palabras existente en la oración $\mathbf{x}_{1:l}, \mathbf{x}_{2:l+1}, \ldots, \mathbf{x}_{n-l+1:n}$ para así producir un mapeo de características (*fea-*

*ture map*)

$$\mathbf{c} = [\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_{n-l+1}}] \tag{2.17}$$

con $\mathbf{c} \in \mathbf{r}^{n-l+1}$. Posteriormente, se aplica la operación *max-overtime pooling* al mapeo de características $\mathbf{c}$ y se toma el valor máximo $\hat{c} = \max c$ como la característica de este filtro. En el fondo, lo que se desea es capturar las características más importantes vía aquellas que tienen un valor mayor.

Lo anterior describe el proceso de extraer una característica empleando un filtro. Generalmente, para obtener varias características, se usan múltiples filtros de diversos tamaños. Aquellos de menor tamaño se espera que capturen patrones locales frecuentes y a medida que su tamaño se incremente, se espera que capturen patrones significativos pero poco frecuentes. Dicho de otra forma, con el proceso de sacar ventanas y convolución se obtienen de forma automática combinaciones de palabras o caracteres (n-gramas) del documento.

Las características obtenidas mediante los filtros forman la penúltima capa de la arquitectura (ver Figura 2.1) y en la última capa se usa la función *softmax* para así obtener un vector de probabilidades para las categorías.

#### La regularización

La forma de regulación que emplea Kim (2014) es es el *dropout*: suponiendo que se usaron $m$ filtros, entonces en la penúltima capa $\mathbf{z} = [\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_m]$ en lugar de usar

$$y = \nu \cdot \mathbf{z} + b \tag{2.18}$$

para obtener la unidad de salida $y$ en la propagación hacia adelante, el *dropout* usa

$$y = \nu \cdot (\mathbf{z} \circ \mathbf{dr}) + b \tag{2.19}$$

el operador $\circ$ representa la multiplicación elemento a elemento; $\mathbf{dr} \in \mathbf{R}^m$ es un

vector cuya función es "ocultar" unidades mediante variables aleatorias Bernoulli con probabilidad $p$ de ser igual a 1. Los gradientes se propagan únicamente sobre aquellas unidades que no están ocultas. Al momento de trabajar con el conjunto de prueba, los pesos de los vectores aprendidos son escalados por $p$ de forma que $\hat{v} = p\nu$ y $\hat{v}$ se usa sin *dropout* para ponderar las oraciones que no se han visto. Asimismo se restringen las normas $l_2$ de los vectores de pesos reescalando $\nu$ para tener $||\nu|| = s$ si $||\nu|| > s$ después de calcular el gradiente descendente.

**Las variaciones del modelo**

Al inicio de la sección se habló del vector $x_i$ de dimensión $k$ que representa a la i-ésima palabra de una oración. En la arquitectura, los vectores que representan a las palabras, los también llamados *word embedding vectors*, se pueden inicializar con los vectores obtenidos de forma no supervisada en alguna otra tarea para incrementar el desempeño cuando el conjunto de entrenamiento no es suficientemente grande, es decir, se usa tranferencia de conocimiento (*transfer learning*). Kim (2014) empleó algunas variaciones a partir de los pesos de estos vectores:

1. **Con pesos estáticos**. Los pesos de los vectores asociados a las palabras se mantienen fijos.

2. **Con pesos no estáticos**. Con esta variante se permite que los pesos se modifiquen durante el entrenamiento.

3. **Con pesos aleatorios**. Los pesos para todas las palabras se inicializan de forma aleatoria y posteriormente se modifican durante el entrenamiento.

### 2.3.3. Redes recurrentes

Las redes neuronales recurrentes (RNN, por sus siglas en inglés) son un conjunto de redes neuronales que sirven para procesar datos secuenciales como el texto.

En esta sección se dará una breve explicación de su funcionamiento con base en el trabajo de Chung, Gulcehre, Cho, y Bengio (2014).

Una RNN es una red neuronal que consiste en un estado oculto $h$ y una salida opcional $\mathbf{y}$; la red opera sobre una secuencia de tamaño variable, $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$. En cada tiempo $t = \{0, 1, 2, \ldots, T\}$, el estado oculto $h_t$ se actualiza es

$$h_t = \begin{cases} 0 & t = 0 \\ f(h_{t-1}, \mathbf{x}_t) & t > 0 \end{cases} \tag{2.20}$$

donde $f$ es una función de activación no lineal. La salida de la red recurrente puede también ser una secuencia $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$ de un tamaño variable, o bien, puede constar de un solo elemento, como es en el caso de la clasificación de texto.

La forma en cómo se actualiza el estado oculto actual de una RNN simple como la que se acaba de mostrar es la siguiente:

$$h_t = g(W\mathbf{x}_t + Uh_{t-1}) \tag{2.21}$$

donde $g$ es una función acotada y suave. Un ejemplo frecuentemente usado de este tipo de función es la tangente hiperbólica (Cho, Van Merriënboer, Bahdanau, y Bengio, 2014).

Una RNN puede aprender una distribución de probabilidad sobre una secuencia cuando se entrena para pronosticar el siguiente símbolo en la secuencia; por tanto, tiene la capacidad de capturar la distribución de cadenas de tamaño variable si dentro de los símbolos probables en la secuencia se emplea uno para representar el final de la misma. La probabilidad de una secuencia puede ser descompuesta como sigue

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_T) = p(\mathbf{x}_1)p(\mathbf{x}_2|\mathbf{x}_1) \cdots p(\mathbf{x}_T|\mathbf{x}_1, \ldots, \mathbf{x}_{T-1}) \tag{2.22}$$

donde el último elemento es el símbolo que indica el final de la secuencia. La pro-

babilidad condicional de cada elemento se modela como

$$p(\mathbf{x}_T|\mathbf{x}_1,\ldots,\mathbf{x}_{T-1}) = g(h_t) \tag{2.23}$$

es decir, como una función del estado oculto actual visto en la ecuación 2.21.

Una estrategia empleada al realizar la clasificación de texto es generar un vector de representación empleando una RNN. Intuitivamente, este vector resume la información secuencial en la oración, texto o documento procesado. Este vector, a su vez, es la entrada de una función *softmax* que produce la probabilidad de pertenecer a determinada clase.

### Gated Recurrent Units

Las *Gated Recurrent Units* (GRU), propuestas por Cho, Van Merriënboer, Gulcehre, y cols. (2014), son un tipo de unidad oculta, ($f$ de la ecuación 2.20). A continuación, se describirá su funcionamiento.

La salida de la j-ésima unidad GRU, $h_t^j$, también llamada la activación de la GRU en el tiempo $t$ se calcula como una ponderación entre la activación en el tiempo anterior $h_{t-1}$ y la candidata a ser la nueva activación, $\tilde{h}_t$,

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j\tilde{h}_t^j \tag{2.24}$$

$z_t^j$ funge como una compuerta de actualización que controla qué tanta información del estado oculto anterior se transmitirá al estado oculto actual. Su cálculo es como sigue:

$$z_t^j = \sigma\Big(W_z\mathbf{x}_t + U_z h_{t-1}\Big)^j \tag{2.25}$$

donde $\sigma$ es la función sigmoide, $W_z$ y $U_z$ son matrices de pesos a ser aprendidas.

La candidata a ser la nueva activación, $\tilde{h}_t$, se calcula de forma similar a como es

calculada la activación de una RNN simple (ecuación 2.21),

$$\tilde{h}_t^j = tanh\Big(W\mathbf{x}_t + U(r_t \circ h_{t-1})\Big)^j \tag{2.26}$$

donde $W$ y $U$ son matrices de pesos, $r_t$ es un conjunto de compuertas de reajuste (reset) y $\circ$ es la multiplicación elemento a elemento. El efecto que tiene la compuerta de reajuste puede verse así: cuando está cercana a cero, el estado oculto actual es forzado a ignorar el estado oculto previo y continuar únicamente con la entrada actual. De este modo, se borra información que pudiera considerarse irrelevante para el futuro.

Finalmente, la compuerta de reajuste es calculada de la siguiente manera,

$$r_t^j = \sigma\Big(W_r\mathbf{x}_t + U_r h_{t-1}\Big)^j \tag{2.27}$$

donde $W_r$ y $U_r$ son matrices de pesos a ser aprendidas.

Cabe notar que, por tener compuertas de actualización y reajuste separadas, la GRU aprende a capturar dependencias en diferentes tiempos de más largo rango. Se esperaría que las unidades que aprenden dependencias a corto plazo tengan compuertas de reajuste activas.

### 2.3.4. Modelo de Atención

Bahdanau, Cho, y Bengio (2014) introdujeron el mecanismo de atención para combinar los estados ocultos de una RNN con enfoque *encoder-decoder* en la tarea de traducción automática. La intuición detrás de este mecanismo es que no todas las palabras de un documento son igual de importantes y determinar las secciones más importantes involucra modelar las interacciones entre sus partes.

Formalmente, dado un modelo que produce un estado oculto, $h_t$, en cada periodo de tiempo, un modelo basado en la atención calcula un vector de contexto,

$a_t$, como el promedio ponderado de los estados ocultos:

$$a_t = \sum_{j=1}^{T} \alpha_{tj} h_j \tag{2.28}$$

donde cada uno de los pesos $\alpha$ se calcula en cada periodo de tiempo y para cada estado oculto. En un siguiente paso, estos vectores $a_t$ son usados para calcular $s_t$, el estado oculto de otra RNN en el tiempo $i$. Esta última a su vez depende de su estado previo $s_{t-1}$, de $a_t$ y de la salida del modelo.

Los pesos $\alpha_{tj}$ se calculan de la siguiente forma,

$$\alpha_{tj} = \frac{e_{tj}}{\sum_{k=1}^{T} e_{tk}} \tag{2.29}$$

donde

$$e_{tj} = \tau(s_{t-1}, h_j) \tag{2.30}$$

$\tau$ es una función que se aprende y depende del estado $h_j$ y de la secuencia $s$ en el tiempo $t-1$.

## 2.4. El perfilado de autor

La tarea de perfilado de autor, tiene como objetivo relacionar el estilo de escritura de un autor con sus características demográficas (Wiegmann y cols., 2019), sus preferencias políticas o incluso su personalidad (Rangel, Rosso, Montes-y Gómez, Potthast, y Stein, 2018). Para ello se emplea el contenido de los documentos que las personas comparten, pueden ser textos formales, *blogs* y, en años recientes, también se usa el contenido compartido en redes sociales como Facebook y Twitter.

Un ejemplo de la tarea de perfilado de autor es cuando, dado un texto, se desea determinar el género, la edad, la lengua materna, ocupación, etc. Este ejercicio

ha resultado ser útil en áreas como mercadotecnia donde se emplea para enviar publicidad a grupos específicos; también ha resultado eficaz en la ciencia forense donde el perfil de los autores puede ser utilizado como evidencia adicional en investigaciones criminales (Rangel y cols., 2018).

## 2.5. Métricas para la evaluación del desempeño

Para determinar qué algoritmo funciona mejor, se debe asignar una medida cuantitativa para estimar el desempeño. Para la tarea de clasificación binaria existen evaluaciones numéricas que producen un valor que generaliza el desempeño del clasificador. A continuación, se abordarán algunas de las más usadas.

Por convención, en la clasificación binaria, la etiqueta de la clase minoritaria se considera como la clase positiva mientras que la clase mayoritaria como negativa. La Figura 2.2 muestra la matriz de confusión cuando se tienen dos clases. Los renglones muestran la clase actual y las columnas su pronóstico. TP y TN denotan las observaciones de la clase positiva y de la clase negativa, respectivamente, que fueron clasificados correctamente. En tanto, FP y FN indican las observaciones que fueron clasificadas de forma errónea de la clase negativa como de la clase positiva, respectivamente.

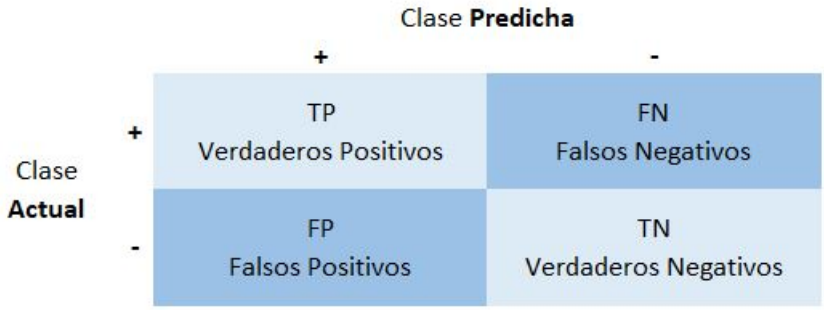



**Figura 2.2:** Matriz de confusión para dos clases.

Con base en la matriz de confusión de la Figura 2.2 se pueden definir las métri-

cas de la Tabla 2.1.

**Tabla 2.1:** Métricas para evaluar el desempeño.

| Métrica | Fórmula | Interpretación |
|---|---|---|
| Exactitud | $\frac{TP+TN}{TP+TN+FP+FN}$ | Desempeño general del modelo |
| Precisión | $\frac{TP}{TP+FP}$ | Qué tan exactas son las predicciones positivas |
| Recuerdo (*Recall*) | $\frac{TP}{TP+FN}$ | Cobertura de muestra positiva real |
| Especificidad | $\frac{TN}{TN+FP}$ | Cobertura de muestra negativa real |
| F | $\frac{(1+\beta)\cdot\text{Precisión}\cdot\text{Recuerdo}}{\beta\cdot\text{Precisión}+\text{Recuerdo}}$ | Métrica híbrida útil para clases desequilibradas |

Generalmente, la exactitud es la métrica más empleada. Sin embargo, cuando se cuenta con clases no balanceadas, la exactitud ya no es la mejor métrica para evaluar dado que no distingue las observaciones correctamente clasificadas entre las distintas clases. Por ejemplo, puede darse el caso que, al tener dos clases, la exactitud sea alta al considerar todas las observaciones, pero al observar la misma métrica dentro de cada una de las clases, ésta sea alta para la clase con mayor número de casos y muy baja para la otra clase.

Para el problema de clases no balanceadas, la métrica F es una medida comúnmente usada (Luque, Carrasco, Martín, y de las Heras, 2019). Esta métrica es una combinación entre la precisión y el recuerdo que depende del parámetro $\beta$, mismo que varía entre cero e infinito y es usado para controlar la influencia la precisión y el Recuerdo de forma separada. Cuando $\beta$ es igual a cero, la métrica F no es otra cosa que la precisión y si $\beta \to \infty$ se aproxima a ser el recuerdo.

Un caso especial de la métrica F se presenta cuando $\beta = 1$, se le conoce como $F_1$,

$$F_1 = \frac{2 \cdot \text{Precisión} \cdot \text{Recuerdo}}{\text{Precisión} + \text{Recuerdo}}$$

Esta forma resulta útil porque es una media armónica entre dos números y ésta tiende a estar más cerca del más pequeño de los dos. Por tanto, un valor alto de $F_1$

asegura que tanto la precisión como el Recuerdo sean razonablemente altos.

## 2.6. Métodos vistos y su uso en la detección de agresividad

En este capítulo se han abordado diversos métodos que serán empleados en el Capítulo 6 y el Capítulo 7 para la detección de agresividad. La SVM será empleada para realizar un modelo de referencia con n-gramas a nivel de palabras y para pronosticar características de perfilado de autor como el género, la localización y la ocupación. La GRU con y sin un modelo de atención se usará para evaluar y contrastar con el resultados de la SVM de referencia. Posteriormente, a la SVM y a los modelos generados con la GRU se les adicionarán las características pronosticadas de los usuarios para así evaluar si esta estrategia permite tener un mejor desempeño mediante la evaluación del $F_1$.

# Capítulo 3

# Trabajo relacionado

La detección de agresividad en el lenguaje español es una tarea relativamente poco estudiada en comparación con el inglés. Por esta razón, resulta conveniente buscar en la literatura los avances alcanzados en otros idiomas, como el inglés, donde la tarea ha sido estudiada de forma amplia (Badjatiya, Gupta, Gupta, y Varma, 2017; Burnap y Williams, 2014; Davidson, Warmsley, Macy, y Weber, 2017; De Gibert, Perez, García-Pablos, y Cuadros, 2018; Gambäck y Sikdar, 2017; "Hate speech Detection on Twitter: Feature Engineering vs Feature Selection", s.f.; Kumar, Ojha, Malmasi, y Zampieri, 2018; MacAvaney y cols., 2019; Waseem, 2016; Waseem y Hovy, 2016; Zhang, Robinson, y Tepper, 2018). En la sección 3.1 se enuncian los métodos empleados en el discurso del odio y en la sección 3.2 se describen los métodos más exitosos en la detección de agresividad dentro del MEX-A3T en las ediciones pasadas.

## 3.1. Métodos empleados en el discurso del odio

La detección del discurso del odio es comúnmente vista como una tarea de clasificación de documentos de acuerdo con Zhang y cols. (2018). Los métodos empleados para clasificar suelen caer en dos categorías:

1. La primera categoría consiste en construir características de forma manual para posteriormente ingresarlo a un algoritmo de aprendizaje automático (*machine learning algorithm*).

2. La segunda hace uso del aprendizaje profundo para aprender de forma automática múltiples capas de características abstractas a partir de los datos.

### 3.1.1. Métodos con construcción de características y aprendizaje automático

Schmidt y Wiegand (2017) proveen un resumen de los enfoques y los tipos de características empleadas como entradas en los algoritmos usados para detectar el discurso de odio. A continuación, un breve resumen de los mismos:

- Las *características simples* o tradicionales como la bolsa de palabras (BoW, por sus siglas en inglés), los n-gramas de palabras y caracteres han dado evidencia de ser altamente predictivos en la detección del discurso de odio (Burnap y Williams (2014), Davidson y cols. (2017), Nobata, Tetreault, Thomas, Mehdad, y Chang (2016) ). Otras características también empleadas incluyen el conteo de menciones de URL, hashtags, ciertos signos de puntuación, la longitud de palabras y documentos, el número de mayúsculas usadas, entre otras.

- La *generalización de palabras* (*Word generalisation*) se hace usando técnicas como *clusters* de palabras y el modelado de temas (*topic modelling*) mediante la Asignación Latente de Dirichlet -*Latent Dirichlet Allocation*, LDA por sus siglas en inglés- (Xiang, Fan, Wang, Hong, y Rose, 2012) o *Brown clustering* (Warner y Hirschberg, 2012). En este sector también están los vectores de palabras (*word embeddings*), que se usan posteriormente para a partir de ellos construir vec-

tores de características de los mensajes ("Hate speech Detection on Twitter: Feature Engineering vs Feature Selection" (s.f.), Zhang y Luo (2019)).

- *Análisis de sentimientos*. Generalmente, los mensajes de odio también implican un sentimiento negativo, esta idea puede ser explotada incluyendo el grado de polaridad como una característica o bien, realizando la clasificación en dos pasos, primero se detectan los textos con polaridad negativa y después se busca detectar si los textos filtrados contienen discurso de odio. Una forma más simple de incluir el análisis de sentimientos es agregar el número de palabras positivas, negativas y neutrales en un documento como características.

- Los *Recursos léxicos* se usan bajo el supuesto de que los mensajes de odio contienen algunas palabras negativas específicas y, por tanto, su presencia en el documento se emplea como característica. Las listas públicas de palabras relacionadas de forma general con el odio y los lexicones especializados suelen ser empleados para este fin. Schmidt y Wiegand (2017) dan como ejemplo de listas públicas a Noswearing, The Racial Slur DataBase y Hatebase, ésta última ofrece listas para distintos idiomas y nacionalidades.

- Las *características sintácticas* son empleadas como características así como algunas relaciones de dependencia. Un ejemplo es el trabajo de Xu, Jun, Zhu, y Bellmore (2012), donde exploran el uso de n-gramas con las etiquetas resultado de hacer *Part of Speech (PoS)*. Otro trabajo que usa características sintácticas es el de Davidson y cols. (2017), donde también emplean n-gramas, *POS* y análisis de sentimientos.

En términos de clasificación, los algoritmos más usados son las SVM, el clasificador *Naïve Bayes*, la regresión logística y el *Random Forest*.

### 3.1.2. Métodos con aprendizaje profundo

En cuanto a las entradas en los métodos que emplean aprendizaje profundo, éstas pueden adoptar diversas formas, incluidas las antes vistas o, también suele emplearse como entrada el texto sin modificación alguna, ya sea a nivel de palabras o de carácter. Sin embargo, los datos de entrada no se utilizan únicamente para la clasificación, sino que la estructura de múltiples capas aprende nuevas representaciones de características abstractas que se utilizan para el aprendizaje. Es por ello que los métodos basados en el aprendizaje profundo se suelen centrar en el diseño arquitectura de la red pues es la responsable de extraer características útiles para clasificar a partir de la representación de características de entrada.

Entre las arquitecturas más comunes para minería de texto están las redes neuronales convolucionales y las redes neuronales recurrentes. Dentro de éstas últimas, las dos más usadas son la *Long Short-Term Memory network* (LSTM) y la *Gated Recurrent Unit* (Badjatiya y cols., 2017; Del Vigna, Cimino, Dell'Orletta, Petrocchi, y Tesconi, 2017; Gao y Huang, 2017; "Hate speech Detection on Twitter: Feature Engineering vs Feature Selection", s.f.; Wang, 2018; Zhang y cols., 2018). A las CNN se les conoce como una red eficaz para extraer características mientras que las RNN suelen tener un buen desempeño al modelar problemas de aprendizaje con secuencias como lo es el texto.

## 3.2. Métodos empleados en el MEX-A3T

En la Tabla 3.1 se encuentran los cuatro trabajos más sobresalientes en la detección de agresividad dentro del MEX-A3T, tanto de 2018 como de 2019. A continuación, se dará un breve resumen de lo realizado por ellos.

Graff y cols. (2018) (INGEOTEC) ganaron la competencia en el 2018 con EvoMSA, una arquitectura de dos niveles que usa información de diferentes modelos y

**Tabla 3.1:** Trabajos previos en el MEX-A3T y propuesta

| Enfoque | UACh | INGEOTEC | PRHLT | mineriaUNAM | Propuesta |
|---|---|---|---|---|---|
| Preproceso | | | | | ✓ |
| Minúsculas | ✓ | | | ✓ | ✓ |
| Normalizar | ✓ | ✓ | ✓ | ✓ | |
| Representación | | | | | |
| N-gramas, caracteres | | ✓ | | ✓ | |
| N-gramas, palabras | | ✓ | ✓ | | ✓ |
| Palabras agresivas | | | ✓ | | |
| Taylor-made lexicons | | ✓ | | | |
| Word embeddings | ✓ | ✓ | ✓ | | ✓ |
| Jerárquicos (textos) | ✓ | | | | |
| LIWC (textos) | | | | ✓ | |
| Clasificación | | | | | |
| SVM | ✓ | | | ✓ | ✓ |
| EvoMSA | | ✓ | | | |
| Deep-learning | | | ✓ | | ✓ |
| Model selection / Ensembles | | | ✓ | | |

obtiene una predicción vía consenso, combinada con modelo de Bernoulli basado en un Lexicon (LexB) y un modelo de conteo de palabras afectivas (*UpDown*).

Casavantes y cols. (2019) (UACh) emplearon un Perceptrón multicapa con n-gramas de caracteres usados de forma ponderada mediante TF-IDF (del inglés *Term Frequency Inverse Document Frequency*). De igual forma, exploraron la inclusión de las predicciones de ocupación y localización del usuario para probar si las personas atacan de manera distinta dependiendo de sus rasgos sin que obtuvieran cambios notables en los resultados.

De la Pena Sarracén y Rosso (2019) (PRHLT) participaron proponiendo un método que combina diferentes estrategias de clasificación: una red neuronal convolucional a nivel de palabras cuyas salidas alimentan una red neuronal LSTM; un codificador de oraciones universal pre-entrenado para codificar oraciones en vec-

tores de palabras; y un Perceptron multicapa que obtiene la representación TF-IDF del tuit. Los mejores resultados se obtuvieron con el modelo más simple, el Perceptrón multicapa con la representación TF-IDF de los tuits.

Ortiz, Gómez-Adorno, Reyes-Magaña, Bel-Enguix, y Sierra (2019) (mineriaUNAM) también participaron en el 2019, abordando problema con características lingüísticas y varios tipos de n-gramas (palabras, caracteres, palabras funcionales, signos de puntuación, entre otros) como entrada para una SVM entrenada usando un marco combinatorio que optimiza los resultados del clasificador.

## 3.3. Discusión

En la edición 2019 del MEX-A3T, la propuesta de INGEOTEC ha resultado ser la mejor para detectar la agresividad. El enfoque de la Universidad de Chihuahua (UACh) logró el segundo mejor puesto y además, es considerado por los organizadores de la competencia como un enfoque más simple (Aragón y cols., 2019).

Por otro lado, De la Pena Sarracén y Rosso (2019) se ubican en el tercer lugar con una estrategia que, casi por completo, usa métodos de aprendizaje profundo y mostraron que, con su enfoque, el desempeño de los modelos se ve mermado por la falta de más datos y de las palabras mal escritas o raras. El siguiente lugar en la competencia está ocupado por Ortiz y cols. (2019), quienes participaron con una propuesta que es enteramente del ámbito de aprendizaje máquina y muestran que el usar un marco combinatorio les permitió mejorar sus resultados respecto del 2018.

En el presente trabajo se propone explorar el uso de características simples como los n-gramas en conjunto con una SVM, así como métodos del aprendizaje profundo para detectar mensajes agresivos en el corpus del MEX-A3T 2020. De igual forma, se retoma la idea de Casavantes y cols. (2019) de incluir caracterísi-

cas de los autores de los tuits como el género, la localización y la ocupación, pero explorando con mayor profundidad la forma de incluir estas características en los modelos desarrollados.

# Capítulo 4

# Propuesta

La propuesta de esta tesis consiste en generar clasificadores con enfoques de aprendizaje profundo y aprendizaje máquina para la detección de agresividad. Una vez desarrollados estos métodos, se les proporciona mayor información de quien escribe el tuit mediante características de perfilado de autor para así evaluar si contribuyen a obtener mejores resultados. Teniendo este objetivo, en el resto de este capítulo se explicarán los clasificadores que se emplearán así como la forma de obtener las características de perfilado de autor.

## 4.1. SVM con n-gramas

El método de aprendizaje máquina a emplear es una SVM lineal y es también un método que se empleará como referencia. Se proponen dos variantes en las entradas al algoritmo:

1. Unigramas a nivel de palabra.

2. N-gramas de una, dos y tres palabras.

En el procedimiento previo para obtener los n-gramas se propone llevar a minúsculas los tuits y quitar los signos de puntuación. Para ambos experimentos se usa

la transformación TF-IDF (Jurafsky y Martin, 2014) para así obtener la relevancia de cada n-grama en cada tuit. Para escoger a los mejores n-gramas, se usa la $\chi^2$. Es decir, se obtiene el valor de la $\chi^2$ para cada n-grama y se ordena con base en él para después seleccionar los $k$ mejores que son la entrada para la SVM.

## 4.2. Métodos de aprendizaje profundo

Esta sección tiene por objetivo describir las arquitecturas propuestas para detectar agresividad y se divide en dos, la subsección 4.2.1 muestra las especificaciones de las CNN y la subsección 4.2.2 las especificaciones propias de las RNN.

Para cada red se preserva el contenido de los tuits y se mantiene el formato de mayúsculas y minúsculas que tienen originalmente. Para el proceso de separar un texto en piezas o *tokens*, se eliminan los signos de puntuación, convirtiendo así cada tuit en una secuencia de palabras separadas por espacios. El conjunto de todos los distintos *tokens* forman un vocabulario y cada palabra dentro del mismo se representa con un vector previamente entrenado. En este trabajo se emplean los vectores de palabras de tamaño 300 entrenados con el corpus *Spanish Billion Word Corpus* (Cardellino, 2019) mediante *FastText*, un método para producir vectores de palabras que puede verse como una extención de *word2vec* en el que el vector que representa a una palabra es la suma de las representaciones de los n-gramas a nivel de caracter de esa palabra. En todas las arquitecturas se comienza con una secuencia de tamaño 86 -este valor corresponde al segundo cuartil de la longitud de los tuits- los textos más largos se truncan y los más cortos se rellenan con ceros.

### 4.2.1. Redes convolucionales

Las CNN son el primer método propuesto con aprendizaje profundo, se basa en el trabajo de Kim (2014) visto en la sección 2.3.2 y con este tipo de arquitectu-

ra se pretende extraer de forma automática las características más relevantes. Los *tokens* de los tuits son representados mediante los vectores de palabras descritos anteriormente y se consideran tres variantes en los pesos de los vectores:

- Pesos no estáticos. Aquí se permite que los pesos se modifiquen para afinar el ajuste y aprender detalles propios de la tarea de detección de agresividad.

- Pesos estáticos. Con esta variación se hace el supuesto de que los pesos son útiles en su forma original para detectar agresividad.

- Pesos Aleatorios. Esta variación tiene el propósito de tener un modelo que sirva de referencia.

En la Figura 4.1 se muestra la forma general de las CNN a utilizar. Las variaciones se aplican en la capa *Embedding* que es donde se hace el mapeo entre las palabras y sus respectivos vectores, su salida es tamaño $86 \times 300$. De esta capa se desprenden tres brazos, cada uno realiza una convolución con 100 filtros, el primero de tamaño 3, el segundo de tamaño 4 y el tercero de tamaño 5. Posteriormente, se disminuye la dimensión realizando la operación *Global Max Pooling* en la siguiente capa, por lo que su salida es de tamaño $1 \times 100$; en la siguiente capa se concatenan los tres vectores, de tal manera que ahora su dimensión es de $1 \times 300$. En la siguiente capa se hace regularización mediante el *dropout* a una tasa de 0.5 y al final se pasa a una capa densa con una función de activación sigmoide.

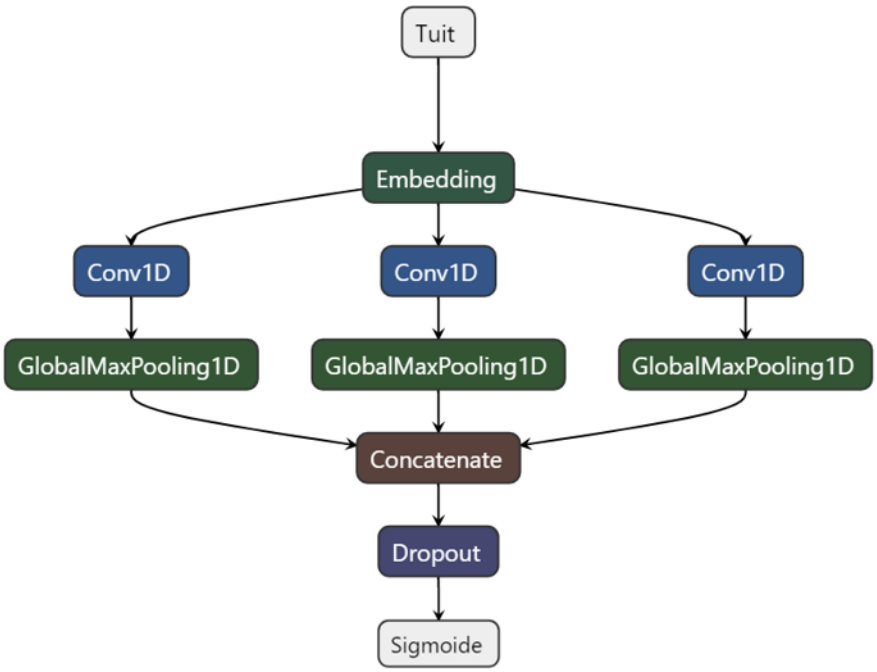



**Figura 4.1:** Forma general de la arquitectura de la CNN a emplear para detectar agresividad.

Se espera que el desempeño de las CNN sea al menos tan bueno como el de la SVM. Intuitivamente, con este tipo de red se obtienen de forma automática las entradas que se le dieron de forma "manual" a la SVM. Por ejemplo, los filtros de convolución capturan patrones locales de palabras.

### 4.2.2. Redes recurrentes

El segundo método propuesto con aprendizaje profundo se conforma por tres arquitecturas con la GRU como red recurrente. Este tipo de arquitectura, al tener sólo dos compuertas permite tener una estructura simple y con menos parámetros que otras, como la LSTM; en teoría, estas particularidades permiten entrenar de forma más rápida y generalizar mejor conjuntos de datos relativamente pequeños (Zhang y Luo, 2019).

En dos de las tres arquitecturas se propone usar el modelo de atención pro-

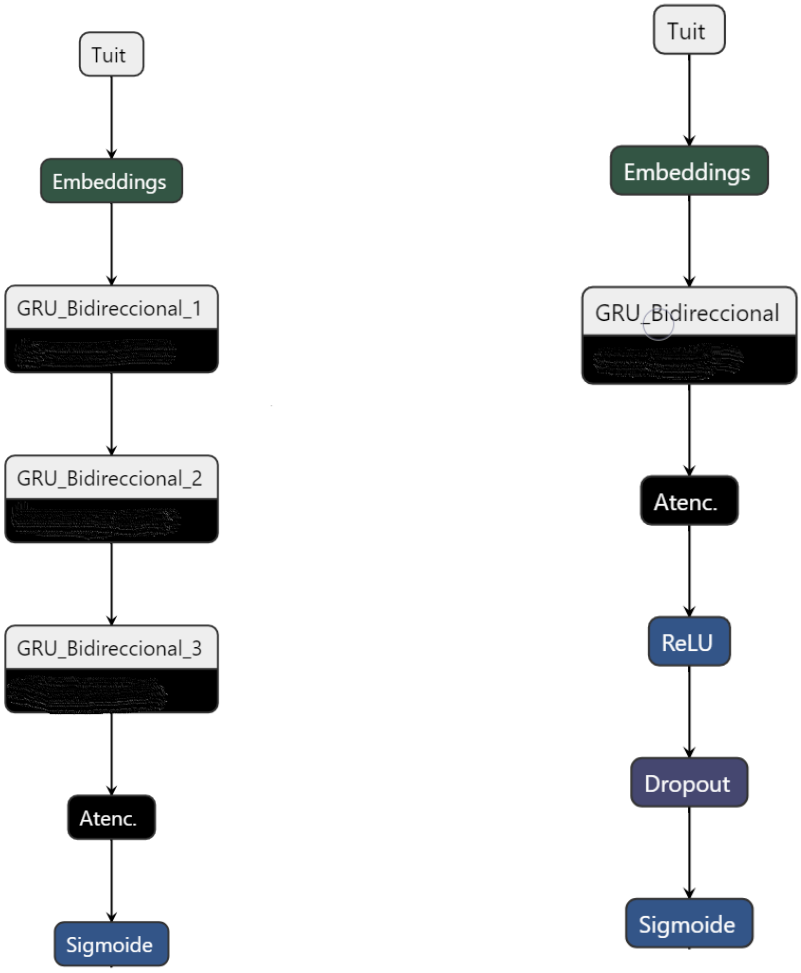
puesto por Raffel y Ellis (2015) esperando que se pueda asignar un peso mayor a las palabras más relevantes para la detección de agresividad. En la Figura 4.2 se muestra su estructura, al igual que las CNN, ambas comienzan con una secuencia de tamaño 86 -valor correspondiente al segundo cuartil de la longitud de los tuits- que entra a una capa encargada de mapear esta secuencia con los vectores de palabras (*embedding layer*) y tiene una salida de dimensión $86 \times 300$, -los vectores de palabras tienen tamaño 300-. A continuación, se da una descripción de las capas siguientes para cada arquitectura:

(a) En la arquitectura de la izquierda, la extracción de las características más relevantes se hace con tres capas con GRU Bidireccionales apiladas que tienen 128, 100 y 64 unidades respectivamente. La salida de la última capa GRU es de dimensión $1 \times 128$ y entra a una capa con atención para salir con la misma dimensión y finalmente pasar a una capa densa con función de activación sigmoide para obtener la predicción.

(b) La arquitectura de la derecha, en la siguiente capa tiene una GRU Bidireccional con 64 unidades por lo que su salida es de tamaño $1 \times 128$. Ésta, a su vez, es la entrada de una capa con atención con una salida de igual dimensión; luego, viene una capa densa con función de activación ReLU con 16 unidades seguido de *dropout* a una tasa de 0.10 y finalmente, se ingresa a una capa con una función de activación sigmoide para obtener la predicción.

La arquitectura de la tercera red recurrente a considerar se aprecia en la Figura 4.3, al no tener atención sirve para contrastar con el desempeño de las arquitecturas que sí la tienen. Así como las dos arquitecturas previas, comienza con una secuencia de tamaño 86 y una capa que mapea los índices de la secuencia con los vectores de palabras. La salida de esta capa tiene dimensión $86 \times 300$ y alimenta a una capa con una GRU Bidireccional con 64 unidades, su salida es de tamaño

$1 \times 128$. La siguiente capa tiene 16 unidades con una función de activación ReLU y su salida es de dimensión $1 \times 16$; le sigue otra capa con *dropout* a una tasa de 0.1 y por último una capa con una función sigmoide encargada de dar la predicción.




**(a)** Tres GRU Bidireccionales con atención.

**(b)** GRU Bidireccional y una capa con el modelo atención.

**Figura 4.2:** Diagramas de flujo de las arquitecturas propuestas con modelos de atención.

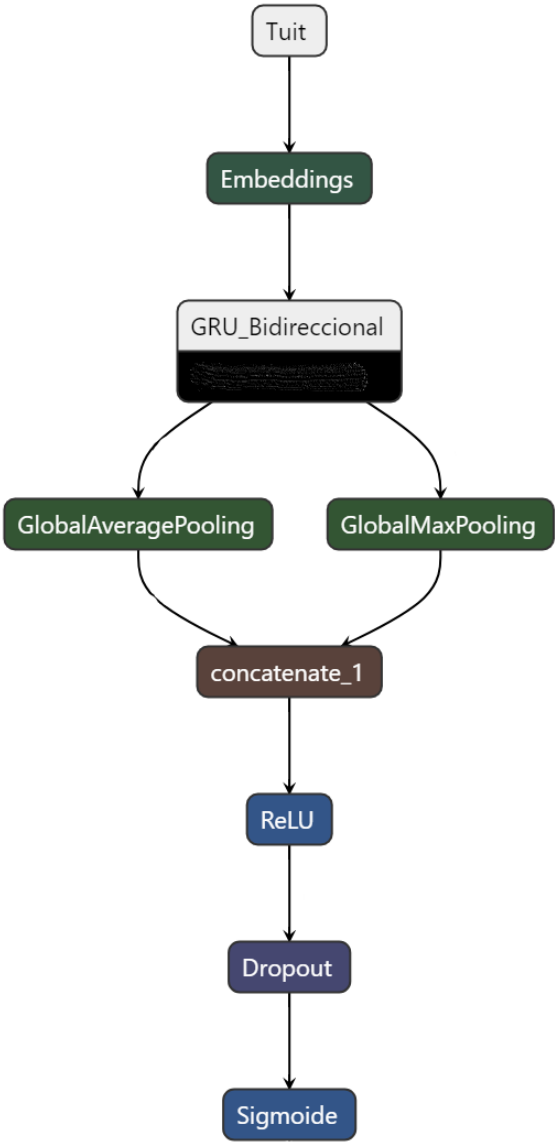



**Figura 4.3:** Diagrama de flujo de la arquitectura propuesta con una GRU Bidireccional sin un modelo de atención.

Cabe señalar que las arquitecturas mostradas en las figuras 4.2a, 4.2b y 4.3 son arquitecturas que funcionaron para otras tareas y se usaron para la detección de agresividad. Se hicieron ajustes manuales y los resultados fueron peores por lo que se determinó dejar las unidades iniciales, es decir, las unidades arriba descritas.

## 4.3. Características de perfilado de autor

Una vez que se han construido los modelos descritos en secciones anteriores surge la pregunta, ¿puede mejorarse su desempeño incorporando información extra?, ¿qué tipo de información? Russell y Miller (1977) mostraron que conocer la edad, el género, el idioma nativo y los dialectos de los usuarios de las redes sociales puede ayudar a identificar a potenciales terroristas. Por tanto, agregar datos socio-demográficos parece ser una idea que vale la pena explorar para detectar la agresividad en tuits.

Como se revisó en Capítulo 3 esta idea fue abordada en el MEX-A3T por Casavantes y cols. (2019), los autores emplearon dos variables producto de la tarea de perfilado de autor: la localización y la ocupación. Sin embargo, al incluir estas variables sus resultados mostraron cambios casi imperceptibles o ligeramente inferiores a sus modelos de referencia. En este trabajo de tesis se retoma la idea de agregar características de perfilado autor usando además de la ocupación y la localización, el género y, en lugar de usar todas las clases de una característica, se seleccionan sólo aquellas que posiblemente tengan una mayor utilidad, con la intuición de que algunas podrían meter menos ruido y perjudicar al modelo. De esta manera, podría suceder que, saber si quien escribe el tuit es estudiante ayude a detectar un mensaje agresivo; en cambio, si tiene un puesto administrativo pudiera ser irrelevante, más adelante, en la sección de resultados se verá que en efecto, esto sucede. A continuación, se describe el procedimiento para predecir características del autor así como la selección de las clases relevantes.

Para pronosticar el género, el lugar de residencia y ocupación se emplea el corpus *Author Profiling track of the MEX-A3T, 2019* (Aragón y cols., 2019) y se usa un modelo distinto para cada etiqueta.

Para cada una de las etiquetas, se considera el enfoque el basado en n-gramas propuesto por Aragón y López-Monroy (2018) con una pequeña variación en el

tamaño de los n-gramas. Este enfoque implica cuatro pasos: el primero extrae grupos de n-gramas de tamaño uno a tres a nivel de palabra y de tamaño tres a cinco a nivel de carácter. En el segundo paso, para cada grupo, los mejores n-gramas se seleccionan utilizando el criterio de la $\chi^2$ (Schütze, Manning, y Raghavan, 2008). Todos ellos se concatenan en el tercer paso y se usan para clasificar con una SVM en el cuarto paso. Una vez que se realiza la predicción, se aplica una codificación *one hot* para cada categoría de cada etiqueta, las variables resultantes se filtran aún más con el criterio de la $\chi^2$ para así seleccionar las mejores.

## 4.4. Inclusión de las características de perfilado de autor.

Una vez que se han seleccionado las características de perfilado de autor, el siguiente paso es incluirlas a la SVM y los modelos de aprendizaje profundo. Describir cómo se realiza esta inclusión es el propósito de esta sección.

Para la SVM, la fusión se hace concatenando las características del perfil con la matriz TDF-IDF que el algoritmo toma como entrada. Para los modelos de aprendizaje profundo, se abre otro "canal" por donde ingresan las características de los usuarios mediante una capa densa con la identidad como una función de activación, su salida se adiciona a la arquitectura en un nivel distinto:

- En los modelos CNN -con pesos estáticos, aleatorios y no estáticos- así como en los modelos con GRU Bidireccional y atención, las características se concatenan con las extraídas por la red antes de realizar las predicciones, es decir, de la capa con la función de activación sigmoide, la Figura 4.4 en los incisos (a) y (b) muestra los modelos con GRU Bidireccionales y atención, mientras que el inciso (d) muestra la forma general para la CNN.

- En el modelo con una GRU Bidireccional y sin atención, las características de perfilado de autor se adicionan a la arquitectura en la capa que concatena las salidas del *Global Average Pooling* y *Global Max Pooling*. En el inciso (c) de la Figura 4.4 se puede ver lo anteriormente descrito de forma gráfica.

La razón de tener incluir las características en distintos niveles de la arquitectura es sólo para probar su desempeño.

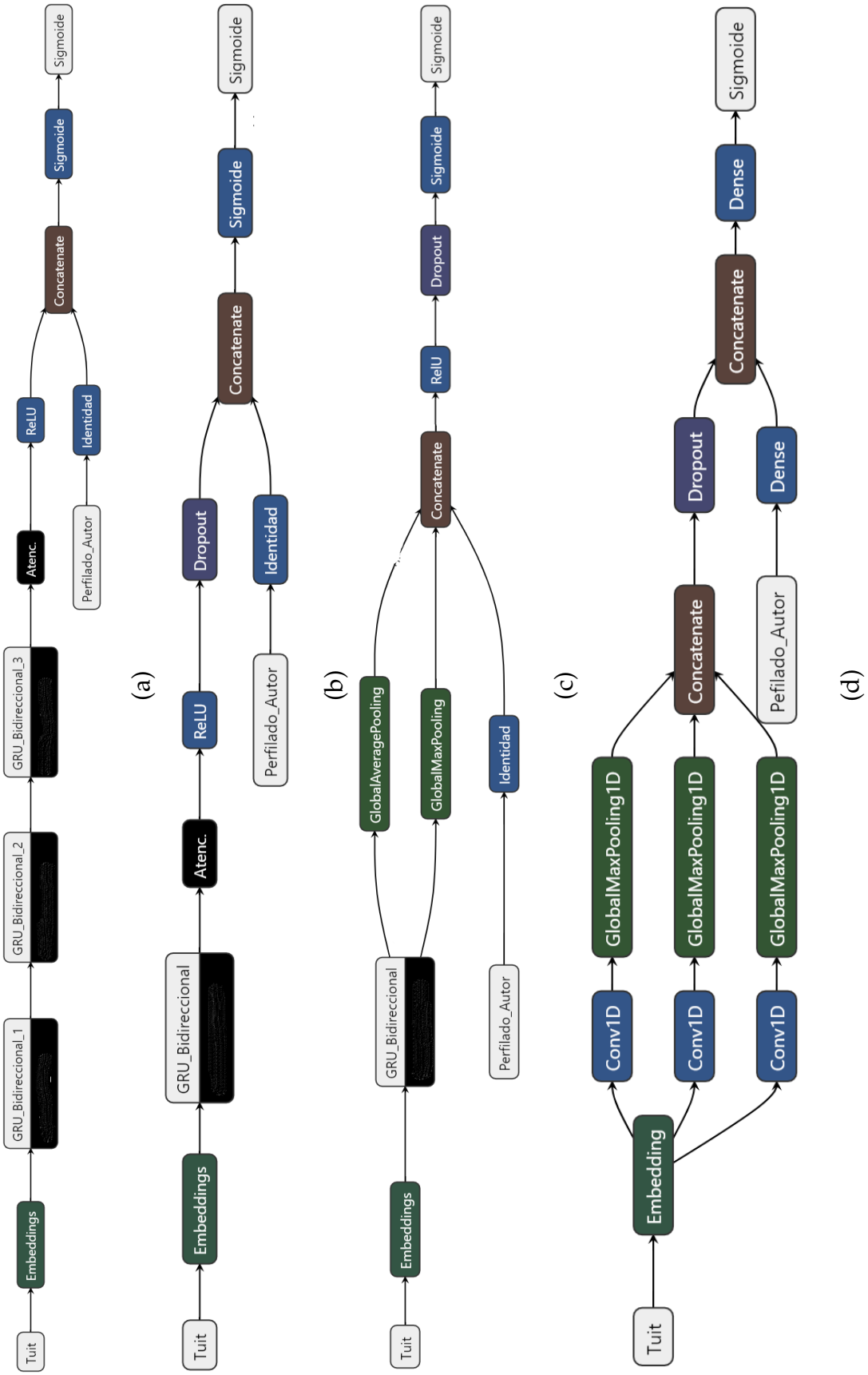



**Figura 4.4:** Arquitecturas con métodos de aprendizaje profundo que incluyen las características de perfilado de autor. Los incisos (a) y (b) son las arquitecturas con GRU bidireccionales y con atención, la (c) involucra GRU sin atención y el inciso (d) la arquitectura CNN.

# Capítulo 5

# Conjunto de datos y ajustes para los experimentos

En este capítulo tiene la finalidad de mostrar el corpus a emplear y la configuración de los experimentos. En la Sección 5.1 se da una breve descripción del corpus MEX-A3T para la detección de agresividad y en la sección Sección 5.2 se da la configuración métodos propuestos en las secciones 4.1 y 4.2.

## 5.1. El corpus del MEX-A3T

Para la edición 2020 del MEX-A3T los organizadores suministraron un conjunto de datos distinto del empleado en ediciones pasadas si bien conservaron una proporción similar de tuits agresivos. En esta sección se describirá el corpus y la forma en como fue construido con base en la información disponible en el sitio web de la competencia[1].

[1] https://sites.google.com/view/mex-a3t

### 5.1.1. Construcción del corpus del MEX-A3T

Para la construcción del corpus, se recolectaron tuits durante tres meses teniendo como centro a la Ciudad de México. Emplearon palabras groseras y *hashtags* controvertidos para acotar la búsqueda. Para seleccionar el conjunto de términos que sirvieron como semillas para extraer los tuits se emplearon palabras clasificadas como vulgares y no coloquiales en el Diccionario de Mexicanismos de la Academia Mexicana de la Lengua, así como palabras y *hashtags* identificados por el Instituto Nacional de las Mujeres como relacionados con la violencia y el acoso sexual contra las mujeres en Twitter.

### 5.1.2. Descripción del corpus

La Tabla 5.1 muestra la distribución del corpus, el 71.2 % del corpus pertenece a la clase no agresiva mientras que el 28.8 % conforma la parte agresiva.

| Clase | No. Tuits | % |
|---|---|---|
| No agresiva | 5,222 | 71.2 % |
| Agresiva | 2,110 | 28, 8 % |
| Total | 7,332 | 100 % |

**Tabla 5.1:** Distribución del corpus MEX-A3T.

Para la realización de experimentos, se dividió el corpus en tres partes de forma aleatoria, el 70 % fue usado para entrenar, el 10 % para validar y el 20 % para probar los resultados. La partición se realizó con la librería scikit-learn usando una semilla igual a 1.

### 5.1.3. Preproceso del corpus

En cuanto al preproceso, es importante señalar que para los experimentos con la SVM, los tuits se llevaron a minúsculas, mientras que para los experimentos con redes neuronales los tuits se mantuvieron en su forma original y en ambos se removieron los signos de puntuación.

## 5.2. Configuraciones de los experimentos

En esta sección se listarán los detalles necesarios para reproducir los resultados obtenidos en este trabajo. Para comenzar a realizar los experimentos el corpus de divide de forma aleatoria en tres: entrenamiento, validación y prueba. El ajuste de parámetros se realiza con el conjunto de entrenamiento y validación, una vez escogidos, estos dos conjuntos se consideran como uno solo, se vuelve a entrenar con los parámetros seleccionados y se reportan los resultados obtenidos con el conjunto de prueba. La métrica a usar para medir el desempeño de los diferentes experimentos será el $F_1$ sobre la clase agresiva dado que es la métrica usada por los organizadores del MEX-A3T para determinar el mejor método.

### SVM

En la Sección 4.1 se vieron dos variantes para las entradas al algoritmo. Para la primera, es decir, aquella que sólo considera los unigramas se emplean 5 mil mejores palabras; para la segunda, se emplean los mejores mil trigramas, 2 mil bigramas y 5 mil unigramas.

Para entrenar la SVM lineal se usó la librería scikit-learn, con pesos balanceados para las clases y los demás parámetros se toman por defecto. Para la selección del parámetro de penalización, C, se busca con una malla (*grid search*) considerando como posibles valores el conjunto {0.05, 0.12, 0.25, 0.5, 1.00, 2.00, 4.00}. Además, se

usan 7 pliegues de validación cruzada y se toma el $F_1$ como métrica para evaluar el desempeño, esta misma configuración se toma para ajustar las SVM empleadas para predecir los atributos de perfilado de autor.

## CNN y RNN

En todos los experimentos con aprendizaje profundo se empleó la entropía cruzada binaria (*binary crossentropy*) como función de pérdida, optimizador *adam* con una tasa de aprendizaje igual a 0.001. Al momento de entrenar se emplea la detención temprana (*early stopping*) como forma de regularización, la métrica a dar seguimiento es el la función de pérdida del conjunto de validación para las CNN y el $F_1$ también del conjunto de validación para las RNN.

Es importante mencionar que en los experimentos de aprendizaje profundo, al momento de realizar el entrenamiento los modelos se ejecutan una vez y con el conjunto de validación de esa ejecución se determina el número de épocas, el tamaño del lote (*batch size*) y el punto de corte en la probabilidad de ser agresivo o no; el número de unidades dentro de cada capa así como el número de capas y la determinación de la función de activación se hacen con base en descubrimientos empíricos y su comportamiento en el conjunto de validación. Una vez definidos los mejores parámetros, los modelos se vuelven a ajustar considerando ahora el conjunto de entrenamiento y validación como uno solo y el desempeño se evalúa con el conjunto de prueba.

# Capítulo 6

# Experimentos y Resultados

En este capítulo tiene el objetivo de mostrar los resultados de los métodos propuestos en las secciones 4.1 y 4.2. Se divide en tres partes: en la Sección 6.1 se muestra el $F_1$ obtenido en los experimentos sin incluir variables de perfilado de autor. La Sección 6.2 se divide a su vez en dos, la primera parte describe el corpus empleado para perfilado de autor, muestra los resultados de los modelos propuestos para la predicción de características del perfil de los usuarios y enseguida se da un resumen de los pronósticos hechos para los usuarios del corpus MEX-A3T; posteriormente, se muestra el proceso y resultados de seleccionar las mejores características. En la segunda parte, se dan los resultados de incluir las características de perfilado de autor en los experimentos. Finalmente, en la Sección 6.3 a manera de resumen, se comparan los mejores experimentos con y sin información adicional de los usuarios.

## 6.1. Experimentos sin perfilado de autor

En esta sección se muestran los resultados de los experimentos que servirán como marco de referencia para en secciones posteriores contrastar con estos mismos experimentos, pero incluyendo las características perfilado de autor.

La Tabla 6.1 tiene los experimentos realizados con la SVM, la CNN, la RNN y su respectivo $F_1$ en el conjunto de prueba[1]. En esta primera fase, los mejores resultados se obtienen con la CNN de pesos no estáticos con un valor de $F_1$ de 71.680 en en el conjunto de prueba.

| | Variante | $F_1$ Prueba |
|---|---|---|
| SVM | Unigramas | 70.975 |
| | $\{1, 2, 3\}$ gramas | 71.296 |
| CNN | Estático | 69.850 |
| | Aleatorio | 70.950 |
| | No estático | ***71.680*** |
| RNN | 1 GRU Bid. + Attention | 69.900 |
| | 1 GRU Bid. + AvgPool + MaxPool | 71.000 |
| | 3 GRU Bid. + Attention | 68.960 |

**Tabla 6.1:** $F_1$ de la fase de prueba de los experimentos con SVM, CNN y RNN. De estos tres bloques, el modelo no estático con CNN es el que muestra los mejores resultados en el conjunto de prueba.

Por otro lado, también pueden resaltarse otros dos aspectos: 1) el incluir bigramas y trigramas en la SVM tiene un efecto positivo. De hecho, en esta primera serie de experimentos la SVM con $\{1, 2, 3\}$ gramas tiene un desempeño mejor que todos los otros métodos propuestos, con excepción de la CNN con pesos no estáticos. 2) Dentro de los experimentos con RNN, el modelo con una GRU sin atención resulta tener una mejora apenas perceptible respecto del modelo con una GRU y con atención. Por tanto, no podría decirse que hasta ahora el modelo de atención tenga efectos en el desempeño, de igual forma, esto demuestra que la SVM aunque tradicional, es un modelo fuerte y difícil de derrotar incluso con Deep Learning.

[1]Dado que en algunos casos la diferencia en el $F_1$ es en centésimas o milésimas, el $F_1$ se presenta multiplicado por 100.

## 6.2. Experimentos con perfilado de autor

La segunda parte de la experimentación consiste en agregar mayor información sobre los usuarios con la finalidad de incrementar el desempeño de los modelos y se compone de dos fases: 1) crear clasificadores que permitan predecir el género, la región y la ocupación del corpus del MEX-A3T, y 2) agregar esta información a los modelos que ya se tienen para posteriormente probar si existe una mejora en su desempeño, en otras palabras, se retroalimenta al modelo de agresividad con el de perfilado. A continuación, se muestran los resultados de la primera fase en la Subsección 6.2.1 y en la Subsección 6.2.2 los resultados de la segunda.

### 6.2.1. Predicción de características

Para predecir los atributos de perfilado de autor de los usuarios que escriben los tuits, se seleccionó un corpus, luego se ajustaron clasificadores para cada uno de los rasgos y, posteriormente, se usaron los clasificadores para predecir el perfil de los usuarios del MEX-A3T. Finalmente, se seleccionaron sólo aquellos atributos que resultaron ser más relevantes. En el resto de esta sección, se explicará con detalle los resultados obtenidos.

#### Corpus para perfilado de autor

El corpus empleado para crear clasificadores que sirvan para pronosticar características del autor fue el *Author Profiling track of the MEX-A3T, 2019* (Aragón y cols., 2019). Consta de tuits de 3,500 usuarios mexicanos y busca predecir el género, la ocupación y la región del país donde se localiza el usuario. A continuación, se muestran las clases incluidas en cada característica:

- Género: femenino y masculino.
- Región: centro, norte, noreste, noroeste, sureste y oeste.

- Ocupación: artes, estudiante, social, ciencias, deportes, administrativo, salud y otros.

La Tabla 6.2 muestra la distribución del corpus para cada una de las características arriba mencionadas. Género se caracteriza por ser la única con clases balanceadas; región, tiene sus clases no balanceadas y un porcentaje alto de los usuarios -el 62.28 %- se concentra en las regiones centro y noreste. La ocupación tampoco es una etiqueta balanceada: el 63.36 % de los usuarios son estudiantes o trabajan en el área de las ciencias sociales.

| Característica | Clase | No. Tuits | % en clase |
|---|---|---|---|
| Género | Masculino | 1,750 | 50 % |
| | Femenino | 1,750 | 50 % |
| Región | Norte | 106 | 3.02 % |
| | Noroeste | 573 | 16.45 % |
| | Noreste | 914 | 26.11 % |
| | Centro | 1,266 | 36.17 % |
| | Oeste | 322 | 9.20 % |
| | Sureste | 316 | 9.02 % |
| Ocupación | Artes | 240 | 6.85 % |
| | Estudiante | 1,648 | 47.08 % |
| | Sociales | 570 | 16.28 % |
| | Ciencias | 185 | 5.28 % |
| | Deportes | 45 | 1.28 % |
| | Administrativo | 632 | 18.05 % |
| | Salud | 105 | 3.00 % |
| | Otros | 75 | 2.14 % |

**Tabla 6.2:** Distribución del corpus *Author Profiling track of the MEX-A3T, 2019* para las características de género, región y localización.

### Clasificadores para la predicción de características

En el corpus *Author Profiling track of the MEX-A3T 2019* se concatenaron todos los tuits de un mismo usuario y se consideraron como uno solo. Luego, se dividió de forma aleatoria en tres partes, el 70 % fue usado para entrenar, el 10 % para validar y el 20 % para probar los resultados. Como preproceso, todos los tuits se llevaron a minúsculas y se removieron los signos de puntuación.

Para cada una de las etiquetas -género, región y ocupación-, se ajustó una SVM siguiendo la metodología descrita en la Sección 4.3 y el $F_1$ en el conjunto de prueba para las distintas etiquetas se muestra en la Tabla 6.3. El género es la característica que tuvo mejores resultados en sus clases, tiene un 91.827 para el género femenino y un 92.307 para el masculino. En la región, todas sus clases con excepción de la clase oeste, tuvieron un $F_1$ igual o superior a 88.695. Por otro lado, en la ocupación es donde el $F_1$ es más variable, posiblemente debido a que las clases no están balanceadas; las clases con mejores resultados fueron estudiante con un $F_1$ igual a 90.909, ciencias sociales con 65.079 y deportes con 63.157.

### Pronósticos de las características de perfilado de autor

Las SVM ajustadas para el género, la ocupación y la región, se usan para pronosticar estas características para los usuarios del corpus datos MEX-A3T. La Figura 6.1, Figura 6.2 y Figura 6.3 muestran el número de usuarios pronosticados en cada clase de cada característica.

Dentro de los pronósticos destaca que el 71.72 % del total del corpus MEX-A3T se predice que es del sexo masculino. Respecto a la ocupación, se pronostica que el 61.52 % trabaja en el área de las ciencias sociales o es estudiante; finalmente, el 73.22 % se localiza en la región centro u oeste del país.

| Característica | Clase | $F_1$ Prueba |
|---|---|---|
| Género | Femenino | 91.827 |
| | Masculino | 92.307 |
| Región | Norte | 91.666 |
| | Noroeste | 89.215 |
| | Noreste | 92.876 |
| | Centro | 90.485 |
| | Oeste | 74.015 |
| | Sureste | 88.695 |
| Ocupación | Artes | 40.476 |
| | Estudiante | 90.909 |
| | Sociales | 65.079 |
| | Ciencias | 31.250 |
| | Deportes | 63.157 |
| | Administrativo | 52.488 |
| | Salud | 18.181 |
| | Otros | 9.090 |

**Tabla 6.3:** $F_1$ de la fase de prueba de las SVM ajustadas para predecir el género, la ocupación y la localización.

### Selección de características

Con la finalidad de incluir en los experimentos sólo aquellas características que conlleven a una mejora en el $F_1$ de los experimentos mostrados en la Tabla 6.1, se tomó del conjunto de entrenamiento cada categoría pronosticada para la ocupación, la región y el género en su representación binaria, esto es, como un vector binario por cada clase de cada característica. De esta forma, se seleccionan sólo las clases que están más relacionadas con la variable objetivo.

Una vez que se obtuvieron las clases en su forma binaria se obtuvo el valor del estadístico $\chi^2$ al igual que para todos los n-gramas de una, dos y tres palabras y se

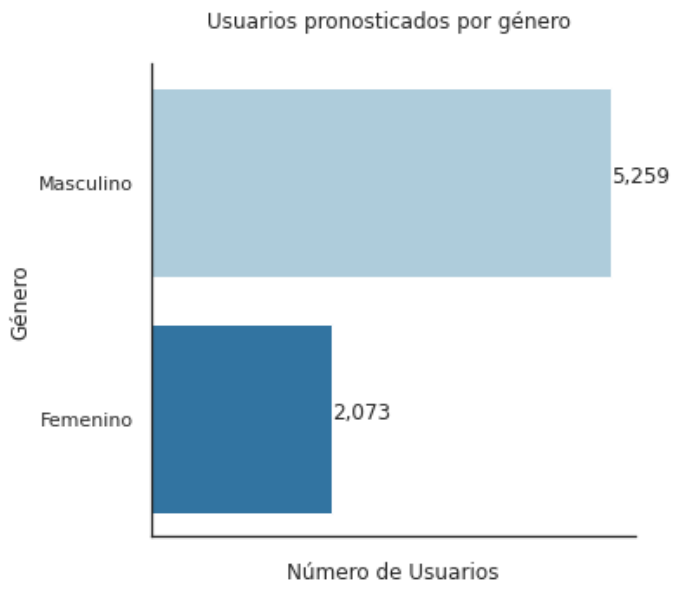



**Figura 6.1:** Pronóstico del género para la totalidad del conjunto de entrenamiento.

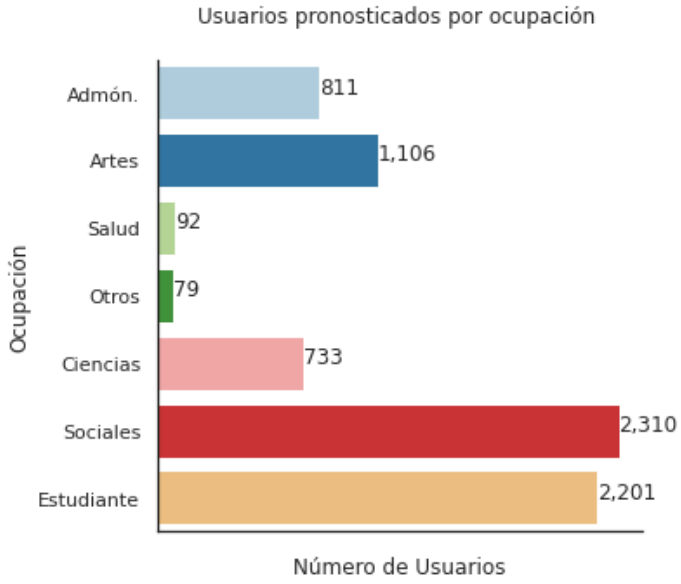



**Figura 6.2:** Pronóstico de la ocupación para la totalidad del conjunto de entrenamiento.

ordenó de forma descendente, la Figura 6.4 muestra los aquellos que obtuvieron un valor de $\chi^2$ superior a 20.

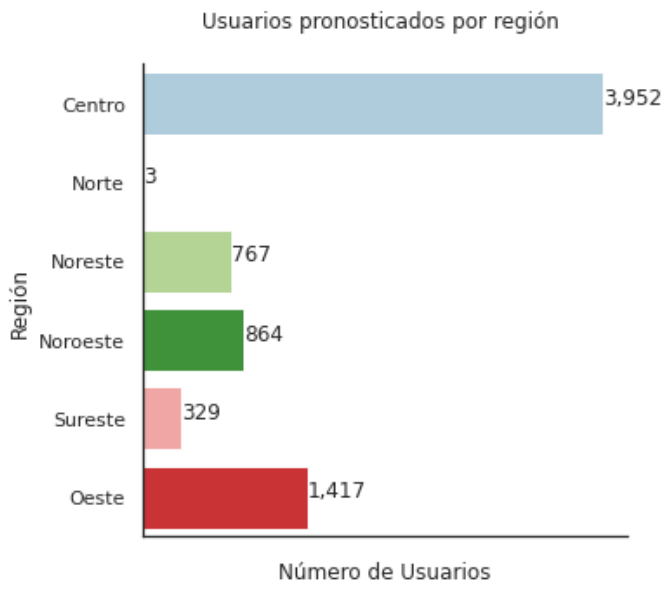



**Figura 6.3:** Pronóstico de la región donde se localiza cada uno de los usuarios del conjunto de entrenamiento.

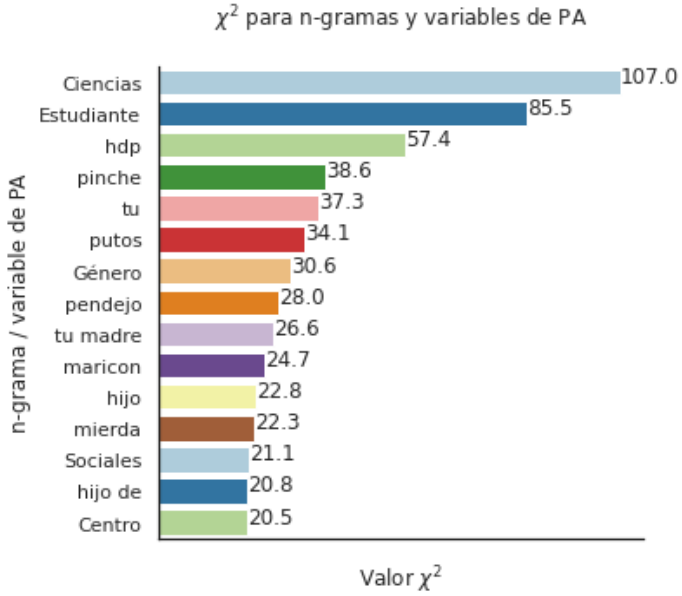



**Figura 6.4:** Valor de la $\chi^2$ para las variables *one-hot encoding* de perfilado de autor y para los n-gramas de una, dos y tres palabras. *Ciencias*, *Estudiante*, *Sociales*, *Género* y *Centro* se corresponden a categorías de perfilado de autor.

Las variables de perfilado de autor con una posición más alta son ciencias, estudiante, género, sociales y la región centro. Por tanto, son estas variables las que se incluirán en los experimentos para detectar tuits agresivos pues el resto de las

categorías parecen ser irrelevantes.

### 6.2.2. Inclusión de características de perfilado de autor

Una vez que se ha decidido qué atributos incluir a los experimentos, la siguiente parte es agregarlos. Se continúan usando los experimentos de la Tabla 6.1 y la forma en cómo se fueron adicionando fue uno seguido de otro, comenzando por el género, seguido de ciencias, sociales, estudiante y la región centro. En las Tablas 6.4 y 6.5 están los resultados de este procedimiento.

| | | Variante | $F_1$ Prueba |
|---|---|---|---|
| Género, Ciencias, Sociales, Estudiante | SVM | Unigramas | 71.366 |
| | | $\{1,2,3\}$ gramas | 70.915 |
| | CNN | Estático | 69.470 |
| | | Aleatorio | 71.410 |
| | | No estático | 72.100 |
| | RNN | 1 GRU Bid. + Attention | 68.820 |
| | | 1 GRU Bid. + AvgPool + MaxPool | 71.330 |
| | | 3 GRU Bid. + Attention | ***72.300*** |
| Género, Ciencias, Sociales, Centro, Estudiante | SVM | Unigramas | 71.523 |
| | | $\{1,2,3\}$ gramas | 71.214 |
| | CNN | Estático | 69.680 |
| | | Aleatorio | 71.530 |
| | | No estático | ***71.730*** |
| | RNN | 1 GRU Bid. + Attention | 70.550 |
| | | 1 GRU Bid. + AvgPool + MaxPool | 71.000 |
| | | 3 GRU Bid. + Attention | 69.240 |

**Tabla 6.5:** $F_1$ de la fase de prueba de los experimentos que incluyen las características de Género, si la ocupación está en el área de las Ciencias, las Ciencias Sociales, si es Estudiante y si pertenece a la región Centro.

| | | Variante | $F_1$ Prueba |
|---|---|---|---|
| Género | SVM | Unigramas | 70.989 |
| | | {1,2,3} gramas | 71.379 |
| | CNN | Estático | 69.840 |
| | | Aleatorio | ***71.680*** |
| | | No estático | 71.580 |
| | RNN | 1 GRU Bid. + Attention | 69.430 |
| | | 1 GRU Bid. + AvgPool + MaxPool | 71.430 |
| | | 3 GRU Bid. + Attention | 68.440 |
| Género, Ciencias | SVM | Unigramas | 70.925 |
| | | {1,2,3} gramas | 71.379 |
| | CNN | Estático | 70.330 |
| | | Aleatorio | 71.820 |
| | | No estático | 71.480 |
| | RNN | 1 GRU Bid. + Attention | 70.250 |
| | | 1 GRU Bid. + AvgPool + MaxPool | 70.970 |
| | | 3 GRU Bid. + Attention | ***72.060*** |
| Género, Ciencias, Sociales | SVM | Unigramas | 71.145 |
| | | {1,2,3} gramas | 71.296 |
| | CNN | Estático | 69.930 |
| | | Aleatorio | 71.510 |
| | | No estático | ***72.000*** |
| | RNN | 1 GRU Bid. + Attention | 68.340 |
| | | 1 GRU Bid. + AvgPool + MaxPool | 71.770 |
| | | 3 GRU Bid. + Attention | 68.960 |

**Tabla 6.4:** $F_1$ de la fase de prueba de los experimentos que incluyen las características de Género, si la ocupación está en el área de las Ciencias o las Ciencias Sociales.

Dentro de los resultados en el conjunto de prueba, destacan aquellos que obtienen un $F_1$ mayor o igual a 72, éstos corresponden a los experimentos donde se

agregan las siguientes combinaciones de atributos de perfilado de autor:

a) Género y ciencias.

b) Género, ciencias y sociales.

c) Género, ciencias, sociales y estudiante.

En las combinaciones a y c, la arquitectura con 3 GRU Bidireccionales y el modelo de atención es donde se logran mejores resultados, el $F_1$ es de 72.060 y 72.300, respectivamente. Por otro lado, la CNN con pesos no estáticos logra un $F_1$ de 72.000 con la combinación b y un 72.100 con la combinación c; es decir, alcanza un desempeño ligeramente inferior pero con una arquitectura más simple.

## 6.3. Comparación entre experimentos con y sin características de perfilado de autor

Una vez que se tienen los resultados de los experimentos, la pregunta que se desea responder es: ¿incrementa el desempeño si se incorporan los rasgos particulares de cada usuario? En la Tabla 6.6 se muestra el experimento más sobresalientes sin información adicional y los experimentos con perfilado de autor que obtuvieron un $F_1$ con los datos de prueba mayor o igual a 72.

Puede verse que, el mejor resultado sin adicionar información propia del usuario resulta inferior a cualquiera de los mejores experimentos que sí la incluyen.

| | Variante | $F_1$ Prueba |
|---|---|---|
| Sin perfilado de autor | CNN pesos no estáticos | 71.680 |
| Género, Ciencias | 3 GRU Bid. + Attention | 72.060 |
| Género, Ciencias, Sociales | CNN pesos no estáticos | 72.000 |
| Género, Ciencias, Sociales, Estudiante | CNN pesos no estáticos | 72.100 |
| | 3 GRU Bid. + Attention | ***72.300*** |

**Tabla 6.6:** $F_1$ de los mejores modelos con y sin características de perfilado de autor.

# Capítulo 7

# Conclusiones y trabajo futuro

En este documento se mostró el trabajo realizado para detectar agresividad en tuits escritos en español por mexicanos dentro del marco del MEX-A3T 2020 así como el resultado obtenido dentro de la competencia.

La propuesta de esta tesis consistió en generar un conjunto de métodos de aprendizaje máquina y aprendizaje profundo que sirvieron como base para posteriormente contrastar con ellos mismos, pero ahora incluyendo características del perfil de los usuarios. Como métodos de aprendizaje máquina con características tradicionales se usó una SVM con unigramas así como $\{1, 2, 3\}$-gramas; dentro de los métodos de aprendizaje profundo se usaron arquitecturas con CNN al igual que GRU con y sin atención a nivel de palabra. Las características de perfilado de autor que se exploró incluir fueron el género, la ocupación y la región del país a la que pertenece el usuario.

El mejor de los resultados encontrados tuvo un $F_1$ de 72.3 con 3 GRU Bidireccionales con atención y el género, ciencias, ciencias sociales y estudiante como características del autor. Este método tiene un mejor desempeño que la CNN con pesos no estáticos, el mejor experimento sin características de perfilado de autor encontrado en este trabajo de tesis que tuvo un $F_1$ de 71.68, por lo tanto, se puede

concluir que el uso de características de perfilado de autor sí contribuye a mejorar la detección de agresividad.

Ahora bien, la propuesta ganadora del MEX-A3T obtuvo un $F_1$ de 79.98 ( para más detalles ver Apéndice A) lo cual muestra que el trabajo realizado para esta tesis es mejorable. Dos estrategias que podrían explorarse para ello en un futuro son las siguientes:

I) Experimentos a nivel de carácter. En este trabajo todos los experimentos que se abordaron son a nivel de palabra y dado que los tuits son textos cortos, los métodos a nivel de caracter podrían ayudar a tener un mejor desempeño.

II) Optimización del número de unidades en las RNN. Dadas las restricciones de cómputo, en este trabajo el número de unidades dentro de cada capa en las RNN se ajustó "manualmente´´, por lo que puede existir una mejora al emplear métodos como la optimización bayesiana o búsqueda aleatoria (*random seach*).

Finalmente, es posible que la mejora en el desempeño aún incorporando estas dos estrategias esté acotada dado que el tamaño del corpus es relativamente pequeño por lo que usar transferencia de conocimiento o técnicas para incrementar el número de tuits podrían ayudar a incrementar el $F_1$.

# Referencias


Albon, C. (2018). *Machine learning with python cookbook: Practical solutions from preprocessing to deep learning*. O'Reilly Media, Inc.

Álvarez-Carmona, M. Á., Guzmán-Falcón, E., Montes-y Gómez, M., Escalante, H. J., Villasenor-Pineda, L., Reyes-Meza, V., y Rico-Sulayes, A. (2018). Overview of MEX-A3T at IberEval 2018: Authorship and Aggressiveness Analysis in Mexican Spanish Tweets. En *Notebook Papers of 3rd SEPLN Workshop on Evaluation of Human Language Technologies for Iberian Languages (IBEREVAL), seville, spain* (Vol. 6).

ALW3. (2020). *3rd Workshop on Abusive Language Online.* Consultado el 23-03-2020 en https://sites.google.com/view/alw3/.

Aragón, M. E., Álvarez-Carmona, M. Á., Montes-y Gómez, M., Escalante, H. J., Villasenor-Pineda, L., y Moctezuma, D. (2019). Overview of MEX-A3T at IberLEF 2019: Authorship and Aggressiveness Analysis in Mexican Spanish Tweets. En *Notebook Papers of 1st SEPLN Workshop on Iberian Languages Evaluation Forum (IberLEF), Bilbao, Spain.*

Aragón, M. E., y López-Monroy, A. P. (2018). Author Profiling and Aggressiveness Detection in Spanish Tweets: MEX-A3T 2018. En *IberEval@ SEPLN* (pp. 134–139).

Badjatiya, P., Gupta, S., Gupta, M., y Varma, V. (2017). Deep Learning for Hate Speech Detection in Tweets. En *Proceedings of the 26th International Conference*

*on World Wide Web Companion* (pp. 759–760).

Bahdanau, D., Cho, K., y Bengio, Y. (2014). Neural Machine Translation by jointly Learning to Align and Translate. *arXiv preprint arXiv:1409.0473*.

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rosso, P., y Rangel Pardo, F. M. (2019). SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. En *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 54–63).

Burnap, P., y Williams, M. L. (2014). Hate Speech, Machine Classification and Statistical Modelling of Information Flows on Twitter: Interpretation and Communication for Policy Decision Making. *Internet, Policy and Politics Conference, Oxford, UnitedKingdom*.

Cardellino, C. (2019, August). *Spanish Billion Words Corpus and Embeddings.* Descargado de https://crscardellino.github.io/SBWCE/

Casavantes, M., López, R., y González, L. C. (2019). UACH at MEX-A3T 2019: Preliminary Results on Detecting Aggressive Tweets by Adding Author Information via an Unsupervised Strategy. En *Proceedings of the First Workshop for Iberian Languages Evaluation Forum (IberLEF 2019), CEUR WS Proceedings.*

Cho, K., Van Merriënboer, B., Bahdanau, D., y Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder-decoder Approaches. *arXiv preprint arXiv:1409.1259*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., y Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv preprint arXiv:1406.1078*.

Chung, J., Gulcehre, C., Cho, K., y Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv preprint arXiv:1412.3555*.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., y Kuksa, P. (2011). Natural Language Processing (almost) from Scratch. *Journal of Machine Learning Research*, *12*(Aug), 2493–2537.

Cortes, C., y Vapnik, V. (1995). Support-vector Networks. *Machine learning*, *20*(3), 273–297.

Cuza, C. E. M., De la Peña Sarracén, G. L., y Rosso, P. (2018). Attention Mechanism for Aggressive Detection. En *CEUR Workshop Proc.* (Vol. 2150, pp. 114–118).

Davidson, T., Warmsley, D., Macy, M., y Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. En *Eleventh International AAAI Conference on Web and Social Media.*

De Gibert, O., Perez, N., García-Pablos, A., y Cuadros, M. (2018). Hate Speech Dataset from a White Supremacy Forum. *arXiv preprint arXiv:1809.04444*.

De la Pena Sarracén, G. L., y Rosso, P. (2019). Aggressive Analysis in Twitter Using a Combination of Models. En *In Proceedings of the First Workshop for Iberian Languages Evaluation Forum (IberLEF 2019), CEUR WS Proceedings.*

Del Vigna, F., Cimino, A., Dell'Orletta, F., Petrocchi, M., y Tesconi, M. (2017). Hate me, Hate me not: Hate Speech Detection on Facebook. En *Proceedings of the first italian conference on cybersecurity (itasec17)* (pp. 86–95).

Fersini, E., Rosso, P., y Anzovino, M. (2018). Overview of the Task on Automatic Misogyny Identification at IberEval 2018. En *IberEval@ SEPLN* (pp. 214–228).

Gambäck, B., y Sikdar, U. K. (2017). Using Convolutional Neural Networks to Classify Hate-speech. En *Proceedings of the First Workshop on Abusive Language Online* (pp. 85–90).

Gao, L., y Huang, R. (2017). Detecting Online Hate Speech using Context aware Models. *arXiv preprint arXiv:1710.07395*.

Gehrmann, S., Dernoncourt, F., Li, Y., Carlson, E. T., Wu, J. T., Welt, J., … others (2017). Comparing rule-based and deep learning models for patient phe-

notyping. *arXiv preprint arXiv:1703.08705*.

Goldberg, Y. (2017). Neural Network Methods for Natural Language Processing. *Synthesis Lectures on Human Language Technologies*, *10*(1), 1–309.

Graff, M., Miranda-Jiménez, S., Tellez, E. S., Moctezuma, D., Salgado, V., Ortiz-Bejar, J., y Sánchez, C. N. (2018). INGEOTEC at MEX-A3T: Author Profiling and Aggressiveness Analysis in Twitter Using $\mu$TC and EvoMSA. En *IberEval@ SEPLN* (pp. 128–133).

Gu, Q., Zhu, L., y Cai, Z. (2009). Evaluation Measures of the Classification Performance of Imbalanced Data Sets. En *International Symposium on Intelligence Computation and Applications* (pp. 461–471).

Hate speech detection on twitter: Feature engineering vs feature selection. (s.f.).

IberEval. (2020). *IberEval: Evaluation of Human Language Technologies for Iberian Languages.* Consultado el 10-02-2020 en https://sites.google.com/view/ibereval-2018.

IberLEF. (2020). *IberLEF: Iberian Languages Evaluation Forum.* Consultado el 22-03-2020 en https://sites.google.com/view/iberlef-2019.

Izenman, A. J. (2008). Modern Multivariate Statistical Techniques. *Regression, classification and Manifold Learning*, *10*, 978–0.

Jurafsky, D., y Martin, J. H. (2014). Speech and language processing. vol. 3. *US: Prentice Hall*.

Kalchbrenner, N., Grefenstette, E., y Blunsom, P. (2014). A Convolutional Neural Network for Modelling Sentences. *arXiv preprint arXiv:1404.2188*.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *arXiv preprint arXiv:1408.5882*.

Krizhevsky, A., Sutskever, I., y Hinton, G. E. (2012). Imagenet Classification with Deep Convolutional Neural Networks. En *Advances in Neural Information Processing Systems* (pp. 1097–1105).

Kumar, R., Ojha, A. K., Malmasi, S., y Zampieri, M. (2018). Benchmarking Aggression Identification in Social Media. En *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)* (pp. 1–11).

Liddy, E. D. (2001). Natural language processing. *Encyclopedia of Library and InformationScience, Marcel Decker*.

LREC. (2020). *Second Workshop on Trolling, Aggression and Cyberbullying.* Consultado el 23-03-2020 en https://sites.google.com/view/trac2/home?authuser=0.

Luque, A., Carrasco, A., Martín, A., y de las Heras, A. (2019). The Impact of Class Imbalance in Classification Performance Metrics based on the Binary Confusion Matrix. *Pattern Recognition*, *91*, 216–231.

MacAvaney, S., Yao, H.-R., Yang, E., Russell, K., Goharian, N., y Frieder, O. (2019). Hate Speech Detection: Challenges and Solutions. *PloS one*, *14*(8), e0221152.

MEX-A3T. (2020). *MEX-A3T: Fake News and Aggressiveness Analysis.* Consultado el 10-02-2020 en https://sites.google.com/view/mex-a3t/home?authuser=0.

Mikolov, T., Chen, K., Corrado, G., y Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., y Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. En *Advances in Neural Information Processing Systems* (pp. 3111–3119).

Müller, K., y Schwarz, C. (2019). Fanning the Flames of Hate: Social Media and Hate Crime. *Available at SSRN 3082972*.

Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., y Chang, Y. (2016). Abusive Language Detection in Online User Content. En *Proceedings of the 25th International Conference on World Wide Web* (pp. 145–153).

Nockleby, J. T. (2000). Hate Speech. *Encyclopedia of the American Constitution*, *3*(2),

1277–1279.

Ortiz, G., Gómez-Adorno, H., Reyes-Magaña, J., Bel-Enguix, G., y Sierra, G. (2019). Detection of Aggressive Tweets in Mexican Spanish Using Multiple Features with Parameter Optimization. En *In Proceedings of the First Workshop for Iberian Languages Evaluation Forum (IberLEF 2019), CEUR WS Proceedings.*

Pennebaker, J. W., Mehl, M. R., y Niederhoffer, K. G. (2003). Psychological Aspects of Natural Language use: Our Words, our Selves. *Annual Review of Psychology*, *54*(1), 547–577.

Qiu, J. X., Yoon, H.-J., Fearn, P. A., y Tourassi, G. D. (2017). Deep learning for automated extraction of primary sites from cancer pathology reports. *IEEE journal of biomedical and health informatics*, *22*(1), 244–251.

Raffel, C., y Ellis, D. P. (2015). Feed-forward Networks with Attention can solve some Long-term Memory Problems. *arXiv preprint arXiv:1512.08756*.

Rangel, F., Rosso, P., Montes-y Gómez, M., Potthast, M., y Stein, B. (2018). Overview of the 6th Author Profiling Task at PAN 2018: Multimodal Gender Identification in Twitter. *Working Notes Papers of the CLEF*.

Russell, C. A., y Miller, B. H. (1977). Profile of a Terrorist. *Studies in conflict & terrorism*, *1*(1), 17–34.

Schmidt, A., y Wiegand, M. (2017). A Survey on Hate Speech Detection using Natural Language Processing. En *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media* (pp. 1–10).

Schütze, H., Manning, C. D., y Raghavan, P. (2008). *Introduction to Information Retrieval* (Vol. 39). Cambridge University Press Cambridge.

Shen, Y., He, X., Gao, J., Deng, L., y Mesnil, G. (2014). A Latent Semantic Model with Convolutional-pooling Structure for Information Retrieval. En *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management* (pp. 101–110).

EEA and Norway Grants. (2020). *Countering hate speech online.* Consultado el 22-03-2020 en https://eeagrants.org/news/countering-hate-speech-online.

Real Time Statistics Project. (2020). *Internet Live Stats.* Consultado el 22-06-2020 en https://www.internetlivestats.com/twitter-statistics/.

U.S. Department of Health Human Services, Stop Bullying. (2020). *Stop Bullying.* Consultado el 19-02-2020 en https://www.stopbullying.gov/cyberbullying/what-is-it.

Vapnik, V. (1996). *The nature of Statistical Learning Theory*. Springer, New York.

Wang, C. (2018). Interpreting Neural Network Hate Speech Classifiers. En *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)* (pp. 86–92).

Warner, W., y Hirschberg, J. (2012). Detecting Hate Speech on the World Wide Web. En *Proceedings of the Second Workshop on Language in Social Media* (pp. 19–26).

Waseem, Z. (2016). Are you a Racist or am I seeing Things? Annotator Influence on Hate Speech Detection on Twitter. En *Proceedings of the First Workshop on NLP and Computational Social Science* (pp. 138–142).

Waseem, Z., y Hovy, D. (2016). Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. En *Proceedings of the NAACL student Research Workshop* (pp. 88–93).

Wiegmann, M., Stein, B., Potthast, M., Cappellato, L., Ferro, N., Losada, D., y Müller, H. (2019). Overview of the Celebrity Profiling Task at PAN 2019. En *CLEF (Working Notes).*

Xiang, G., Fan, B., Wang, L., Hong, J., y Rose, C. (2012). Detecting Offensive Tweets via Topical Feature Discovery over a large scale Twitter Corpus. En *Proceedings of the 21st ACM International Conference on Information and Knowledge Management* (pp. 1980–1984).

Xu, J.-M., Jun, K.-S., Zhu, X., y Bellmore, A. (2012). Learning from Bullying Traces

in Social Media. En *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 656–666).

Yih, W.-t., He, X., y Meek, C. (2014). Semantic Parsing for Single-Relation Question Answering. En *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 643–648).

Zhang, Z., y Luo, L. (2019). Hate speech detection: A solved problem? the challenging case of long tail on twitter. *Semantic Web*, *10*(5), 925–945.

Zhang, Z., Robinson, D., y Tepper, J. (2018). Detecting Hate Speech on Twitter using a Convolution-GRU based Deep Neural Network. En *European Semantic Web Conference* (pp. 745–760).

# Apéndice A

# Competencia y Resultados

Los resultados de la competencia pueden ser consultados en el sitio web del MEX-A3T [1] y por simplicidad se muestra el $F_1$ de los modelos de referencia propuestos por los organizadores así como el $F_1$ de los modelos enviados al concurso. El nombre del equipo con el que se participó es *DeepMath* y la arquitectura con la que se realizaron los pronósticos fue la GRU con *Global Average Pooling* y *Global Max Pooling*. La ejecución 1 corresponde al experimento con esta arquitectura e incluye género y ciencias, mientras que la ejecución 2 también incorpora si el usuario es o no estudiante. Es importante mencionar que la arquitectura y las características de perfilado de autor con las que se participó son los métodos que mejores resultados tenían al momento de enviar las predicciones, después del envío se continuó trabajando, ahora sólo para la tesis y se obtuvo un mejor desempeño.

[1] https://sites.google.com/view/mex-a3t/results?authuser=0

| | $F_1$ (clase agresiva) |
|---|---|
| Baseline (Bi-GRU) | 71.24 |
| DeepMath-1 | 70.01 |
| DeepMath-2 | 69.57 |
| Baseline (BoW-SVM) | 67.60 |

**Tabla A.1:** $F_1$ de los modelos de referencia y los modelos enviados a la competencia para detectar agresividad en el MEX-A3T.