Universidad del Bío-Bío, Chile

Facultad de Ciencias Empresariales

Departamento de Sistemas de Información

# Modelos híbridos basados en Lexicones y Machine Learning para la detección de agresividad sobre textos en idioma Español

Tesis presentada por Manuel Lepe Faúndez
para obtener el grado de Magíster en Ciencias de la Computación
Dirigida por Alejandra Segura
Christian Vidal

2021

# Abstract

In recent years, the use of social networks has increased explosively, which has led to a significant increase in cyberbullying. Currently, in the field of Computer Science, research has been done on how to detect aggressiveness in texts, which is a prelude to detect cyberbullying. In this field the main works have been done for English language texts, mainly using Machine learning approaches, in a smaller percentage Lexicons approaches and very few working use hybrid approaches. That is, they use Lexicons and Machine learning algorithms such as counting the number of bad words in a sentence using a Lexicon of bad words, which serves as an input feature of the classification algorithms.

This research aims to be a contribution in detecting aggressiveness in Spanish language by creating different models that combine the approach of Lexicons and Machine Learning. Twenty-two models that combine techniques and algorithms from both approaches are proposed and for their application certain hyperparameters are adjusted in the training datasets of the corpora to obtain the best results in the test datasets.

Three Spanish language corpus; Chileno, Mexicano and ChilenoMexicano are used in the evaluation. The results indicate that the hybrid models obtain the best results in the 3 corpus, over the implemented models that do not use Lexicons. This shows that by mixing the approaches the aggressiveness detection improves.

Finally, a web application is developed that gives applicability to each model by classifying tweets, allowing to evaluate the performance of the models with external corpus and to receive feedback on the prediction of each of the models for future research. In addition, an API is available that can be integrated in technological tools for parental control, online plugins for the analysis of writing in social networks, educational tools, etc.

# Resumen

En los últimos años ha aumentado de forma explosiva el uso de las redes sociales, lo que ha llevado a un aumento en el ciberacoso de forma significativa. Actualmente, en el campo de la Ciencia de la Computación se ha investigado cómo detectar agresividad en los textos, lo cual es una antesala para detectar ciberacoso. En este campo los principales trabajos se han realizado para textos en idioma inglés, principalmente utilizando enfoques de Machine learning, en un menor porcentaje enfoques de Lexicones y muy pocos trabajando utilizan enfoques híbridos. Es decir, usan Lexicones y algoritmos de Machine learning como por ejemplo contando la cantidad de malas palabras de una frase utilizando un Lexicón de malas palabras, lo que sirve como una característica de entrada de los algoritmos de clasificación.

Esta investigación tiene como propósito ser un aporte en detectar agresividad en idioma español, creando diferentes modelos que combinan el enfoque de Lexicones y Machine Learning. Se proponen 22 modelos que combinan técnicas y algoritmos de ambos enfoques y para su aplicación se ajustan determinados hiperparámetros en los dataset de entrenamiento de los corpus para obtener los mejores resultados en los dataset de prueba.

En la evaluación se utilizan 3 corpus en idioma español; Chileno, Mexicano y Chileno-Mexicano. Los resultados indican que los modelos híbridos obtienen los mejores resultados en los 3 corpus, por encima de los modelos implementados que no utilizan Lexicones. Lo que demuestra que al mezclar los enfoques la detección de agresividad mejora.

Finalmente, se desarrolla una aplicación web que da aplicabilidad a cada modelo clasificando tweets, permitiendo evaluar el rendimientos de los modelos con corpus externos y recibir retroalimentación sobre la predicción de cada uno de los modelos para futuras investigaciones. Además, queda disponible una API que puede ser integrada en herramientas tecnológicas de control parental, plugins en línea para el análisis de escritura en redes sociales, herramientas educativas, etc.

# Índice general

# Índice de figuras

# Índice de tablas

# Capítulo 1

# Introducción

El creciente uso de redes sociales ha proporcionado un nuevo canal para expresar masivamente opiniones y sentimientos sin restricciones. Esto trae consigo un nuevo tipo de acoso, este fenómeno se denomina ciberacoso; el cual se define como el uso de las tecnologías de la información y las comunicaciones, como el correo electrónico, los mensajes de texto de teléfonos celulares, redes sociales, para apoyar el comportamiento deliberado, repetido y hostil de un individuo o grupo con el fin de perjudicar a otros, mediante ataques personales, divulgación de información confidencial o falsa, entre otros medios (Belsey, 2004).

Según los datos extraídos del estudio de Garaigordobil, Mollo, Torrico y Larraín (Garaigordobil et al., 2019), los resultados aportan porcentajes de víctimas y agresores de acoso y ciberacoso en América Latina desde el 2005 al 2018, que evidencian una alta prevalencia de estos problemas, en varios países donde el fenómeno ha sido estudiado, por ejemplo, Colombia, México, Argentina, Brasil, Bolivia, Perú, Chile, Nicaragua, Venezuela, Panamá, Ecuador y Puerto Rico. La revisión muestra que la prevalencia de ciberacoso oscila entre un 3.5 % y 58 % de cibervíctimas; y entre un 2.5 % y 32 % de ciberagresores/(as). Mayoritariamente los implicados en estos hechos son varones. En el caso particular de Chile, según (Campbell y Morgan, 2017) estudio liderado por la PUCV[1] con financiamiento del MINEDUC[2] y en convenio con UNESCO-Santiago, un 20 % reporta haber sido tratado de manera ofensiva o desagradable por otras personas. Estas experiencias son en un 58 % de los casos en persona (sin excluir otras formas), un 28 % por medio de una red social, un 25 % por medio de mensajería, y un 13 % mediante un juego online. Un 14 % de niños que usa internet reconoce haber tratado ofensivamente o de manera desagradable a alguna persona en el último año. El 76 % de estas interacciones ha sido cara a cara, el 19 % por medio de una red social, el 18 % por mensajería y el 11 % mediante mensajes con telefonía celular.

Generalmente, el acoso y el ciberacoso comienzan en la escuela primaria, luego sigue en la secundaria donde alcanza su punto máximo. En algunos casos estas conductas siguen después de la secundaria, resultando en conductas de aislamiento social, ausencia de clases y bajas calificaciones (Hicks et al., 2018) (Campbell y Morgan, 2017).

[1] https://www.pucv.cl/
[2] https://www.mineduc.cl/

La Tabla 1.1 muestra los tipos de ciberacoso que se pueden presentar según (Willard). Se puede observar que comparten elementos comunes como la intención de dañar, contienen mensajes ofensivos y/o contenido agresivo. Por lo tanto, generalmente el ciberacoso se presenta con mensajes agresivos, reiterados hacia una persona por parte de otra o de un grupo de personas, donde se identifica a un agresor y una víctima. Por lo que el identificar mensajes agresivos sirve como antesala para detectar ciberacoso si se cumplen una serie de condiciones.

El Procesamiento del Lenguaje Natural (PLN) es un área de investigación y de aplicación que explora cómo las computadoras pueden ser utilizadas para comprender y manipular en texto las expresiones del ser humano (Scherer, 1984). El PLN abarca distintas disciplinas como lo son la computación y ciencias de la información, la lingüística, matemática, inteligencia artificial, psicología, entre otras.

En los últimos años se ha vuelto popular el uso de distintas técnicas para identificar las emociones que quiere transmitir el autor. Como subcategoría del PLN se puede encontrar el análisis de subjetividad de texto que se encarga de extraer y clasificar las distintas emociones que quiere transmitir el autor de un texto y con esto obtener información valiosa para analizar y apoyar la toma de decisiones.

El análisis de la subjetividad en texto nos entrega una herramienta para detectar la agresividad en los textos de las redes sociales, mientras más temprano se detecten estas conductas mejores serán las oportunidades que tendrá la sociedad para tomar las medidas remediales e incluso preventivas.

Dado que los mayores aportes en este tema son realizados para textos en inglés, el presente trabajo se enfoca en detectar agresividad sobre textos en español. Trabajos previos del grupo de investigación SoMos[3] de la Universidad del Bío-Bío han permitido avanzar sobre este tema probando 2 enfoques, el enfoque de Lexicones y el de Machine learning. En particular esta investigación se focaliza en el enfoque híbrido, creando diferentes modelos que usen este enfoque y comparando sus resultados con modelos implementados que no usan Lexicones con el propósito de medir el rendimiento de este nuevo planteamiento y analizar si permite mejorar la detección de agresividad en textos en español.

Lo que resta de este informe está estructurado de la siguiente forma. En el Capítulo 2 se presenta la hipótesis de la tesis, el objetivo general, los objetivos específicos, alcance y limitaciones de la investigación y metodología de trabajo. En el Capítulo 3 se presentan los conceptos fundamentales y la revisión de la literatura. En el Capitulo 4 se describen los distintos corpus utilizados. En el Capítulo 5 se presentan de forma detallada los distintos enfoques y modelos creados para detectar agresividad. En Capítulo 6 se muestran los resultados de los experimentos realizados con los modelos en los distintos corpus. En el Capítulo 7 se presenta la aplicación web desarrollada para dar aplicabilidad a los distintos modelos. Finalmente, en el Capítulo 8 se presentan las conclusiones y posibles trabajos futuros.

[3] https://dsi.face.ubiobio.cl/somos/

| **Nombre del tipo de ciberacoso** | **Descripción** |
|---|---|
| Flaming | Enviar mensajes agresivos, groseros y vulgares dirigidos a una o más personas por privado o en un grupo en línea. |
| Acoso | Envío de mensajes agresivos, groseros y vulgares a una persona de forma repetitiva. |
| Cyberstalking | Acoso que incluye amenazas de daño o es altamente intimidante. |
| Denigración | Enviar o publicar declaraciones dañinas, agresivas, falsas o crueles sobre una persona a otras personas. |
| Suplantación | Fingir ser otra persona y enviar o publicar material que haga que esa persona se vea mal o que la ponga en peligro. |
| Outing and trickery | Enviar o publicar material sobre una persona que contenga información sensible, privada o embarazosa, incluyendo el reenvío de mensajes o imágenes privadas. Haciendo trucos para solicitar información embarazosa que luego se hace pública. |
| Exclusión | Acciones que específicamente e intencionalmente excluyen a una persona de un grupo en línea. |

Tabla 1.1: Tipos de ciberacoso según (Willard)

# Capítulo 2

# Propuesta de tesis

## 2.1. Hipótesis

Los modelos híbridos que mezclan el enfoque de Lexicones y Machine Learning permiten mejorar el rendimiento de la predicción de agresividad presente en textos en idioma español.

## 2.2. Objetivo general

Crear y evaluar distintos modelos híbridos para identificar agresividad en textos en idioma español, los que quedarán disponibles en una plataforma web que permitirá recibir retroalimentación de los distintos usuarios.

## 2.3. Objetivos específicos

- Revisar el estado del arte de los distintos trabajos que tengan como objetivo predecir agresividad en texto usando modelos de Machine Learning y Lexicones, principalmente en idioma español.

- Crear diferentes modelos híbridos; usando el enfoque de Lexicones y Machine Learning, principalmente para la extracción de características del texto.

- Comparar el rendimiento de los distintos modelos creados en diferentes corpus en idioma español mediante una herramienta web que, además, quedará disponible para darle aplicabilidad a los modelos creados y recibir retroalimentación de los usuarios.

- Analizar los resultados obtenidos por los distintos modelos para generar conclusiones objetivas y proponer trabajos futuros.

## 2.4. Alcance de la investigación

En esta tesis se utilizarán Lexicones ya creados, por lo tanto, no se crearán Lexicones nuevos. Por otra parte, los algoritmos de Machine Learning se implementarán mediante la librería Scikit-learn[1] para Python. Por último, para evaluar los diferentes modelos creados, no se construirán nuevos corpus, se utilizarán corpus en idioma español ya creados.

## 2.5. Metodología de trabajo

La metodología de trabajo considera las siguientes actividades:

1. Completar el estado del arte revisando los modelos y técnicas mas recientes aplicadas sobre corpus en idioma español, especialmente revisando modelos de los distintos teams que participaron en el desafío de predecir agresividad en los workshops que utilizan el corpus mexicano que se creó en (Álvarez Carmona et al., 2018). Con esta revisión se tiene en cuenta los diferentes modelos y técnicas para crear de una mejor forma los modelos híbridos. Especialmente, se estudia el uso de Word Embedding que utiliza el modelo ganador en (Aragón et al., 2019) y cómo los modelos que mezclan distintos clasificadores basados en Lexicones y Machine Learning obtienen buenos resultados, como se indica en (Álvarez Carmona et al., 2018).

2. Para realizar los nuevos modelos se toma en cuenta el estudio que se lleva a cabo en la actividad 1, además de evaluar los nuevos modelos en forma constante para identificar cuáles de ellos obtienen mejores resultados.

3. Selección y obtención de los distintos corpus para evaluar los modelos híbridos creados. Esta actividad se lleva a cabo teniendo en cuenta los corpus utilizados en los diferentes trabajos que utilizan corpus en idioma español.

4. Evaluación los diferentes modelos creados en los corpus seleccionados, usando distintas métricas para llegar a conclusiones acertadas y objetivas.

5. Desarrollo de la plataforma web que permite dar aplicabilidad a los distintos modelos creados. Entre sus funciones está permitir clasificar frases ingresadas por el usuario, tweets de una tendencia en Twitter o que contenga palabras específicas y tweets de un usuario específico, usando los diferentes modelos implementados. Por otra parte, permite tener una retroalimentación del usuario cuando los clasificadores predicen un resultado, indicando si el usuario considera que la clasificación es correcta o no, esto se guarda en una base de datos.

[1] https://scikit-learn.org/

# Capítulo 3

# Estado del arte

Para una mejor comprensión de este documento, a continuación se presentan los conceptos fundamentales y definiciones asociadas, junto con los resultados de la revisión sistemática de literatura realizada.

## 3.1. Marco conceptual

### 3.1.1. Definición de agresividad

La definición que entrega la Real Academia Española (RAE) al termino de agresividad es el siguiente: "Tendencia de actuar o responder violentamente" [1]. Además, hace referencia al concepto de acometividad, que es la propensión para acometer, atacar, embestir. De esta definición podemos concluir que la agresividad es un conjunto de patrones de actividad que puede manifestarse con intensidad variable, desde las expresiones verbales y gestuales hasta la agresión física. En el lenguaje cotidiano se asocia la agresividad con la falta de respeto, la ofensa o la provocación.

Por lo tanto, podemos definir que un texto manifiesta una actitud agresiva de su autor hacia otra persona, cuando tenga contenido violento, ofensivo, con falta de respeto o con provocación.

### 3.1.2. Definición de Lexicón

Un Lexicón se define como un documento que contiene palabras que ya se encuentran etiquetadas con su polarización, son parte de una clase afectiva o tienen una característica en común (Wilson et al., 2005). En esta tesis se utilizan dos tipos de Lexicones:

**Lexicón de intensidad:** Lexicón con palabras etiquetadas con la intensidad (del 10 al 100, donde 10 es baja intensidad y 100 la máxima) de una clase afectiva, es decir un Lexicón por cada una de las 8 clases afectivas definidas por Plutchik (PLUTCHIK, 1980). La Tabla

[1] https://dle.rae.es/agresividad

3.1 muestra un extracto del Lexicón de intensidad (Segura Navarrete et al., 2021) para la clase afectiva enojo.

| **Palabra** | **Intensidad** |
|---|---|
| Bronca | 88 |
| Cabreao | 75 |
| Furia | 75 |
| Inspirar | 10 |

Tabla 3.1: Extracto Lexicón de intensidad para enojo

**Lexicón de características:** Lexicón que contiene palabras que comparten una característica en común. Para el caso de esta tesis se utilizas Lexicones que contienen malas palabras y Stop Word del español. La Tabla 3.2 muestra un extracto del Lexicón de malas palabras.

| **Palabra** |
|---|
| cornudo |
| fecas |
| conchetumare |
| aweonao |
| weko |

Tabla 3.2: Extracto Lexicón de malas palabras

### 3.1.3. Procesamiento del Lenguaje Natural

El Procesamiento del Lenguaje Natural (PLN) es un área de investigación y de aplicación que explora cómo las computadoras pueden ser utilizadas para comprender y manipular en texto las expresiones del ser humano (Scherer, 1984). El PLN abarca distintas disciplinas como lo son la computación y ciencias de la información, la lingüística, matemática, inteligencia artificial, psicología, y más. En la actualidad, el PLN está siendo aplicado a distintos dominios, tales como la traducción automática (Johnson et al., 2017)], responder preguntas de forma automática (Ferrucci et al., 2010), generación de lenguajes naturales (Brown et al., 2020), síntesis de voz (Morise et al., 2016), reconocimiento del habla (Yu y Deng, 2015), entre otras.

### 3.1.4. Subjetividad de textos

En los últimos años se ha vuelto popular el uso de distintas técnicas para identificar las emociones que quiere transmitir el autor. Como subcategoría del PLN se puede encontrar el análisis de subjetividad de texto que se encarga de extraer y clasificar las distintas emociones que quiere transmitir el autor de un texto y con esto obtener información valiosa para analizar y apoyar la toma de decisiones. Un ejemplo de esto es el análisis que puede realizar una compañía respecto a las opiniones positivas o negativas de los usuarios de una red social, es decir, la compañía al analizar estas opiniones puede descartar o afirmar supuestos que se planteen, por ejemplo, agregar una nueva característica a un determinado producto. Dentro de la subjetividad de textos podemos identificar 2 áreas: Análisis de Sentimientos y Análisis de Emociones.

**Análisis de sentimientos:** El Análisis de Sentimientos se encarga de identificar sentimientos en el texto, utilizando herramientas computacionales para formalizar y polarizar este contenido (Witten et al., 2011) . Al analizar un texto se podrá obtener la polarización que presenta ya sea positiva, negativa o neutra. En la Tabla 3.3 se muestra distintas frases y su polaridad definida.

| **Frase** | **Polaridad** |
|---|---|
| “muy buen profesor excelente profesional.” | Positiva |
| “no me gusto el ramo no lo entendi no aprendi nada” | Negativa |
| “Sin comentarios” | Neutra |

Tabla 3.3: Ejemplo análisis de sentimiento

**Análisis de emociones:** Incluye un conjunto de técnicas del Procesamiento del Lenguaje Natural para detectar la emoción expresada en un texto (Grefenstette et al., 2004). Las emociones se clasifican en distintas categorías, emociones básicas y complejas. Dichas propuestas provienen del área de la psicología. Existen variadas taxonomías para clasificar las emociones, en particular, para este trabajo se utilizará la clasificación de Plutchik; que plantea que todas las emociones pueden ser clasificadas en 8 categorías básicas (PLUTCHIK, 1980); Enojo (ira), Anticipación, Disgusto (aversión), Miedo, Alegría, Tristeza, Sorpresa y Confianza. Por otra parte, Plutchik también señala una serie de emociones avanzadas que surgen de la combinación de las consideradas como básicas. Así, la unión de la alegría con la anticipación es el optimismo; el amor es la unión de la confianza con la alegría; la sumisión es la suma del miedo más la confianza; el susto es la sorpresa más el miedo; la decepción, la unión de la tristeza con la sorpresa; el remordimiento, la aversión más la tristeza; el desprecio es la suma de la ira con la aversión, y la anticipación más la ira da como resultado la alevosía. Esto se plasma de forma gráfica en la Figura 3.1 denominada la rosa de Plutchik.

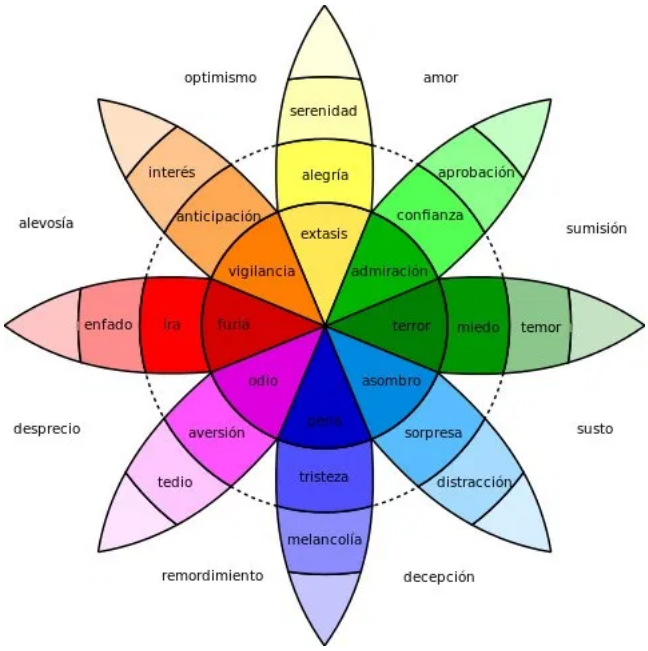


Figura 3.1: Rosa de Plutchik

### 3.1.5. Enfoques para el análisis de subjetividad de textos

A continuación se describen los 2 enfoques que existen para realizar el análisis de la subjetividad de textos, ya sea para el análisis de sentimiento o de emociones (Reagan et al., 2015).

**Enfoque basado en Lexicones:** Es un enfoque semántico donde se define un documento (Lexicón) que contiene una serie de palabras polarizadas (Wilson et al., 2005) o clasificadas en una determinada categoría de emoción (Strapparava y Mihalcea, 2008). Para realizar el análisis se definen determinadas reglas, por ejemplo, para el análisis de sentimientos lo más básico es contar las palabras positivas, negativas o neutras según el Lexicón y la polaridad que tenga mayor cantidad de palabras, es la polaridad resultante del texto. De la misma forma se puede realizar el análisis de emociones. En (Mohammad, 2011) se presentan diferentes análisis de sentimientos y emociones usando Lexicones, sobre corpus de gran tamaño. Por otra parte, en (Molina Beltrán et al., 2019) se crea un Lexicón reducido solo con palabras emotivas enriquecido con la intensidad de cada clase afectiva a la cual pertenece, este Lexicón incluye solo palabras en idioma inglés. En esta misma linea, en (Segura Navarrete et al., 2021) se extiende este Lexicón para idioma español, lo que permite ser utilizado para realizar análisis de subjetividad en texto en español.

**Enfoque basado en Machine Learning:** Machine Learning (Aprendizaje automático) se define como el campo de estudio que le da a las computadoras la habilidad de aprender

sin haber sido explícitamente programadas (Samuel, 1969). Para realizar el análisis de sentimiento o emociones, se utilizan distintos algoritmos de Machine Learning que identifican tendencias o patrones en los datos que permiten posteriormente estimar la polaridad o emoción que presenta un texto. Existen dos grandes categorías de Machine Learning que, básicamente, se diferencian en la forma de entrenar los algoritmos:

- **Aprendizaje supervisado:** El aprendizaje se realiza a partir de un conjunto de datos, es decir, para lo que el valor del atributo objetivo o que interesa estimar (por ejemplo, polaridad) es conocido. En base a esto los algoritmos identifican patrones que permiten estimar el valor del atributo objetivo en nuestros datos.
- **Aprendizaje no supervisado:** A diferencia del aprendizaje supervisado los datos de entrenamientos no tienen especificados los resultados. Por lo tanto, el algoritmo va aprendiendo, ajustando los datos o maximizando una función objetivo.

### 3.1.6. Algoritmos de Machine Learning clasificadores

De los enfoques utilizados en Machine Learning descritos anteriormente el más utilizado es el aprendizaje supervisado, debido a la forma de trabajar, que puede verificar los resultados de la clasificación para poder estudiar el rendimiento del algoritmo, que es uno de los objetivos de este trabajo. Existen distintos algoritmos de Machine Learning supervisado que permiten estimar la probabilidad con que un "individuo" pertenezca a una clase, generalmente estas clases son excluyentes. Con esta probabilidad se puede clasificar a cada "individuo" en una clase. Como por ejemplo, la clasificación de una frase, si esta es agresiva o no agresiva. Donde existen dos clases y los algoritmos deben determinar la probabilidad que tienen la frase para cada una de ellas, se clasifica la frase con la clase que tenga mayor probabilidad.

Para el análisis de emociones se han aplicado distintos algoritmos de Machine Learning de clasificación. En (Elgueta, 2017) se realiza una comparación de los algoritmos; Árboles de decisión, Naive Bayes y Support Vector Machine para identificar las emociones que presentan los titulares de diarios chilenos. Cabe recalcar la importancia de este estudio, ya que la mayoría de los estudios en que se han aplicado estos algoritmos se han enfocado sobre textos en inglés. Como resultado de la investigación se observa que la técnica de Support Vector Machine obtiene los mejores resultados.

A continuación, se describen los algoritmos de Machine Learning supervisados de clasificación utilizados en (Elgueta, 2017), para el que caso del algoritmo Árboles de decisiones se presenta Random Forest que es un derivado de este.

**Random Forest:** Algoritmo de Machine learning presentado en (Breiman, 2001) que combina el uso de varios árboles de decisión, construidos a partir de los datos de entrenamiento de forma independiente. Para construir cada árbol de forma independiente se elige la mejor característica que divide el nodo de un árbol, entre un subconjunto aleatorio de características, a diferencia de los árboles de decisiones tradicionales que se construyen eligiendo siempre la mejor característica entre todas ellas. El principio detrás de Random

Forest es simple, un conjunto de muchos árboles de decisiones que actúen como un "comité", tendrá mejores resultados que un solo árbol de decisión. Es por esto que Random Forest entra en la categoría de los algoritmos de Machine Learning Ensemble Learning (Opitz y Maclin, 1999). La clave para tener mejores resultados es la baja correlación entre los árboles del modelo y el número de árboles que se construyen, con esto se disminuye el sobreajuste (del inglés overfitting) a los datos de entrenamiento que presentan los árboles de decisiones cuando son muy extensos. En la Figura 3.2 se puede observar un ejemplo de cómo se realiza una predicción en el algoritmo Random Forest, se aprecia 9 árboles de decisión, donde cada uno predice de forma independiente, 6 predijeron 1 y 3 predijeron 0, por lo tanto, la predicción final será 1.

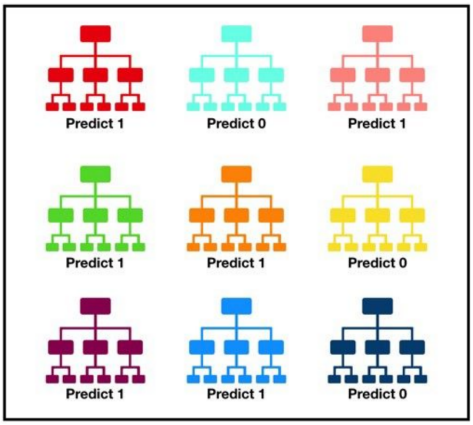


Figura 3.2: Ejemplo Random Forest

**Naive Bayes:** Este algoritmo se basa en el teorema de Bayes y en la premisa de la independencia de los atributos de una clase, esto quiere decir que el comportamiento estadístico de un atributo no se ve afectado por los valores que toman los otros atributos (por esto es que se llama Naive de ingenuo). Considerando el teorema de Bayes, se expresa como se muestra en la ecuación 3.1.6:

$$\begin{gathered} P(y|x_1,...,x_n)\alpha P(y)\prod_{i=1}^{n}P(x_i|y) \\ \Downarrow \\ \hat{y} = arg_y max P(y)\prod_{i=1}^{n}P(x_i|y) \end{gathered} \tag{3.1}$$

Donde $x_i$ son los atributos (vector de características), y es la clase que se quiere estimar, por ejemplo, agresivo o no agresivo. Las probabilidades de $P(y)$ y $P(x_i|y)$ son calculadas mediante los datos de entrenamiento. Los diferentes clasificadores Naive Bayes difieren principalmente por los supuestos que hacen con respecto a la distribución de $P(x_i|y)$. Pese

a su simplicidad, los clasificadores Naive Bayes han funcionado bastante bien en muchas situaciones del mundo real como en la clasificación de documentos y de spam. Requieren una pequeña cantidad de datos de entrenamiento para estimar los parámetros necesarios (Zhang, 2004).

**Support Vector Machines:** Este algoritmo genera un hiperplano discriminatorio que permite separar los datos usando discriminantes lineales. Esto se realiza usando una función que se denomina kernel, la cual, permite transformar los datos a una dimensionalidad mayor y en esta dimensión buscar un hiperplano con el máximo margen entre los vectores de soporte (Hsu et al., 2008). El uso de los vectores de soporte permite generar un modelo más preciso, donde el hiperplano se encuentra a la misma distancia de los vectores que representan el límite de cada clase. A modo de ejemplo, la Figura 3.3 [2] muestra los vectores de soporte en línea punteada y el hiperplano como una línea negra.

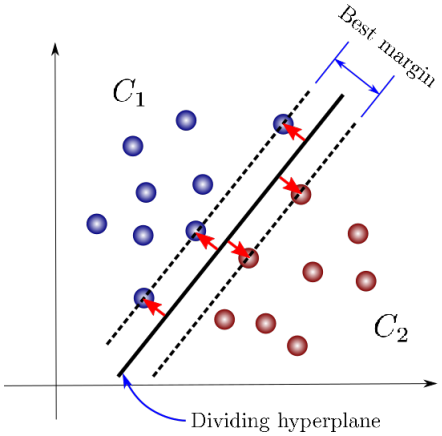


Figura 3.3: Representación de los vectores de soporte y el hiperplano

El mayor número de puntos separables por un modelo de clasificación (Vapnik - Chervonenkis) es k+1, donde k es la dimensión. Por ejemplo, en el plano 2D siempre se va a encontrar un hiperplano en 2D que separe 3 puntos, pero para 4 puntos no siempre se va a cumplir esto (Blumer et al., 1989). En la Figura 3.4[3] se puede observar que en el gráfico de la izquierda no es posible aplicar un discriminante lineal, por lo tanto, se transforma mediante la función Kernel a la dimensión 3D, donde sí es posible.

[2] Recuperado desde `https://bit.ly/34DH1Uj` el 06/2020.
[3] Recuperado desde `https://bit.ly/2LORwgZ` el 06/2020.

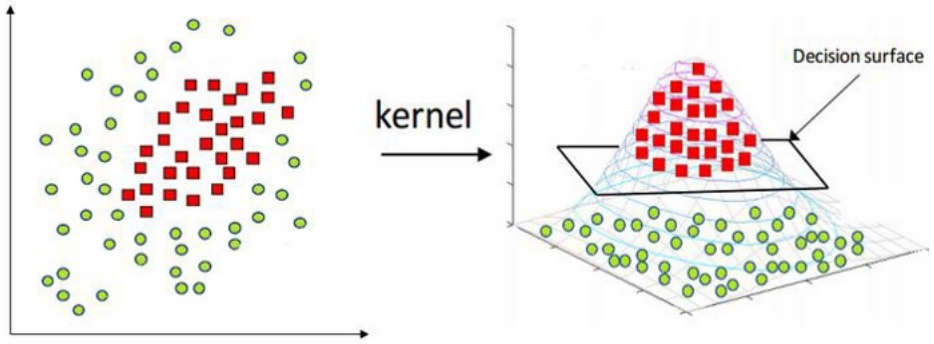


Figura 3.4: Cambio de dimensionalidad

### 3.1.7. Métricas utilizadas para medir el rendimiento de los algoritmos de clasificación

Existen diferentes métricas estadísticas para poder medir el rendimiento de un modelo de Machine Learning de clasificación. A continuación, se describen las métricas que se utilizan para entender el propósito de cada una de ellas. Como el objetivo de los modelos que se crean en esta tesis es estimar si un texto es agresivo o no agresivo, se utiliza el concepto de positivo cuando es agresivo y negativo cuando es no agresivo.

En (Nordhausen, 2009), se describen 4 conceptos que son la base para entender las métricas, que se presentan a continuación:

**Verdaderos Positivos (VP):** Se refieren cuando la clasificación es positiva y la predicción realizada también lo es. Para el caso de esta tesis se refiere cuando un texto es etiquetado como Agresivo y la predicción del modelo también es Agresivo.

**Verdaderos Negativos (VN):** Cuando la clasificación es negativa y el modelo también predice negativa. En nuestro caso cuando la etiqueta del texto es no agresiva y el modelo también predice no agresivo.

**Falsos Positivos (FP):** Se refiere cuando el modelo realiza una predicción de positivo, cuando realmente era negativo, es decir el modelo se equivoca. En nuestro caso, cuando el modelo predice agresivo y realmente es no agresivo.

**Falsos Negativos (FN):** Cuando el modelo predice negativo cuando realmente es positivo. Para esta propuesta, cuando el modelo predice No agresivo cuando realmente estaba etiquetado como Agresivo.

Entendiendo estos conceptos se pueden definir las métricas que se utilizan:

**Accuracy:** Métrica básica que simplemente muestra la proporción de las instancias que acertó con el total de instancias. La ecuación 3.2 muestra la métrica Accuracy.

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN} \tag{3.2}$$

**Precision:** Se refiere al porcentaje de casos positivos que fueron predichos de forma correcta, se define como la razón entre los verdaderos positivos entre todos los predichos como positivos. La ecuación 3.3 muestra la métrica Precision.

$$Precision = \frac{VP}{VP + FP} \tag{3.3}$$

**Recall** Refleja qué tan completa fue la predicción de positivos, se calcula dividiendo la cantidad de datos predichos positivamente entre la cantidad de datos que realmente eran positivos. La ecuación 3.4 muestra la métrica Recall.

$$Recall = \frac{VP}{VP + FN} \tag{3.4}$$

**F-measure:** Se interpreta como un promedio ponderado entre la Precision y el Recall. La ecuación 3.5 muestra la métrica F-measure.

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.5}$$

## 3.2. Revisión de la literatura

Se realizó una revisión sistemática de la literatura siguiendo el protocolo propuesto por (Kitchenham, 2004). El cual se organiza en 3 etapas: planificación, realización y por último, un informe de presentación de los resultados. El objetivo es conocer el estado del arte de la detección de agresividad y/o cyberbullying usando Machine Learning supervisado y Lexicones, además de buscar trabajos que combinen los dos enfoques y que tengan aplicabilidad en el idioma español. Se eligieron las siguientes librerías digitales para extraer los artículos: ScienceDirect[4], IEEE[5], ACM[6] y actas del congreso y revista SEPLN[7]. Después de aplicar todos los filtros, se seleccionaron 10 artículos (Gordeev, 2016) (Kumar Sharma et al., 2018) (Serhrouchni, 2014) (Pawar et al., 2018) (Balakrishnan et al., 2019) (Al-garadi et al., 2016) (Ptaszynski et al., 2016) (Del Bosque y Garza, 2014) (Murnion et al., 2018) (Chatzakou et al., 2017). A continuación, se presentan los principales resultados.

[4] https://www.sciencedirect.com/
[5] https://ieeexplore.ieee.org/
[6] https://dl.acm.org/
[7] http://www.sepln.org/sepln

### 3.2.1. Idioma del corpus utilizado por los artículos

La gran mayoría de los trabajos seleccionados realizan experimentos sobre corpus en idioma inglés, como ha sido la tónica los últimos años. Además, se observa que no se encontraron trabajos relacionados que realicen experimentos sobre textos en español. La Tabla 3.4 muestra la cantidad de artículos por idioma.

| | **Inglés** | **Ruso** | **Japonés** | **Árabe** | **Español** |
|---|---|---|---|---|---|
| **Cantidad de artículos** | 9 | 1 | 1 | 1 | 0 |

Tabla 3.4: Cantidad de artículos por idioma

### 3.2.2. Enfoques que utilizan los trabajos para detectar agresividad

Más del 50 % de los artículos seleccionados usan solo enfoque de Machine Learning, sin embargo, existe un porcentaje importante que combina los 2 enfoques, 30 %, y finalmente, el enfoque menos usado es el de Lexicón con 10 %. En la Figura 3.5 se puede observar un gráfico de anillo en donde se visualizan los porcentajes de cada enfoque.

El trabajo que solo usa el enfoque de Lexicón (Ptaszynski et al., 2016), se basa en un Lexicón de 9 malas palabras seleccionadas por los autores de acuerdo a su alta frecuencia en frases que contienen Cyberbullying, se aplica análisis morfológico y técnicas de recuperación de información para determinar el grado de Cyberbullying que contiene cada frase.

Por otro lado, los trabajos (Al-garadi et al., 2016) (Del Bosque y Garza, 2014) (Chatzakou et al., 2017) fueron etiquetados como híbridos dado que buscaban el número de malas palabras que había en una frase para posteriormente entregárselo como característica a los algoritmos de Machine Learning, esto lo hacen mediante un documento donde tenían guardadas ciertas malas palabras.

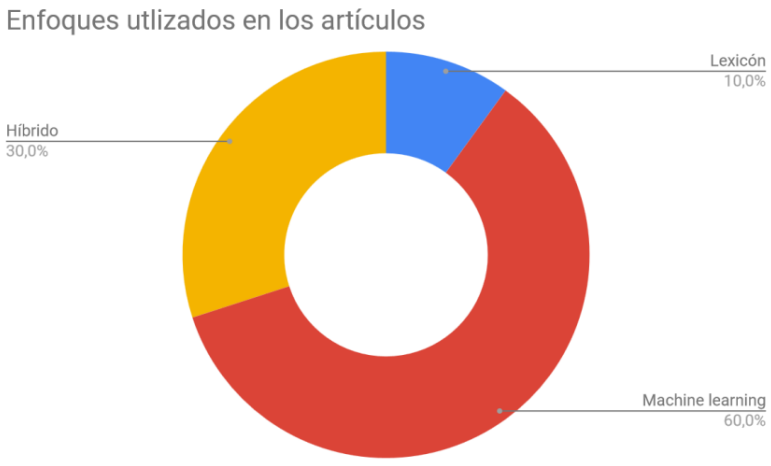


Figura 3.5: Porcentaje de artículos por enfoque utilizado

### 3.2.3. Algoritmos de Machine Learning más utilizados

En la Tabla 3.5 se muestran los algoritmos de Machine Learning más utilizados por los artículos seleccionados. Cabe destacar que estos algoritmos son los que más se utilizan en este tipo de modelos, principalmente porque son clasificadores básicos y tienen buen rendimiento.

| **Algoritmos de Machine Learning** | **N° de artículos que lo utilizan** |
|---|---|
| Naive Bayes | 4 |
| Support Vector Machine | 3 |
| Random Forest | 3 |

Tabla 3.5: Algoritmos de Machine Learning más utilizados

### 3.2.4. Trabajos en corpus en español

Al no encontrar artículos que se hayan aplicado sobre corpus en idioma español en las fuentes digitales elegidas en la revisión, se procedió a realizar una extensa búsqueda en distintos medios digitales (Google, Google Scholar y actas de congresos). A continuación, se describen los artículos encontrados en esta búsqueda.

En (Leon-Paredes et al., 2019) se crean 3 corpus a partir de Twitter; Corpus pequeño (25.304 tweets), Corpus mediano (229.801 tweets) y Corpus Grande (960.578 tweets). Para el etiquetado ("Presunto cyberbullying" o "Sin cyberbullying") de cada tweets se hace de forma automática teniendo en cuenta el "Inventario general de insulto" (Celdrán, 2009), además de agregar insultos propios de Ecuador y los patrones detectados en (Tapia et al., 2018). El modelo se crea usando solo algoritmos de Machine Learning (Naive Bayes, Support Vector Machine y Logistic Regression) y se forma el vector de característica mediante la técnica de TF-IDF[8]. Al evaluar el modelo en los distintos corpus se encuentra un Accuracy entre 80 % y 91 % en promedio, siendo Support Vector Machine el que obtiene un mejor resultado en el corpus mediano con un peak de 94 % de accuracy. Además, se implementa una web[9] donde se puede evaluar en tiempo real desde Twitter el porcentaje de cyberbullying en 3 escenarios; Análisis de frases, análisis de un perfil de twitter (teniendo en cuenta sus últimos tweets) y análisis de una tendencia. Cabe destacar que los corpus y código fuente están disponibles bajo la Licencia Pública General de GNU (v3 o posterior)[10].

En (Mercado et al., 2018)] se entrena un clasificador Naives Bayes mediante la librería NLTK (Loper y Bird, 2002) para Python, para el entrenamiento se utiliza el Lexicón (Rıos

[8] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[9] https://cloudcomputing.ups.edu.ec/SCPSystem

[10] https://www.gnu.org/licenses/gpl-3.0.html

y Gravano, 2013) además de agregar 595 palabras etiquetadas manualmente. Para representar las frases se usa el popular método Bolsa de Palabras (del inglés Bag of Words) mediante una colección de malas palabras. El modelo entrega el porcentaje de "aceptación" que tiene cada frase, donde 0 % indica alta probabilidad de que la frase contenga bullying y 100 % baja probabilidad. Para evaluar el modelo, los autores recuperaron 100 frases de redes sociales y 10 personas las etiquetan de 0 a 100 (siguiendo el mismo enfoque de los porcentajes), se calcula el promedio y se compara con el porcentaje entregado por el modelo. En el 93 % de los casos el modelo clasificó (frase con bullying, sin bullying o neutra según el porcentaje) igual que el promedio del etiquetado manual. En este artículo se utilizaron términos propios de Perú.

En el workshop[11] del año 2018 (Álvarez Carmona et al., 2018) organizado por IberEval para el desafío de detectar agresividad se elabora un corpus de 10.856 instancias (7.700 para entrenamiento y 3.153 para evaluación) etiquetado manualmente ("Agresivo" o "noAgresivo"), el corpus corresponde a tweets en idioma español situados en un radio de 500Km de ciudad de México. Este workshop se volvió a realizar en el año 2019[12] y 2020[13] con el mismo desafío y corpus, a continuación se presentan una revisión de los principales trabajos que realizaron los equipos de cada workshop.

### Trabajos workshop año 2018

Los participantes propusieron una variedad de metodologías, que comprendían características basadas en el contenido (bolsa de palabras, palabra n-gramas, vectores de términos, etc.) y características basadas en el estilo (frecuencias, puntuaciones, POS, elementos específicos de Twitter, etc.), así como algoritmos clásicos de Machine Learning (Naive Bayes, SVM, Logistic regression, etc) y Redes Neuronales.

El equipo ganador obtiene una **F-measure promedio de 0.620 y un Accuracy de 0.667** sobre el corpus de prueba, presentan su modelo en (Sánchez Gómez, 2018), el modelo lo denominan EvoMSA[14] que combina principalmente los siguientes modelos:

- B4MSA: Modelo que se basa en el clasificador Support Vector Machine .
- Dos modelos basado en Lexicones:
  Up-Down: Cuenta las palabras positivas y negativas de las frases.
  Bernoulli Model: Creado para predecir agresividad usando un Lexicón con palabras agresivas.
- EvoDAG: Programación genética con operadores semánticos que realiza la predicción final mediante la combinación de todos los valores de la función de decisión.

[11] https://mexa3t.wixsite.com/home
[12] https://sites.google.com/view/iberlef-2019/home
[13] https://sites.google.com/view/mex-a3t/
[14] https://github.com/INGEOTEC/EvoMSA/tree/master/EvoMSA

El equipo que queda en segundo lugar obtiene una **F-measure promedio de 0.605 y un Accuracy de 0.667**, presentan sus modelo en (Cuza et al., 2018).El modelo que ellos desarrollaron se basan en una red neuronal compuesta por una bi-LSTM, una capa de atención y una Post-Attention LSTM, que finalmente predice si es agresivo o no el texto. Para extraer las características del texto, este se convierte en vector mediante Word Embedding (después de realizar un pre-procesamiento del texto), además agregan características lingüística; se define la presencia o no de palabras obscenas o vulgares en los tweets acorde a un Lexicón.

### Trabajos workshop año 2019

En el workshop del año 2019 el equipo ganador iguala en métricas con el primer lugar del año 2018, pero como lo señalan en (Aragón et al., 2019) el modelo es más simple, ya que para la extracción de las características usan Word Embedding y n-gramas y para clasificar usan Multilayer Perceptron. El equipo ganador presenta su trabajo en (Casavantes et al., 2019), donde experimentan con diferentes características como la ocupación y locación del autor del tweets (lo predicen mediante un modelo no supervisado), agrupan los tweets por tema, etc. Además de probar con el clasificador Support Vector Machine. Es importante destacar que los mejores resultados se dieron usando n-gramas, Word Embedding y Multilayer Perceptron como clasificador.

### Trabajos workshop año 2020

El equipo ganador del workshop del año 2020 obtiene una **F-measure promedio de 0.8596 y un Accuracy de 0.8851**, presentan su trabajo en (Tanase et al., 2020), que consiste en 2 modelos que se resumen a continuación:

- El primer modelo presentado por el equipo se basa en 20 BETOS ajustados mediante el corpus de entrenamiento para predecir agresividad, con esquemas de voto mayoritario y ponderados. BETO[15] (Cañete et al., 2020) es un modelo entrenado sobre idioma español basado en BERT (Devlin et al., 2019) el cual se basa en la arquitectura del encoder de Transformer. En general BERT es utilizado como un método para generar modelos de lenguaje pero es posible ajustarlo para desempeñar funciones de clasificación. Este es el modelo que obtiene el primer lugar en el workshop del año 2020.
- El segundo modelo presentado por el equipo además de los BETOS utiliza la técnica de aumento de datos, obtiene el segundo lugar con una **F-measure promedio de 0.8588 y Accuracy de 0.8858**. Algunas técnicas que se utilizan para aumentar los datos de entrenamiento son el reemplazo de algunas palabras por algún sinónimo, intercambiar las posiciones de palabras, eliminar palabras de una frase, etc.

El equipo que queda en tercer lugar obtiene una **F-measure promedio de 0.8538 y un Accuracy de 0.8759**, presentan sus modelos en (Guzman-Silverio et al., 2020). Los

[15] https://github.com/dccuchile/beto

autores presentaron diferentes enfoques para afinar modelos pre-entrenados en español, inglés y multilingües basados en Transformer. El mejor resultado que obtuvieron fue el uso de BETO, pero ajustado con el conjunto de entrenamiento del workshop y dataset en español de HatEval. También intentaron traducir los tweets a inglés para así usar un modelo entrenado con corpus en inglés pero no obtuvieron buenos resultados.

### 3.2.5. Resumen trabajos en idioma español

En la Tabla 3.6 se presenta información resumida de los 2 artículos que trabajaban con corpus en idioma español. Por otra parte, en la Tabla 3.7 muestra un resumen de los trabajos realizados en las distintas versiones del workshop.

| Nombre | Enfoque | Corpus | Algoritmo de clasificación | Vector de características | Mejor resultado |
|---|---|---|---|---|---|
| Automatic Cyberbullying Detection in Spanish-language Social Networks using Sentiment Analysis Techniques. (2018) (Mercado et al., 2018) | Machine Learning y Lexicón. | 100 frases de redes sociales, etiquetadas manualmente de 0 a 100. | Naive Bayes | Bolsa de Palabras | 93 % de accuracy |
| Presumptive Detection of Cyberbullying on Twitter through Natural Language Processing and Machine Learning in the Spanish. (2019) (Leon-Paredes et al., 2019) | Machine Learning | - Corpus propio, etiquetado de forma automática ("Presunto cyberbullying" o "Sin cyberbullying").<br>- 960.578 instancias. | Naive Bayes, Support Vector Machine y Logistic Regression | TF-IDF | Support Vector Machine con un peak de 94 % de accuracy. |

Tabla 3.6: Resumen artículos que usan corpus en idioma español

| Principales trabajos workshop por año | | | | | | |
|---|---|---|---|---|---|---|
| **Año** | **Nombre** | **Lugar competencia** | **Enfoque** | **Algoritmos de clasificación** | **Vector de características** | **Mejor resultado (F-measure)** |
| 2018 | INGEOTEC at MEX-A3T:Author profiling and aggressiveness analysis in Twitter using µTC and EvoMSA (Sánchez Gómez, 2018) | 1 | Combina 3 modelos:<br>- B4MS (SVM)<br>- Dos modelos Lexicones<br>- EvoDag (evaluación final) | Support Vector Machine | - | 0.620 |
| | Attention mechanism for aggressive detection (Cuza et al., 2018) | 2 | Red neuroanl (bi-LSTM y Post-Attention LST) | Redes neuronales | Word Embedding y presencia de malas palabras o no según un Lexicón. | 0.605 |
| 2019 | UACh at MEX-A3T 2019: Preliminary Results on Detecting Aggressive Tweets by Adding Author Information Via an Unsupervised Strategy (Aragón et al., 2019) | 1 | Machine learning | Multilayer Perceptron y Support Vector Vachine | Word Embedding y n-gramas | 0.620 (MP) |
| 2020 | Transformers and Data Augmentation for Aggressiveness Detection in Mexican Spanish (Tanase et al., 2020) | 1 y 2 | 1.- 20 BETOS ajustados con corpus de entrenamiento para predecir agresividad con esquemas de voto mayoritario y ponderados.<br>2.- 20 BETOS más técnicas de aumento de datos. | Redes neuronales | - | 1.- 0.8851<br>2.- 0.8588 |
| | Detecting Aggressiveness in Mexican Spanish Social Media Content by Fine-Tuning Transformer-Based Models (Guzman-Silverio et al., 2020) | 3 | Afinar modelos pre-entrenados en español, inglés y multilingües basados en Transformer. | Redes neuronales | - | 0.8538 (1 BETO) |

Tabla 3.7: Resumen trabajos workshop

# Capítulo 4

# Corpus utilizados

A continuación, se describen los corpus utilizados para los experimentos y entrenamiento de los diferentes modelos implementados.

## 4.1. Corpus Chileno

Está compuesto por la unión del corpus confeccionado en (Riquelme, 2019) con 1470 tweets etiquetados en *agresivo* o *noAgresivo* en el contexto de la agresividad hacia la mujer y el corpus confeccionado en el proyecto de título del autor de este trabajo que cuenta con 1000 tweets etiquetados en *agresivo* y *noAgresivo*. Todos los tweets corresponden a usuarios de nacionalidad chilena, por esta razón al corpus se le denomina Chileno. En la Figura 4.1 se puede observar el porcentaje de tweets etiquetados en *agresivo* y *noAgresivo* y en el Tabla 4.1 la cantidad de instancias que se destinaron para entrenamiento y prueba (70 % y 30 %, respectivamente).

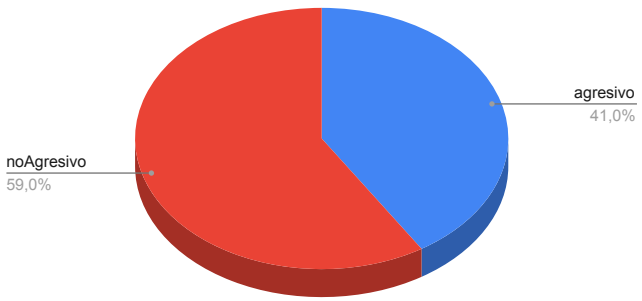


Figura 4.1: Instancias por clases

| | **Número de instancias** |
|---|---|
| Entrenamiento | 1729 |
| Prueba | 741 |
| Total | 2470 |

Tabla 4.1: Cantidad de instancias de entrenamiento y prueba corpus Chileno

## 4.2. Corpus Mexicano

Este corpus es creado en el workshop del año 2018 (Álvarez Carmona et al., 2018) para el desafío de detectar agresividad. Como se describe anteriormente el corpus corresponde a tweets en idioma español situados en un radio de 500Km de Ciudad de México etiquetados manualmente (agresivo y noAgresivo). Al solicitar este corpus se nos entrega 7332 instancias etiquetadas y 3143 no etiquetadas, dado esto se decide utilizar solo las 7332 instancias. En la Figura 4.2 se puede observar el porcentaje de tweets etiquetados en *agresivo* y *noAgresivo* y en la Tabla 4.2 la cantidad de instancias que se destinaron para entrenamiento y prueba (70 % y 30 %, respectivamente).

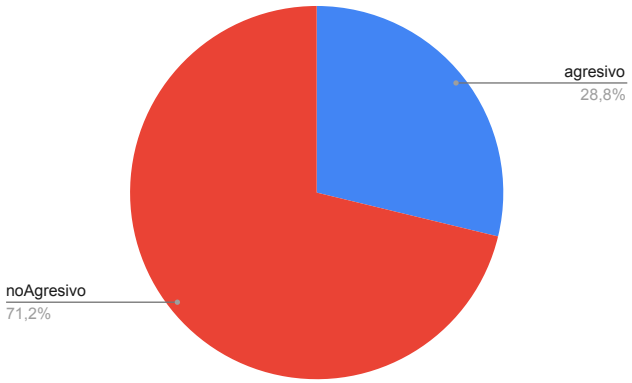


Figura 4.2: Instancias por clases corpus Mexicano

| | **Número de instancias** |
|---|---|
| Entrenamiento | 5132 |
| Prueba | 2200 |
| Total | 7332 |

Tabla 4.2: Cantidad de instancias entrenamiento y prueba corpus Mexicano

## 4.3. Corpus ChilenoMexicano

Este corpus corresponde a la unión de los dos anteriores, se hace con el objetivo de contar con un corpus de mayor tamaño y con tweets de distintos países para probar los diferentes modelos. En la Figura 4.2 se puede observar el porcentaje de tweets etiquetados en *agresivo* y *noAgresivo* y en el Tabla 4.2 la cantidad de instancias que se destinaron para entrenamiento y prueba (70 % y 30 %, respectivamente).

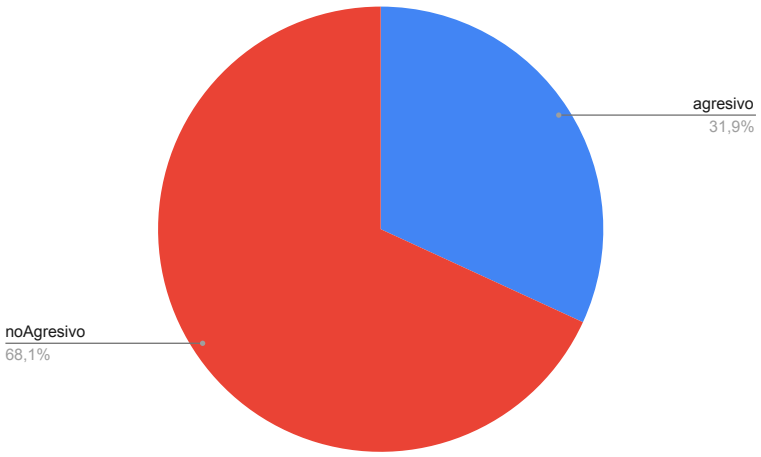


Figura 4.3: Instancias por clases corpus ChilenoMexicano

| | **Número de instancias** |
|---|---|
| Entrenamiento | 6861 |
| Prueba | 2941 |
| Total | 9802 |

Tabla 4.3: Cantidad de instancias entrenamiento y prueba corpus ChilenoMexicano

# Capítulo 5

# Descripción de los modelos

En este Capítulo, se detallan los enfoques creados para llevar a cabo la tarea de detectar agresividad en los textos. La principal característica que difiere entre los enfoques es la forma de representar el vector de características de los tweets que reciben como entrada los algoritmos de Machine Learning.

En cada enfoque se definen hiperparámetros y valores candidatos, los valores definitivos son elegidos mediante el algoritmo GridSearchCV[1] aplicados a los conjuntos de entrenamiento de cada corpus. GridSearchCV crea una matriz de ejecuciones donde se evalúan todas las posibles combinaciones de los valores candidatos y se retiene la mejor combinación. Para evaluar el rendimiento de cada ejecución se hace mediante la técnica de cross-validation[2] (con esto se evita el sobre ajuste) y la métrica seleccionada para elegir la mejor combinación es F-measure. La Figura 5.1 muestra este proceso y cómo se realiza la evaluación final de los modelos, los resultados se presentan en el Capítulo 6.

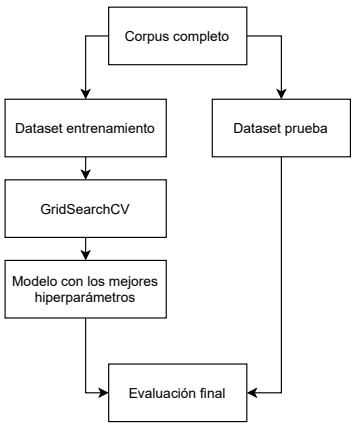


Figura 5.1: Proceso de entrenamiento y evaluación

[1] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[2] https://scikit-learn.org/stable/modules/cross_validation.html

## 5.1. Enfoque TF-IDF

Este primer enfoque es el más sencillo y usa la técnica más tradicional para obtener características de un texto, denominada TF-IDF. El propósito de crear este modelo es que se use como una base para comparar con los resultados de los demás modelos que mezclan Lexicones con clasificadores de Machine Learning.

En primer lugar, se hace un preprocesamiento del texto dependiendo de los valores definitivos de los hiperparámetros definidos en el Tabla 5.4. Para obtener el vector de características del texto se utiliza "Term frequency - Inverse document frequency" conocido como TF-IDF el cual consiste en determinar la importancia de cada palabra en la frase según la frecuencia de aparición de las palabras en el corpus. Para realizar este cálculo se utiliza la fórmula 5.1.

$$
\begin{gathered}
TFIDF(t,d) = TF(t,d) \times IDF(t) \\
\text{Donde:} \\
IDF(t) = \log\left(\frac{|D|}{|d \in D : t \in d|}\right) \\
\Downarrow \\
IDF(t) = \log\left(\frac{\text{N° total de documentos}}{\text{N° de documentos que contiene t}}\right)
\end{gathered}
\tag{5.1}
$$

Donde t es el término, en nuestro caso la palabra y d es el documento que para nosotros es la frase. TF es la frecuencia que aparece un término (palabra) en un documento (frase).

En la Tabla 5.2 se muestra la representación TF-IDF que toma cada frase que se encuentra en el corpus presentado en la Tabla 5.1, donde a cada palabra que se encuentra en la frase se le aplica la fórmula 5.1. Los vectores que se forman para cada frase sirven como entrada para los algoritmos de clasificación de Machine Learning.

| **Documentos** | **Frases** | **Términos (palabras)** |
|---|---|---|
| D1 | web web grafo | web, grafo |
| D2 | grafo web net grafo net | grafo, web, net |
| D3 | página web complejos | página, web, complejos |

Tabla 5.1: Pequeño corpus de ejemplo

Con este enfoque se crearon 3 modelos usando distintos clasificadores:

- TF-IDF_SVM: Utiliza como clasificador a Support Vector Machine.
- TF-IDF_NB: Utiliza como clasificador a Naive Bayes.
- TF-IDF_RF: Utiliza como clasificador a Random Forest.

| | **web** | **grafo** | **net** | **página** | **complejos** |
|---|---|---|---|---|---|
| IDF | log(3/3) | log(3/2) | log(3/1) | log(3/1) | log(3/1) |
| Representación D1 | 0 | 0.06 | 0 | 0 | 0 |
| Representación D2 | 0 | 0.07 | 0.19 | 0 | 0 |
| Representación D3 | 0 | 0 | 0 | 0.16 | 0.16 |

Tabla 5.2: Representación TF-IDF del pequeño corpus

En la Tabla 5.3 se presentan los valores definitivos de los hiperparámetros para los 3 modelos creados con el enfoque TF-IDF, luego de ejecutar GridSearchCV sobre las instancias de entrenamiento de los distintos corpus. Por ejemplo, para el modelo TF-IDF_SVM en el corpus chileno se encontró que para la tokenización se debe usar la libreria NLTK, filtrando StopWord y haciendo el proceso de stemming, el rango de los ngramas debe ser (1,1), eliminar palabras del diccionario que tengan frecuencia mayor a 1.0 y no transformar el vector a denso para obtener el mejor resultado. Esto nos muestra de forma detallada las características de los modelos para cada corpus, con estos hiperparámetros se ejecutan los experimentos sobre las instancias de prueba de los corpus.

| | | **Corpus** | | |
|---|---|---|---|---|
| **Modelo** | **Hiperparámetro** | **Chileno** | **Mexicano** | **ChilenoMexicano** |
| TF-IDF_SVM | tfidf_tokenizer | TokenizeNLTK _stopword _stemming | None | Tokenizer _con _stemming |
| | tfidf_ngram_range | (1,1) | (1,2) | (1,3) |
| | tfidf_max_df | 1.0 | 1.0 | 1.0 |
| | denso_activar | False | False | False |
| TF-IDF_NB | tfidf_tokenizer | Tokenizer _con _stemming | Tokenizer | Tokenizer |
| | tfidf_ngram_range | (1,2) | (1,3) | (1,3) |
| | tfidf_max_df | 0.8 | 1.0 | 1.0 |
| | denso_activar | True | True | True |
| TF-IDF_RF | tfidf_tokenizer | TokenizeNLTK _stopword _stemming | TokenizeNLTK _stopword _stemming | Tokenizer _stopword _stemming |
| | tfidf_ngram_range | (1,1) | (1,1) | (1,1) |
| | tfidf_max_df | 1.0 | 0.8 | 0.7 |
| | denso_activar | False | False | True |
| | clf_n_estimators | 150 | 150 | 150 |

Tabla 5.3: Valores hiperparámetros enfoque TF-IDF

| **Hiperparámetros enfoque TF-IDF** | | |
|---|---|---|
| **Nombre** | **Valores candidatos** | **Descripción** |
| tfidf_tokenizer | [<br>None,<br>Tokenizer,<br>Tokenizer_con_stopwords,<br>Tokenizer_con_lemmatization,<br>Tokenizer_con_stemming,<br>Tokenizer_stopword_stemming,<br>TokenizeNLTK_stopword_stemming<br>] | Función que se encarga de realizar la tokenización de la frases:<br>**None:** Se deja la tokenización por defecto que trae el algoritmo.<br>**Tokenizer:** Se filtran caracteres especiales de la frase y se usa spacy[3] en español para tokenizar.<br>**Tokenizer_con_stopwords:** Se filtran caracteres especiales y StopWord de la frase, se usa spacy en español para tokenizar.<br>**Tokenizer_con_lemmatization:** Se filtran caracteres especiales de la frase, se tokeniza y luego se lematiza cada palabra usando spacy en español.<br>**Tokenizer_con_stemming:** Se filtran caracteres especiales de la frase, se tokeniza usando spacy en español y luego se realiza el proceso de stemming con SnowballStemmer de NLTK[4] para cada palabra.<br>**Tokenizer_stopword_stemming:** Se filtran caracteres especiales y StopWord de la frase, se tokeniza usando spacy en español y luego se realiza el proceso de stemming con PorterStemmer de NLTK para cada palabra.<br>**TokenizeNLTK_stopword_stemming:** Se filtran caracteres especiales y StopWord de la frase, se tokeniza usando NLTK en español y luego se realiza el proceso de stemming con PorterStemmer de NLTK para cada palabra. |
| tfidf_ngram_range | [(1,1),(1,2),(1,3)] | El límite inferior y superior del rango de valores de n para los diferentes n-gramas a extraer. Por ejemplo, un rango de ngramas de (1, 1) significa sólo unigramas, (1, 2) significa unigramas y bigramas. |
| tfidf_max_df | [1.0, 0.9, 0.8, 0.7] | Al construir el vocabulario se ignoran los términos que tienen una frecuencia de documentos estrictamente superior al umbral dado. Palabras que se repiten mucho no son incluidas en el vocabulario. |
| denso_activar | [True, False] | Tranformar el vector en denso con el método toarray de numpy. Cuando se usa el clasificador Naive Bayes se debe activar de forma obligatoria. |
| clf_n_estimators | [200, 150, 100, 80] | Número de árboles de decisión que usa el Random Forest. Esto aplica solo cuando se usa el clasificador Random Forest. |

Tabla 5.4: Hiperparámetros enfoque TF-IDF

[3]Librería de PLN para Python disponible en https://spacy.io/models/es
[4]Librería de PLN para Python disponible en https://www.nltk.org/

## 5.2. Enfoque Lexicón

El segundo enfoque implementado utiliza una mezcla del análisis de emociones mediante Lexicones para formar el vector de características y clasificadores de Machine Learning.

El Lexicón utilizado es el propuesto en (Segura Navarrete et al., 2021) el cual consiste en un léxico afectivo en español basado en un Lexicón enriquecido, el cual represente la intensidad de la emoción de cada palabra, como se muestra en la Tabla 3.1 del Capítulo 3 sección 3.1.2. Este Lexicón considera solo palabras denominadas emotivas.

En primer lugar, se hace un preprocesamiento del texto filtrando caracteres especiales (tildes, puntuación, signo, etc), además de eliminar StopWord y lematización de cada palabra dependiendo de los valores definitivos de los hiperparámetros definidos en el Tabla 5.6. La tokenización se realiza utilizando spacy en español. Posteriormente al preprocesamiento, se realiza el análisis con los Lexicones para formar el vector de características de cada frase que se componen por 10 columnas, detalladas a continuación:

- Resultado de la suma de las intensidades de las palabras de la frase, que aparecen en el Lexicón que representa la clase afectiva enojo.
- Resultado de la suma de las intensidades de las palabras de la frase, que aparecen en el Lexicón que representa la clase afectiva anticipación.
- Resultado de la suma de las intensidades de las palabras de la frase, que aparecen en el Lexicón que representa la clase afectiva disgusto.
- Resultado de la suma de las intensidades de las palabras de la frase, que aparecen en el Lexicón que representa la clase afectiva miedo.
- Resultado de la suma de las intensidades de las palabras de la frase, que aparecen en el Lexicón que representa la clase afectiva alegría.
- Resultado de la suma de las intensidades de las palabras de la frase, que aparecen en el Lexicón que representa la clase afectiva tristeza.
- Resultado de la suma de las intensidades de las palabras de la frase, que aparecen en el Lexicón que representa la clase afectiva sorpresa.
- Resultado de la suma de las intensidades de las palabras de la frase, que aparecen en el Lexicón que representa la clase afectiva confianza.
- Resultado de la división entre el número de malas palabras (MP) encontrada en la frase y la cantidad de palabras de la frase.
- Cantidad de palabras en la frase (CP).

La Tabla 5.5 muestra un ejemplo del vector de características que se obtiene de la frase “Oyyyyy feo culiao insoportable chucha nota esta cagao miedo ” para ejemplificar el proceso. La columna de enojo tiene un valor de 56, ya que en el Lexicón de la clase afectiva

se encuentra la palabra “nota” que tiene una intensidad de 10 y la palabra “miedo” con intensidad de 46, por lo tanto, al sumar estas dos intensidades se obtiene 56. Para las otras columnas de las clases afectiva el proceso que se realiza es el mismo. En la columna que representa el resultado de la división entre la cantidad de malas palabras y la cantidad de palabras en la frase el valor es de 0,333, ya que las malas palabras encontradas en el Lexicón definido son 3; “feo”, “culiao”, “chucha” y la cantidad de palabras que se encuentran en la frase es de 9.

| **Frase filtrada** | **Enojo** | **Anticipación** | **Disgusto** | **Miedo** | **Alegría** | **Tristeza** | **Sorpresa** | **Confianza** | **MP/CP** | **CP** |
|---|---|---|---|---|---|---|---|---|---|---|
| Oyyyyy feo culiao insoportable chucha nota esta cagao miedo | 56 | 64 | 85 | 64 | 92 | 38 | 68 | 46 | 0.333 | 9 |

Tabla 5.5: Representación vector Lexicón para una frase

Con este enfoque se crearon 3 modelos usando distintos clasificadores:

- Lexicón_SVM: Utiliza como clasificador a Support Vector Machine.
- Lexicón_NB: Utiliza como clasificador a Naive Bayes.
- Lexicón_RF: Utiliza como clasificador a Random Forest.

| **Hiperparámetros enfoque Lexicón** | | |
|---|---|---|
| **Nombre** | **Valores Candidatos** | **Descripción** |
| vl_lema | [True, False] | Activa o desactiva la lematización de las plabras, se realiza utilizando spacy.<br>True: Activa<br>False: Desactiva |
| vl_stopword | [True, False] | Activa o desactiva el filtrado de stopword<br>True: Activa<br>False: Desactiva |
| clf_kernel | [linear, sigmoid] | Se selecciona el Kernel de Support Vector Machine.<br>Esto aplica solo cuando se usa el clasificador Support Vector Machine. |
| clf_n_estimators | [200, 150, 100 , 80] | Número de árboles de decisión que usa el Random Forest.<br>Esto aplica solo cuando se usa el clasificador Random Forest. |

Tabla 5.6: Hiperparámetros enfoque Lexicón

En la Tabla 5.7 se presentan los valores definitivos de los hiperparámetros para los 3 modelos creados con el enfoque Lexicón, luego de ejecutar GridSearchCV sobre las instancias de entrenamiento de los distintos corpus. Esto nos muestra, de forma detalla, las características de los modelos para cada corpus.

| | | **Corpus** | | |
|---|---|---|---|---|
| **Modelo** | **Hiperparámetro** | **Chileno** | **Mexicano** | **ChilenoMexicano** |
| Lexicón_SVM | vl_lema | True | True | True |
| | vl_stopword | True | True | True |
| | clf_kernel | linear | linear | linear |
| Lexicón_NB | vl_lema | True | True | True |
| | vl_stopword | False | False | True |
| Lexicón_RF | vl_lema | False | False | True |
| | vl_stopword | True | True | True |
| | clf_n_estimators | 150 | 150 | 80 |

Tabla 5.7: Valores hiperparámetros enfoque Lexicón

## 5.3. Enfoque TF-IDF_Lexicón

Para este enfoque se implementa una mezcla del enfoque TF-IDF y Lexicón, es decir el vector de características es una concatenación del vector TF-IDF y el vector derivado del análisis de Lexicones. En la Figura 5.2 se muestra el proceso que lleva a cabo este enfoque, como se puede observar al principio el corpus toma dos caminos para realizar cada enfoque, el preprocesamiento del corpus se realiza en cada enfoque según sus hiperparámetros definidos anteriormente. Finalmente, se concatenan estos dos vectores como se muestra en la Figura 5.3 para aplicar los algoritmos de Machine Learning. El tamaño del vector depende del corpus y sus palabras, esto ocurre al tener la representación TF-IDF.

Con este enfoque se crearon 3 modelos usando distintos clasificadores:

- TF-IDF_Lexicón_SVM: Utiliza como clasificador a Support Vector Machine.
- TF-IDF_Lexicón_NB: Utiliza como clasificador a Naive Bayes.
- TF-IDF_Lexicón_RF: Utiliza como clasificador a Random Forest.

En la Tabla 5.8 se muestran los hiperparámetros definidos para este enfoque, como se puede observar se toman los hiperparámetros de los dos enfoques involucrados (TF-IDF 5.4 y Lexicón 5.6) además de los definidos para los clasificadores.

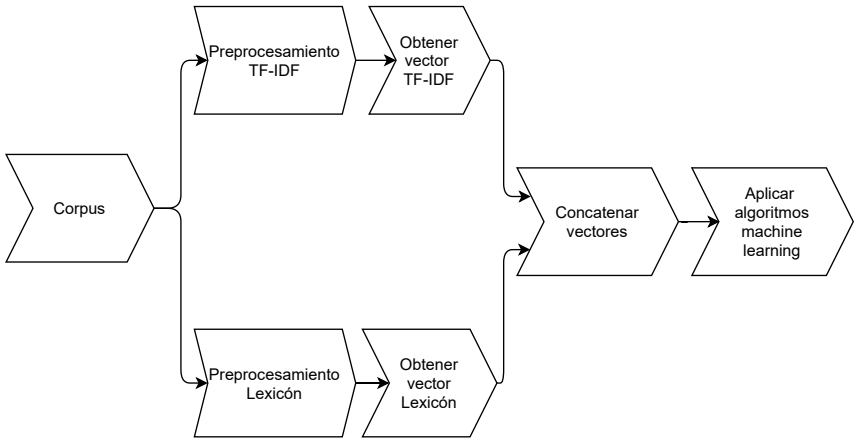


Figura 5.2: Enfoque TF-IDF_Lexicón

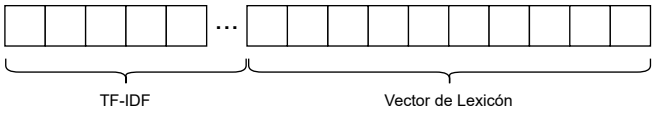


Figura 5.3: Vector de características modelo TF-IDF_Lexicón

| **Hiperparámetros enfoque TF-IDF_Lexicón** | | |
|---|---|---|
| **Nombre** | **Valores Candidatos** | **Descripción** |
| Hiperparámetros enfoque TF-IDF | | |
| Hiperparámetros enfoque Lexicón | | |
| clf_kernel | [linear, sigmoid] | Se selecciona el Kernel de Support Vector Machine.<br>Esto aplica solo cuando se usa el clasificador Support Vector Machine. |
| clf_n_estimators | [200, 150, 100, 80] | Número de árboles de decisión que usa el Random Forest.<br>Esto aplica solo cuando se usa el clasificador Random Forest. |

Tabla 5.8: Hiperparámetros enfoque TF-IDF_Lexicón

En la Tabla 5.9 se presentan los valores definitivos de los hiperparámetros para los 3 modelos creados con el enfoque TF-IDF_Lexicón, luego de ejecutar GridSearchCV sobre las instancias de entrenamiento de los distintos corpus. Esto nos muestra, de forma detallada, las características de los modelos para cada corpus, con estos hiperparámetros se ejecutan los experimentos sobre las instancias de prueba de los corpus.

| | | **Corpus** | | |
|---|---|---|---|---|
| **Modelo** | **Hiperparámetro** | **Chileno** | **Mexicano** | **ChilenoMexicano** |
| TF-IDF_Lexicón_SVM | tfidf_tokenizer | TokenizeNLTK _stopword _stemming | None | Tokenizer_con _stemming |
| | tfidf_ngram_range | (1,2) | (1,2) | (1,2) |
| | tfidf_max_df | 1.0 | 1.0 | 1.3 |
| | denso_activar | False | False | False |
| | vl_lema | True | True | True |
| | vl_stopword | False | False | True |
| | clf_kernel | linear | linear | linear |
| TF-IDF_Lexicón_NB | tfidf_tokenizer | Tokenizer_con _stemming | Tokenizer | Tokenizer |
| | tfidf_ngram_range | (1,2) | (1,3) | (1,3) |
| | tfidf_max_df | 0.8 | 1.0 | 1.0 |
| | denso_activar | True | True | True |
| | vl_lema | False | False | True |
| | vl_stopword | False | True | True |
| TF-IDF_Lexicón_RF | tfidf_tokenizer | TokenizeNLTK _stopword _stemming | TokenizeNLTK _stopword _stemming | TokenizeNLTK _stopword _stemming |
| | tfidf_ngram_range | (1,1) | (1,1) | (1,1) |
| | tfidf_max_df | 1.0 | 0.7 | 0.7 |
| | denso_activar | False | True | True |
| | vl_lema | False | False | True |
| | vl_stopword | False | True | True |
| | clf_n_estimators | 80 | 200 | 150 |

Tabla 5.9: Valores hiperparámetros enfoque TF-IDF_Lexicón

## 5.4. Enfoque Word Embedding

En este enfoque se busca representar el vector de características mediante la técnica de Word Embedding, al igual que el enfoque TF-IDF este se implementa para tener una base de comparación para los demás enfoques que incluyen Lexicones.

Word Embedding es un enfoque de la semántica de distribución que representa palabras como vectores de número reales. Dicha representación tiene propiedades de agrupamiento útiles, ya que agrupa palabras que son semántica y sintácticamente similares. Por ejemplo, esperamos que las palabras "delfín y foca" se encuentren cerca, pero "Paris" y "delfín" no se encuentren cerca, ya que no existe una fuerte relación entre ellas. Por lo tanto, las palabras se representan como vectores de valores reales, donde cada valor captura una dimensión del significado de la palabra. Esto provoca que palabras semánticamente similares, tengan vectores similares. De forma simplificada, cada dimensión de los vectores representa un significado y el valor numérico en cada dimensión captura la cercanía de la asociación de la palabra a dicho significado.

En primer lugar, al igual que los enfoques anterior se realiza un preprocesamiento del texto filtrando caracteres especiales (tildes, puntuación, signo, etc), además de eliminar StopWord y realizar lematización de cada palabra dependiendo de los valores definitivos de los hiperparámetros definidos en el Tabla 5.10. Luego, para representar el vector de característica de cada texto se realiza mediante la suma de los vectores de word embedding de cada palabra presente en la frase. De esta forma se obtendrá un vector que representa todo el texto, cabe señalar que luego de la suma se hace una normalización del vector resultante. En la Figura 5.4 se muestra, a modo de ejemplo, una representación vectorial de la frase "me gustan los gatos" (sin normalizar).

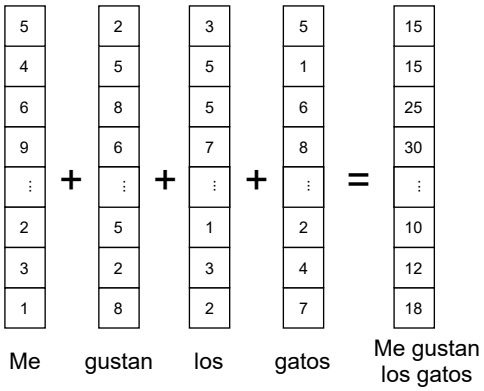


Figura 5.4: Vector de características enfoque Word Embedding

Se utiliza un modelo Word Embedding[5] pre-entrenado, el cual se implementó con FastText y Skipgram (Bojanowski et al., 2017) y se entrenó en 20 épocas y 1.4 billones de palabras utilizando el corpus Spanish Billion Word Corpus (Cardellino, 2019). Cada vector tiene 300 dimensiones, por lo tanto, cada texto será representado con un vector de

[5] dcc.uchile.cl/~jperez/word-embeddings/fasttext-sbwc.vec.gz

tamaño 300. Este vector es recibido como entrada por los algoritmos de clasificación que se implementan en los 3 modelos creados con este enfoque:

- WordEmbedding_SVM: Utiliza como clasificador a Support Vector Machine.
- WordEmbedding_NB: Utiliza como clasificador a Naive Bayes.
- WordEmbedding_RF: Utiliza como clasificador a Random Forest.

| **Hiperparámetros enfoque Word Embedding** | | |
|---|---|---|
| **Nombre** | **Valores Candidatos** | **Descripción** |
| vwe_lema | [True, False] | Activa o desactiva la lematización de las plabras, se realiza utilizando spacy.<br>True: Activa<br>False: Desactiva |
| vwe_stopword | [True, False] | Activa o desactiva el filtrado de stopword<br>True: Activa<br>False: Desactiva |
| clf_n_estimators | [200, 150, 100 , 80] | Número de árboles de decisión que usa el Random Forest.<br>Esto aplica solo cuando se usa el clasificador Random Forest. |

Tabla 5.10: Hiperparámetros enfoque Word Embedding

En la Tabla 5.11 se presentan los valores definitivos de los hiperparámetros para los 3 modelos creados con el enfoque Word Embedding, luego de ejecutar GridSearchCV sobre las instancias de entrenamiento de los distintos corpus. Esto nos muestra, de forma detallada, las características de los modelos para cada corpus.

| | | **Corpus** | | |
|---|---|---|---|---|
| **Modelo** | **Hiperparámetro** | **Chileno** | **Mexicano** | **ChilenoMexicano** |
| WordEmbedding_SVM | vl_lema | False | False | False |
| | vl_stopword | True | True | False |
| WordEmbedding_NB | vl_lema | False | False | False |
| | vl_stopword | True | True | False |
| WordEmbedding_RF | vl_lema | True | False | False |
| | vl_stopword | True | False | False |
| | clf_n_estimators | 200 | 150 | 150 |

Tabla 5.11: Valores hiperparámetros enfoque Word Embedding

## 5.5. Enfoque WE_Lexicón

Este enfoque representa el vector de características como una concatenación de los vectores de salida de los enfoques Word Embedding y Lexicón. En la Figura 5.5 se muestra el proceso que lleva a cabo este enfoque, como se puede observar al principio el corpus toma dos caminos para realizar cada enfoque, el preprocesamiento del corpus se hace en cada enfoque según sus hiperparámetros definidos anteriormente. Finalmente, se concatenan estos dos vectores como se muestra en la Figura 5.6 para aplicar los algoritmos de Machine Learning. El tamaño del vector es de 310, 300 casillas corresponden al vector Word Embedding y 10 al análisis de Lexicones.

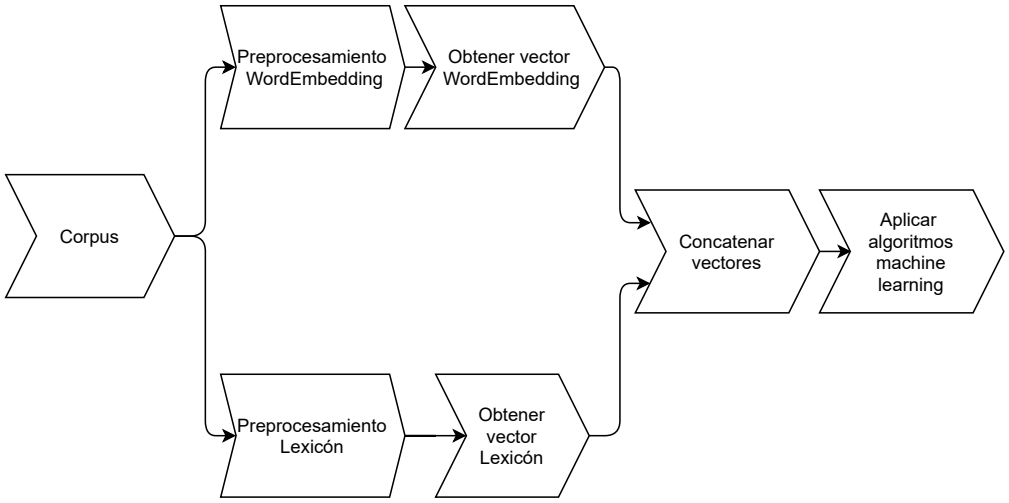


Figura 5.5: Enfoque WordEmbedding_Lexicón

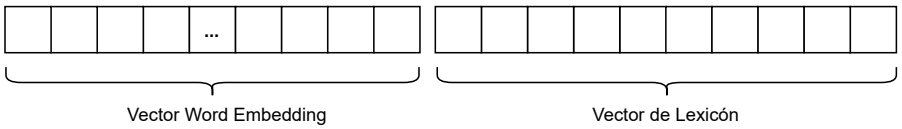


Figura 5.6: Vector de características enfoque WordEmbedding_Lexicón

Con este enfoque se implementan 3 modelos utilizando diferentes clasificadores:

- WE_Lexicón_SVM: Utiliza como clasificador a Support Vector Machine.
- WE_Lexicón_NB: Utiliza como clasificador a Naive Bayes.
- WE_Lexicón_RF: Utiliza como clasificador a Random Forest.

Los hiperparámetros definidos en este enfoque son los propios de los dos enfoques involucrados (Word Embedding 5.10 y Lexicón 5.6), además del hiperparámetro que ha-

| **Hiperparámetros enfoque WordEmbedding_Lexicón** | | |
|---|---|---|
| **Nombre** | **Valores Candidatos** | **Descripción** |
| Hiperparámetros enfoque Word Embedding | | |
| Hiperparámetros enfoque Lexicon | | |
| clf_n_estimators | [200, 150, 100, 80] | Número de árboles de decisión que usa el Random Forest. Esto aplica solo cuando se usa el clasificador Random Forest. |

Tabla 5.12: Hiperparámetros enfoque WordEmbedding_Lexicón

ce referencia al número de árboles de decisión que usa Random Forest en el modelo WE_Lexicón_TF-IDF_RF, como se muestra en la Tabla 5.12.

Los valores definitivos para los hiperparámetros de cada modelo encontrados por GridSearchCv sobre las instancias de entrenamiento de los corpus se presentan en la Tabla 5.13.

| | | **Corpus** | | |
|---|---|---|---|---|
| **Modelo** | **Hiperparámetro** | **Chileno** | **Mexicano** | **ChilenoMexicano** |
| WE_Lexicón_SVM | vl_lema | False | False | False |
| | vl_stopword | True | True | True |
| | vwe_lema | False | False | False |
| | vwe_stopword | True | True | False |
| WE_Lexicón_NB | vl_lema | True | True | False |
| | vl_stopword | False | False | False |
| | vwe_lema | False | False | False |
| | vwe_stopword | True | True | False |
| WE_Lexicón_RF | vl_lema | False | False | True |
| | vl_stopword | False | False | False |
| | vwe_lema | True | True | False |
| | vwe_stopword | True | True | False |
| | clf_n_estimators | 100 | 100 | 150 |

Tabla 5.13: Valores hiperparámetros enfoque WordEmbedding_Lexicón

## 5.6. Enfoque WE_Lexicón_TF-IDF

Al igual que en el enfoque anterior se representa el vector de características con la concatenación del vector Word Embedding y Lexicón, pero además se agrega el vector TF-IDF. En la Figura 5.7 se muestra el proceso que lleva a cabo este enfoque, se puede observar que, a diferencia del enfoque anterior, ahora el corpus toma 3 caminos para ejecutar los 3 enfoques con su preprocesamiento según los hiperparámetros definidos anteriormente. Finalmente, se concatenan los 3 vectores como se muestra en la Figura 5.8 para, posteriormente, aplicar el clasificador de Machine Learning, el tamaño del vector depende del corpus y sus palabras, esto ocurre al tener la representación TF-IDF.

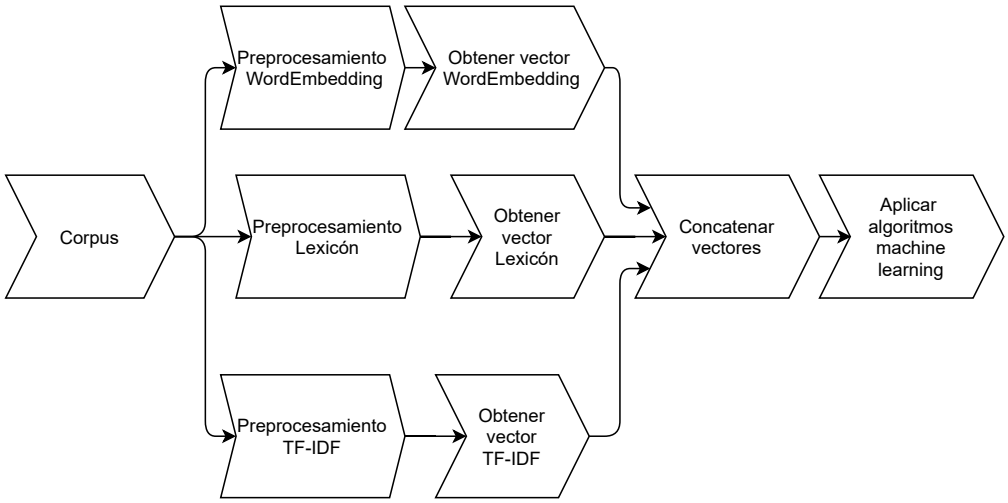


Figura 5.7: Enfoque WE_Lexicón_TF-IDF

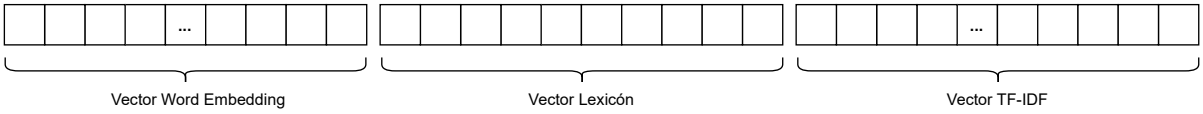


Figura 5.8: Vector de características enfoque WordEmbedding_Lexicón_TF-IDF

Con este enfoque se crearon 3 modelos usando distintos clasificadores:

- WE_Lexicón_TF-IDF_SVM: Utiliza como clasificador a Support Vector Machine.
- WE_Lexicón_TF-IDF_NB: Utiliza como clasificador a Naive Bayes.
- WE_Lexicón_TF-IDF_RF: Utiliza como clasificador a Random Forest.

Los hiperparámetros definidos en este enfoque son los propios de los tres enfoques involucrados (Word Embedding 5.10, Lexicón 5.6 y TF-IDF 5.4), además del hiperparámetro

que hace referencia al número de árboles de decisión que usa Random Forest en el modelo WE_Lexicón_TF-IDF_RF, como se muestra en la Tabla 5.14.

En la Tabla 5.15 se presentan los valores definitivos de los hiperparámetros para los 3 modelos creados con el enfoque WE_Lexicón_TF-IDF, luego de ejecutar GridSearchCV sobre las instancias de entrenamiento de los distintos corpus.

| **Hiperparámetros enfoque WE_Lexicón_TF-IDF** | | |
|---|---|---|
| **Nombre** | **Valores Candidatos** | **Descripción** |
| Hiperparámetros enfoque Word Embedding | | |
| Hiperparámetros enfoque Lexicón | | |
| Hiperparámetros enfoque TF-IDF | | |
| clf_n_estimators | [200, 150, 100, 80] | Número de árboles de decisión que usa el Random Forest.<br>Esto aplica solo cuando se usa el clasificador Random Forest. |

Tabla 5.14: Hiperparámetros enfoque WE_Lexicón_TF-IDF

| | | Corpus | | |
|---|---|---|---|---|
| **Modelo** | **Hiperparámetro** | **Chileno** | **Mexicano** | **ChilenoMexicano** |
| WE_Lexicón_TF-IDF_SVM | vwe_lema | False | True | True |
| | vwe_stopword | False | False | False |
| | vl_lema | True | True | True |
| | vl_stopword | True | True | True |
| | tfidf_tokenizer | Tokenizer_con _stemming | None | Tokenizer_con _stemming |
| | tfidf_ngram_range | (1,1) | (1,2) | (1,3) |
| | tfidf_max_df | 0.8 | 1.0 | 1.0 |
| WE_Lexicón_TF-IDF_NB | vwe_lema | False | True | True |
| | vwe_stopword | True | True | True |
| | vl_lema | False | False | False |
| | vl_stopword | False | True | True |
| | tfidf_tokenizer | Tokenizer_con _stemming | Tolenizer | Tokenizer |
| | tfidf_ngram_range | (1,2) | (1,3) | (1,3) |
| | tfidf_max_df | 0.8 | 1.0 | 1.0 |
| | denso_activar | True | True | True |
| WE_Lexicón_TF-IDF_RF | vwe_lema | True | False | False |
| | vwe_stopword | True | False | False |
| | vl_lema | True | False | False |
| | vl_stopword | False | True | True |
| | tfidf_tokenizer | Tokenizer_con _stemming | Tokenizer_con _stemming | Tokenizer_con _stemming |
| | tfidf_ngram_range | (1,1) | (1,1) | (1,1) |
| | tfidf_max_df | 0.7 | 0.7 | 0.7 |
| | clf_n_estimators | 100 | 100 | 150 |

Tabla 5.15: Valores hiperparámetros enfoque WE_Lexicón_TF-IDF

## 5.7. Enfoque Ensemble

A continuación, se describen 4 modelos implementados bajo la técnica de "Ensemble Learning". Ensemble learning es el proceso de combinar las decisiones de varios modelos de Machine Learning entrenados para mejorar el rendimiento general. Con las decisiones de los distintos modelos se da lugar a una predicción final usando diferentes reglas como, por ejemplo, el voto mayoritario. La motivación para utilizar modelos de Ensemble es reducir el error de generalización de la predicción. Siempre que los modelos combinados sean diversos e independientes, el error de predicción del modelo disminuye cuando se utiliza esta técnica. El enfoque busca la sabiduría de las multitudes para hacer una predicción. Aunque el modelo Ensemble tiene múltiples modelos base dentro del modelo, actúa y se comporta como un único modelo (Kotu y Deshpande, 2015).

En los modelos desarrollados, la predicción final se realiza mediante una votación mayoritaria, esto se implementa con VotingClassifier[6] de la librería scikit-learn.

### 5.7.1. Modelo TF-IDF_Lexicón_E_Clfs

El primer modelo creado bajo este enfoque combina los tres modelos implementados bajo el enfoque TF-IDF_Lexicón como muestra la Figura 5.9, se puede observar que el corpus alimenta a los 3 modelos que se entrenan de forma individual para luego hacer una predicción final sobre el corpus de prueba usando la técnica del voto mayoritario. Este modelo se crea bajo la hipótesis que la combinación de los 3 modelos que usan el enfoque TF-IDF_Lexicón darán mejores resultados que cada uno de ellos, ya que usan distintos clasificadores. Se usan los valores definitivos de los hiperparámetros de cada modelo encontrados anteriormente usando GridSearchCv sobre los distintos dataset de entrenamiento de los corpus (Tabla 5.9).

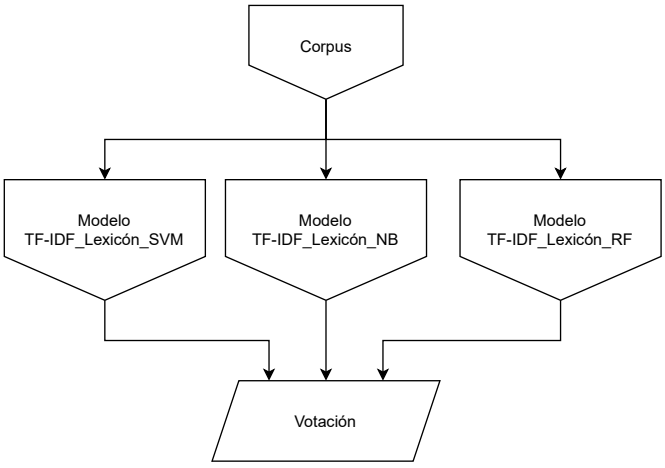


Figura 5.9: Modelo TF-IDF_Lexicón_E_Clfs

[6] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html

### 5.7.2. Modelo TF-IDF_Lexicón_E_SVM

Este modelo combina los modelos creados con el clasificador Support Vector Machine en los enfoques TF-IDF, Lexicón y TF-IDF_Lexicón como se muestra en la Figura 5.10. Al igual que el enfoque anterior, los modelos se entrenan de forma separada para luego hacer una predicción final sobre el corpus de prueba usando la técnica de voto mayoritario. Se implementa, ya que se cree que al combinar las distintas forma de obtener el vector de características puede mejorar el resultado de la clasificación final. Se utiliza el clasificador Support Vector Machine porque en las pruebas preliminares obtiene el mejor rendimiento.

Se usan los distintos valores de los hiperparámetros de cada modelo encontrados anteriormente usando GridSearchCv sobre los distintos dataset de entrenamiento de los corpus (TF-IDF_SVM 5.3, Lexicón_SVM 5.7 y TF-IDF_Lexicón_SVM 5.9).

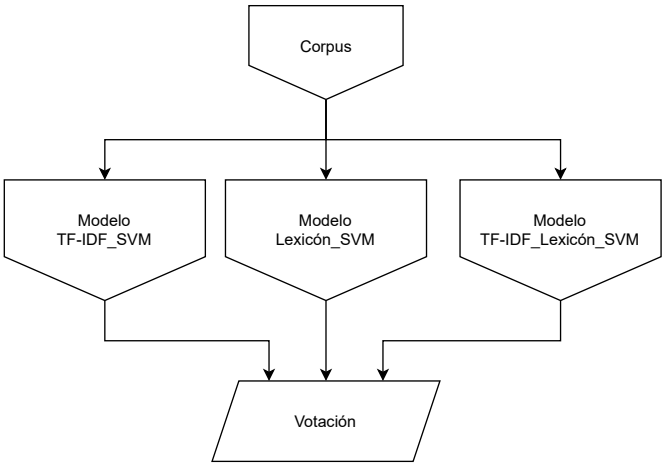


Figura 5.10: Modelo TF-IDF_Lexicón_E_SVM

### 5.7.3. Modelo WE_Lexicón_TF-IDF_E_SVM

El tercer modelo creado combina los modelos implementados con el clasificador Support Vector Machine en los enfoques Word Embedding, WE_Lexicón y WE_Lexicón_TF-IDF como se muestra en la Figura 5.11. Se entrenan los modelos de forma separada para luego hacer una predicción final sobre el corpus de prueba usando la técnica de voto mayoritario. Al igual que el enfoque anterior, este modelo se implementa bajo la hipotésis que al combinar las distintas formas de obtener el vector de característica puede mejorar el resultado de la clasificación final.

Los valores de los hiperparámetros que se usan en cada modelo fueron encontrados anteriormente usando GridSearchCv sobre los distintos dataset de entrenamiento de los corpus (WordEmbedding_SVM 5.11, WE_Lexicón_SVM 5.13 y WE_Lexicón_TF-IDF_SVM 5.15)

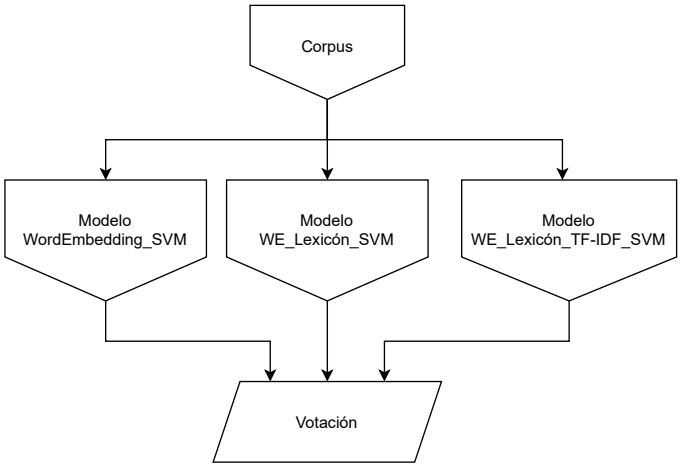


Figura 5.11: Modelo WE_Lexicón_TFIDF_E_SVM

### 5.7.4. Modelo Enfoques_E_SVM

El último modelo Ensemble que se implementa combina todos los modelos implementados con el clasificador Support Vector Machine en los distintos enfoques, como se observa en la Figura 5.12. Al igual que todos los modelos anteriores, se entrenan de forma separada para luego hacer una predicción final sobre el corpus de prueba usando la técnica de voto mayoritario. La hipótesis para implementarlo es que una mayor diversificación de las formas de obtener el vector de características puede mejorar el resultado. Cabe mencionar que este modelo es el más costoso en términos de memoria y tiempo para entrenar y probar corpus. Se usan los valores de los hiperparámetros de los modelos encontrados anteriormente (TF-IDF_SVM 5.3, Lexicón_SVM 5.7, TF-IDF_Lexicón_SVM 5.9, WordEmbedding_SVM 5.11, WE_Lexicón_SVM 5.13 y WE_Lexicón_TF-IDF_SVM 5.15).

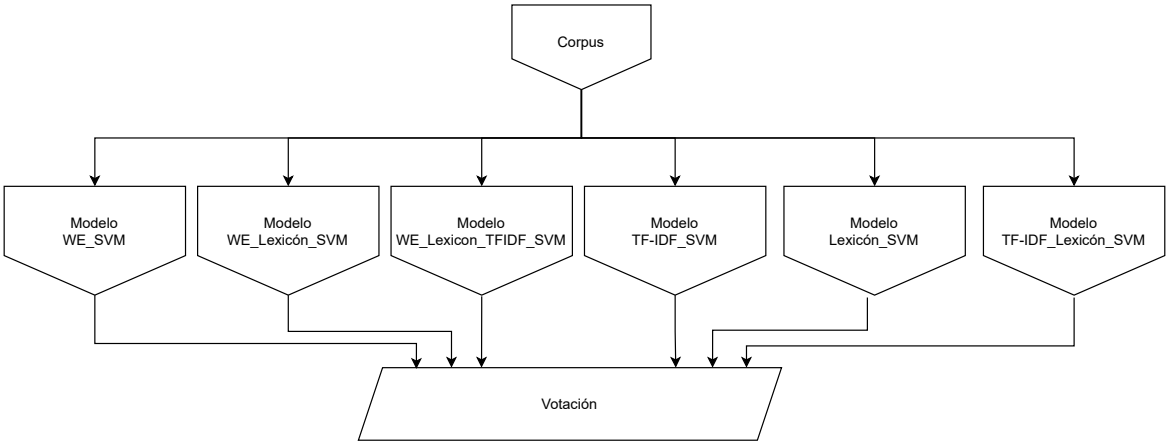


Figura 5.12: Modelo Enfoques_E_SVM

## 5.8. Resumen modelos implementados

A continuación, la Tabla 5.16 resume todos los modelos implementados, agrupados por enfoque. Todos estos modelos son entrenados y probados en los 3 corpus definidos anteriormente.

| **Enfoque** | **Nombre modelo** |
|---|---|
| TF-IDF | TF-IDF_SVM |
| | TF-IDF_NB |
| | TF-IDF_RF |
| Lexicón | Lexicón_SVM |
| | Lexicón_NB |
| | Lexicón_RF |
| TF-IDF_Lexicón | TF-IDF_Lexicón_SVM |
| | TF-IDF_Lexicón_NB |
| | TF-IDF_Lexicón_RF |
| WordEmbedding | WordEmbedding_SVM |
| | WordEmbedding_NB |
| | WordEmbedding_RF |
| WE_Lexicón | WE_Lexicón_SVM |
| | WE_Lexicón_NB |
| | WE_Lexicón_RF |
| WE_Lexicón_TF-IDF | WE_Lexicón_TF-IDF_SVM |
| | WE_Lexicón_TF-IDF_NB |
| | WE_Lexicón_TF-IDF_RF |
| Ensemble | TF-IDF_Lexicón_E_Clfs |
| | TF-IDF_Lexicón_E_SVM |
| | WE_Lexicón_TF-IDF_E_SVM |
| | Enfoques_E_SVM |

Tabla 5.16: Resumen modelos creados

# Capítulo 6

# Experimentación

En este capítulo se presentan los experimentos realizados para medir y comparar el rendimiento de los distintos modelos creados en los diferentes corpus descritos en el capítulo 4. En primer lugar, se definen los experimentos, luego se presentan los resultados de los diferentes modelos agrupados por enfoque y, finalmente, se comparan dichos resultados.

## 6.1. Definición de los experimentos

Los modelos implementados son probados en los dataset de prueba de los tres corpus descritos en el Capítulo 4, cabe destacar que estos dataset no fueron utilizados en el proceso de entrenamiento como muestra la Figura 5.1, de esta forma se puede medir de buena manera la capacidad de generalización de los modelos utilizando las métricas descritas en el Capítulo 3 sección 3.1.7.

Los hiperparámetros utilizados en cada modelo fueron encontrados mediante la técnica GridSearchCv en los diferentes dataset de entrenamiento de los corpus, como se describe en el Capítulo 5.

Los experimentos fueron llevados a cabo en el servidor del Magíster en Ciencias de la Computación de la Universidad del Bío-Bío, que cuenta con la siguientes características de hardware y software:

- Ubuntu 18.04.03 (4.15.0.101-generic kernel)
- 2 x Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz
- 62 Gb RAM
- 48 CPUs en total

## 6.2. Resultados modelos enfoque TF-IDF

La tabla 6.1 muestra los resultados de las métricas para los 3 modelos creados bajo el enfoque TF-IDF. Como se observa, el modelo que mejor resultados obtiene en el corpus

de prueba Chileno es TF-IDF_RF con una F-measure de 0.8701, para el Mexicano y ChilenoMexicano es TF-IDF_SVM con una F-measure de 0.8225 y 0.8424, respectivamente. El modelo que utiliza el clasificador Naive Bayes obtiene los peores resultados. Estos resultados serán usados como punto de comparación para los demás modelos que utilizan la mezcla híbrida de Lexicones y Machine Learning.

| **Modelo** | **Corpus** | **F-measure** | **Accuracy** | **Precision** | **Recall** |
|---|---|---|---|---|---|
| TF-IDF_SVM | Chileno | 0.8671 | 0.8690 | 0.8715 | 0.8690 |
| | Mexicano | **0.8225** | 0.8281 | 0.8234 | 0.8281 |
| | ChilenoMexicano | **0.8424** | 0.8473 | 0.8459 | 0.8473 |
| TF-IDF_NB | Chileno | 0.7514 | 0.7530 | 0.7511 | 0.7530 |
| | Mexicano | 0.7336 | 0.7522 | 0.7372 | 0.7522 |
| | ChilenoMexicano | 0.7351 | 0.75 | 0.7397 | 0.75 |
| TF-IDF_RF | Chileno | **0.8701** | 0.8717 | 0.8736 | 0.8717 |
| | Mexicano | 0.8069 | 0.8204 | 0.8203 | 0.8204 |
| | ChilenoMexicano | 0.833 | 0.8418 | 0.8450 | 0.8418 |

Tabla 6.1: Resultados métricas para los modelos TF-IDF

Se puede determinar que el modelo que obtuvo un mejor rendimiento en este enfoque es TF-IDF_SVM promediando la F-measure obtenida en los 3 corpus, como se observa en la tabla 6.2

| **Modelo** | **Promedio F-measure** |
|---|---|
| TF-IDF_SVM | **0.844** |
| TF-IDF_NB | 0.74 |
| TF-IDF_RF | 0.836 |

Tabla 6.2: Promedio F-measure modelos enfoque TF-IDF

## 6.3. Resultados modelos enfoque Lexicón

Los resultados para este primer enfoque que usa Lexicones para la extracción de características del texto son presentados en la tabla 6.3. Para los corpus Chileno y ChilenoMexicano el modelo que mejor F-measure obtiene es el Lexicón_RF y para el Chileno es el Lexicón_SVM. Ningún modelo pudo superar a los modelos del enfoque base (TF-IDF), sin embargo, en el corpus chileno se acerca bastaste. En el caso de los corpus Mexicano y

Chileno no se superan los 0.7013 de F-measure, esto se puede dar por la falta de palabras mexicanas propias en los Lexicones.

| **Modelo** | **Corpus** | **F-measure** | **Accuracy** | **Precision** | **Recall** |
|---|---|---|---|---|---|
| Lexicon_SVM | Chileno | 0.8535 | 0.8556 | 0.8571 | 0.8556 |
| | Mexicano | 0.5735 | 0.6977 | 0.4868 | 0.6977 |
| | ChilenoMexicano | 0.6528 | 0.7154 | 0.7251 | 0.7154 |
| Lexicón_NB | Chileno | 0.8321 | 0.8367 | 0.8438 | 0.83670 |
| | Mexicano | **0.6408** | 0.6822 | 0.6377 | 0.6822 |
| | ChilenoMexicano | 0.6865 | 0.7113 | 0.6925 | 0.7113 |
| Lexicón_RF | Chileno | **0.8627** | 0.8636 | 0.8635 | 0.8636 |
| | Mexicano | 0.6347 | 0.7 | 0.6572 | 0.7 |
| | ChilenoMexicano | **0.7016** | 0.7211 | 0.7053 | 0.7211 |

Tabla 6.3: Resultados métricas para los modelos Lexicón

El modelo que mejor rendimiento tuvo en este enfoque es el Lexicón_RF según el promedio de F-measure en los 3 corpus, como se muestra en la tabla 6.4

| **Modelo** | **Promedio F-measure** |
|---|---|
| Lexicón_SVM | 0.6932 |
| Lexicón_NB | 0.7198 |
| Lexicón_RF | **0.733** |

Tabla 6.4: Promedio F-measure modelos enfoque Lexicón

## 6.4. Resultados modelos enfoque TF-IDF_Lexicón

La Tabla 6.5 muestra los resultados de este modelo que mezcla los 2 enfoques anteriores. Se puede observar que el modelo TF-IDF_Lexicón_RF obtiene la mejor F-measure en el corpus Chilenos y, además, le gana a los modelos de los enfoques anteriores. En el Mexicano y ChilenoMexicano, TF-IDF_Lexicón_SVM obtiene el mejor resultado en esta métrica y saca ventaja de los demás modelos. Esto demuestra que, al mezclar los dos modelos, se obtienen mejores resultados.

Como se observa en la Tabla 6.6, el modelo que obtuvo mejor rendimiento en este enfoque es TF-IDF_Lexicón_SVM, bajo el criterio del promedio de F-measure obtenidos en los corpus.

| Modelo | Corpus | F-measure | Accuracy | Precision | Recall |
|---|---|---|---|---|---|
| TF-IDF_Lexicón_SVM | Chileno | 0.8648 | 0.8663 | 0.8674 | 0.8663 |
| | Mexicano | **0.8330** | 0.8395 | 0.8363 | 0.8395 |
| | ChilenoMexicano | **0.8372** | 0.8439 | 0.8443 | 0.8439 |
| TF-IDF_Lexicón_NB | Chileno | 0.7829 | 0.7813 | 0.7895 | 0.7813 |
| | Mexicano | 0.7420 | 0.7486 | 0.7391 | 0.7486 |
| | ChilenoMexicano | 0.7543 | 0.7558 | 0.7531 | 0.7558 |
| TF-IDF_Lexicón_RF | Chileno | **0.8839** | 0.8852 | 0.8868 | 0.8852 |
| | Mexicano | 0.7960 | 0.8122 | 0.8131 | 0.8122 |
| | ChilenoMexicano | 0.8231 | 0.8340 | 0.84 | 0.8340 |

Tabla 6.5: Resultados métricas para los modelos TF-IDF_Lexicón

| Modelo | Promedio F-measure |
|---|---|
| TF-IDF_Lexicón_SVM | **0.845** |
| TF-IDF_Lexicón_NB | 0.7597 |
| TF-IDF_Lexicón_RF | 0.8343 |

Tabla 6.6: Promedio F-measure modelos enfoque TF-IDF_Lexicón

## 6.5. Resultados modelos enfoque Word Embedding

Al igual que el enfoque TF-IDF, este modelo se implementa para ser una base de comparación con los demás enfoques que mezclan Word Embedding, Lexicón y Machine Learning. Los resultados de los modelos de este enfoque se muestran en la Tabla 6.7, se puede observar que el modelo WordEmbedding_SVM obtiene mejor F-measure en los 3 corpus. Además, ningún modelo supera el rendimiento de los mejores modelos del enfoque anterior en cada uno de los corpus.

El modelo que mejor rendimiento tiene en este enfoque es WordEmbedding_SVM, bajo la lógica del promedio de F-measure en los corpus, como se observa en la Tabla 6.8.

| Modelo | Corpus | F-measure | Accuracy | Precision | Recall |
|---|---|---|---|---|---|
| WordEmbedding_SVM | Chileno | **0.8547** | 0.8569 | 0.8590 | 0.8569 |
| | Mexicano | **0.7831** | 0.7972 | 0.7913 | 0.7972 |
| | ChilenoMexicano | **0.7900** | 0.8021 | 0.8008 | 0.8021 |
| WordEmbedding_NB | Chileno | 0.8253 | 0.8259 | 0.8252 | 0.8259 |
| | Mexicano | 0.7504 | 0.745 | 0.7605 | 0.745 |
| | ChilenoMexicano | 0.7633 | 0.7599 | 0.7694 | 0.7599 |
| WordEmbedding_RF | Chileno | 0.8170 | 0.8218 | 0.8275 | 0.8218 |
| | Mexicano | 0.7296 | 0.7713 | 0.7899 | 0.7713 |
| | ChilenoMexicano | 0.7252 | 0.7616 | 0.7798 | 0.7616 |

Tabla 6.7: Resultados métricas para los modelos WordEmbedding

| Modelo | Promedio F-measure |
|---|---|
| WordEmbedding_SVM | **0.8092** |
| WordEmbedding_NB | 0.7796 |
| WordEmbedding_RF | 0.7572 |

Tabla 6.8: Promedio F-measure modelos enfoque WordEmbedding

## 6.6. Resultados modelos enfoque WE_Lexicón

La Tabla 6.9 muestra los resultados obtenidos por los modelos que mezclan Word Embedding y Lexicones. El modelo WE_Lexicón_SVM obtiene los mejores rendimientos considerando F-measure en los tres modelos y supera al enfoque base Word Embedding. Esto demuestra que, al agregar un análisis de Lexicones, los resultados mejoran.

| Modelo | Corpus | F-measure | Accuracy | Precision | Recall |
|---|---|---|---|---|---|
| WE_Lexicón_SVM | Chileno | **0.8908** | 0.8920 | 0.8936 | 0.8920 |
| | Mexicano | **0.7874** | 0.8027 | 0.7991 | 0.8027 |
| | ChilenoMexicano | **0.8086** | 0.8184 | 0.8181 | 0.8184 |
| WE_Lexicón_NB | Chileno | 0.8495 | 0.8502 | 0.8496 | 0.8502 |
| | Mexicano | 0.7551 | 0.7495 | 0.7658 | 0.7495 |
| | ChilenoMexicano | 0.7696 | 0.7664 | 0.7756 | 0.7664 |
| WE_Lexicón_RF | Chileno | 0.8833 | 0.8852 | 0.8892 | 0.8852 |
| | Mexicano | 0.7107 | 0.7590 | 0.7760 | 0.7590 |
| | ChilenoMexicano | 0.7257 | 0.7626 | 0.7832 | 0.7626 |

Tabla 6.9: Resultados métricas para los modelos WE_Lexicon

La Tabla 6.10 muestra el promedio de F-measure obtenidos por los modelos en los 3 corpus. Se demuestra que el modelo que obtiene mejor resultados es WE_Lexicón_SVM.

| Modelo | Promedio F-measure |
|---|---|
| WE_Lexicón_SVM | **0.8289** |
| WE_Lexicón_NB | 0.7914 |
| WE_Lexicón_RF | 0.7732 |

Tabla 6.10: Promedio F-measure modelos enfoque WE_Lexicón

## 6.7. Resultados modelos enfoque WE_Lexicón_TF-IDF

Los resultados de los modelos que mezclan los enfoques anteriores se muestran en la Tabla 6.11. Para los tres corpus, el modelo WE_Lexicón_TF-IDF_SVM obtiene el mejor rendimiento en la métrica F-measure. Además, supera en rendimiento a los mejores modelos del enfoque Word Embedding y para los corpus Mexicano y ChilenoMexicano es donde se obtiene el mejor rendimiento de todos los modelos según la métrica F-measure.

| Modelo | Corpus | F-measure | Accuracy | Precision | Recall |
|---|---|---|---|---|---|
| WE_Lexicón_TF-IDF_SVM | Chileno | **0.8731** | 0.8744 | 0.8755 | 0.8744 |
| | Mexicano | **0.8394** | 0.8431 | 0.8395 | 0.8431 |
| | ChilenoMexicano | **0.8507** | 0.8548 | 0.8534 | 0.8548 |
| WE_Lexicón_TF-IDF_NB | Chileno | 0.7842 | 0.7827 | 0.7906 | 0.7827 |
| | Mexicano | 0.7420 | 0.7486 | 0.7391 | 0.7486 |
| | ChilenoMexicano | 0.7537 | 0.7548 | 0.7529 | 0.7548 |
| WE_Lexicón_TF-IDF_RF | Chileno | 0.8501 | 0.8542 | 0.8630 | 0.8542 |
| | Mexicano | 0.7061 | 0.7590 | 0.7874 | 0.7590 |
| | ChilenoMexicano | 0.7033 | 0.7528 | 0.7941 | 0.7528 |

Tabla 6.11: Resultados métricas para los modelos WE_Lexicón_TF-IDF

La Tabla 6.12 demuestra que el mejor modelo de este enfoque es el WE_Lexicón_TF-IDF_SVM considerando el promedio de F-measure obtenida en los 3 corpus utilizados.

| Modelo | Promedio F-measure |
|---|---|
| WE_Lexicón_TF-IDF_SVM | **0.8544** |
| WE_Lexicón_TF-IDF_NB | 0.7599 |
| WE_Lexicón_TF-IDF_RF | 0.7531 |

Tabla 6.12: Promedio F-measure modelos enfoque WE_Lexicón_TF-IDF

## 6.8. Resultados modelos enfoque Ensemble

La Tabla 6.13 muestra los resultados obtenidos en las distintas métricas por los modelos creados bajo el enfoque de Ensemble. Se observa que en el corpus Chileno, considerando la métrica F-measure, el modelo que obtiene un mejor rendimiento es WE_Lexicón_TF-IDF_E_SVM, sin embargo, todos los modelos obtienen más de 0.88 en esta métrica. Para el corpus Mexicano el modelo que obtiene el mejor rendimiento en la métrica F-measure es TF-IDF_Lexicón_E_SVM. Por último, el modelo que obtiene un mejor rendimiento en el corpus ChilenoMexicano es TF-IDF_Lexicón_E_Clfs.

En la Tabla 6.14 se muestra el promedio de F-measure obtenido por los modelos en los 3 corpus. Se observa que el modelo TF-IDF_Lexicón_E_Clfs obtiene el mejor rendimiento considerando este promedio.

| Modelo | Corpus | F-measure | Accuracy | Precision | Recall |
|---|---|---|---|---|---|
| TF-IDF_Lexicón_E_Clfs | Chileno | 0.8828 | 0.8839 | 0.8848 | 0.8839 |
| | Mexicano | 0.8191 | 0.8309 | 0.8316 | 0.8309 |
| | ChilenoMexicano | **0.8399** | 0.8480 | 0.8518 | 0.8480 |
| TF-IDF_Lexicón_E_SVM | Chileno | 0.8716 | 0.8731 | 0.8745 | 0.8731 |
| | Mexicano | **0.8308** | 0.8386 | 0.8362 | 0.8386 |
| | ChilenoMexicano | 0.8356 | 0.8435 | 0.8461 | 0.8435 |
| WE_Lexicón_TF-IDF_E_SVM | Chileno | **0.8851** | 0.8866 | 0.8891 | 0.8866 |
| | Mexicano | 0.7879 | 0.8022 | 0.7977 | 0.8022 |
| | ChilenoMexicano | 0.8219 | 0.8310 | 0.8325 | 0.8310 |
| Enfoques_E_SVM | Chileno | 0.8804 | 0.8825 | 0.8873 | 0.8825 |
| | Mexicano | 0.7868 | 0.8081 | 0.8157 | 0.8081 |
| | ChilenoMexicano | 0.8146 | 0.8289 | 0.8415 | 0.8289 |

Tabla 6.13: Resultados métricas para los modelos Ensemble

| Modelo | Promedio F-measure |
|---|---|
| TF-IDF_Lexicón_E_Clfs | **0.8472** |
| TF-IDF_Lexicón_E_SVM | 0.846 |
| WE_Lexicón_TF-IDF_E_SVM | 0.8316 |
| Enfoques_E_SVM | 0.8272 |

Tabla 6.14: Promedio F-measure modelos enfoque Ensemble

## 6.9. Comparación de enfoques

A continuación, para comparar el rendimiento de los modelos de cada enfoque (agrupados por colores) en los distintos corpus se presentan una serie de gráficos para cada métrica utilizada.

La Figura 6.1 muestra el rendimiento de los modelos en la métrica f-measure en los 3 corpus utilizados. Se puede observar que el modelo que obtiene mejor rendimiento en esta métrica en el corpus Chileno es WE_Lexicón_SVM con 0.8908. En el corpus Mexicano y ChilenoMexicano es el WE_Lexicón_TF-IDF_SVM con 0.8394 y 0.8507, respectivamente.

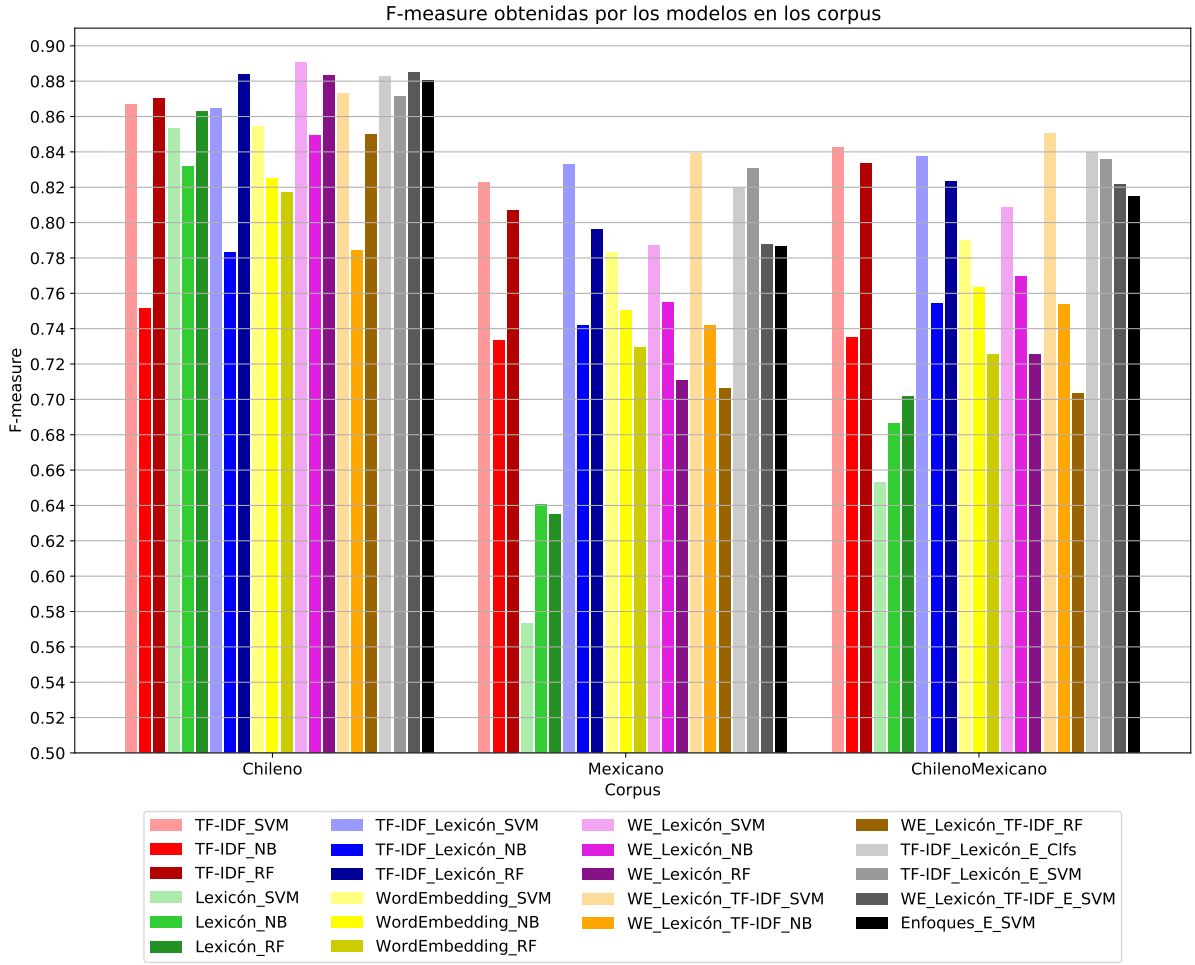


Figura 6.1: F-measure obtenidas por los modelos en los corpus

Como se puede observar en la Figura 6.2 el modelo que mejor rendimiento obtiene en la métrica Accuracy sobre el corpus Chileno es WE_Lexicón_SVM con 0.892. Para los corpus Mexicano y ChilenoMexicano es el modelo WE_Lexicón_TF-IDF_SVM con 0.8431 y 0.8548, respectivamente.

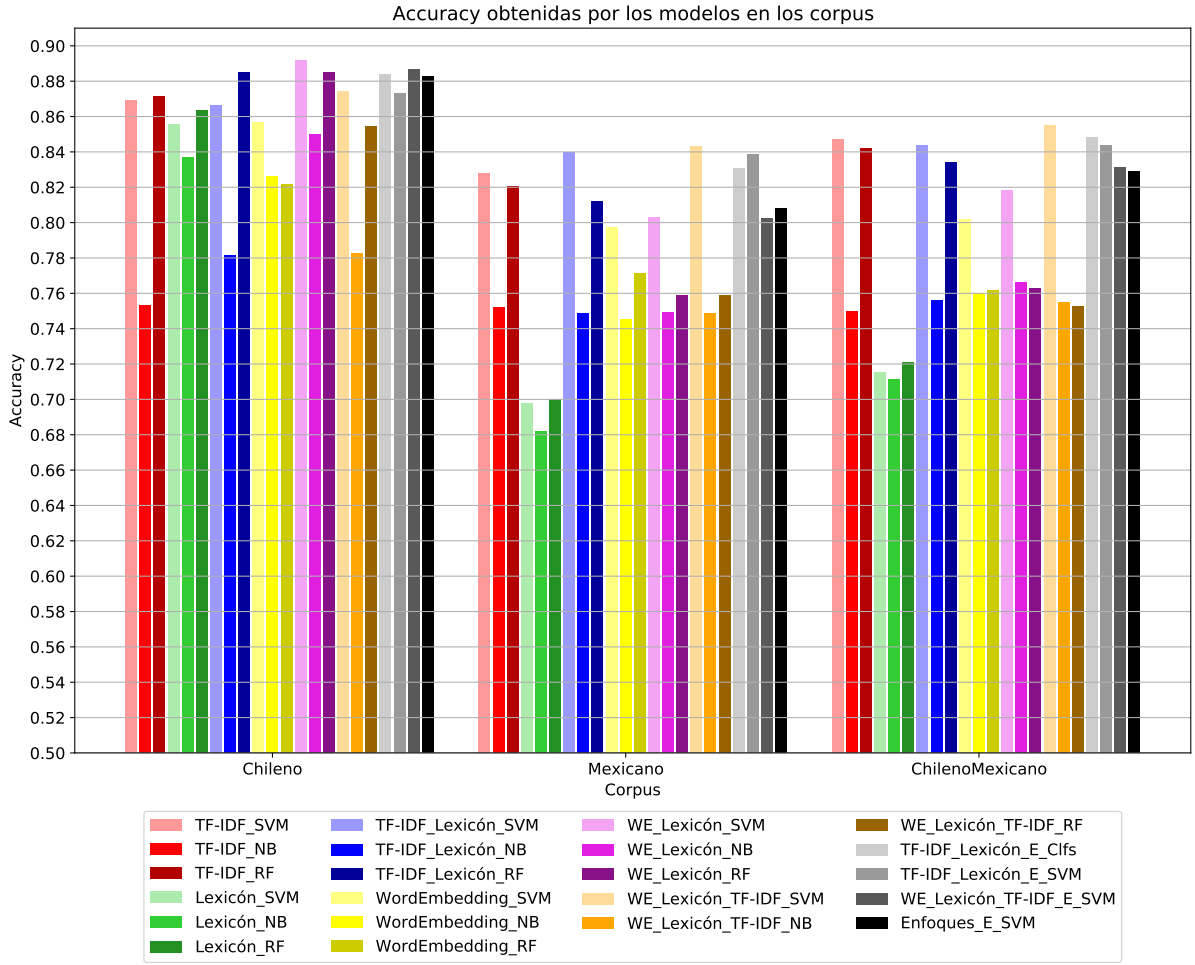


Figura 6.2: Accuracy obtenidas por los modelos en los corpus

La Figura 6.3 muestra el rendimiento de los modelos en la métrica Precision en los 3 corpus utilizados. Se puede observar que el modelo que obtiene mejor rendimiento en esta métrica en el corpus Chileno es WE_Lexicón_SVM con 0.8936. En el corpus Mexicano y ChilenoMexicano es el WE_Lexicón_TF-IDF_SVM con 0.8395 y 0.8534, respectivamente.

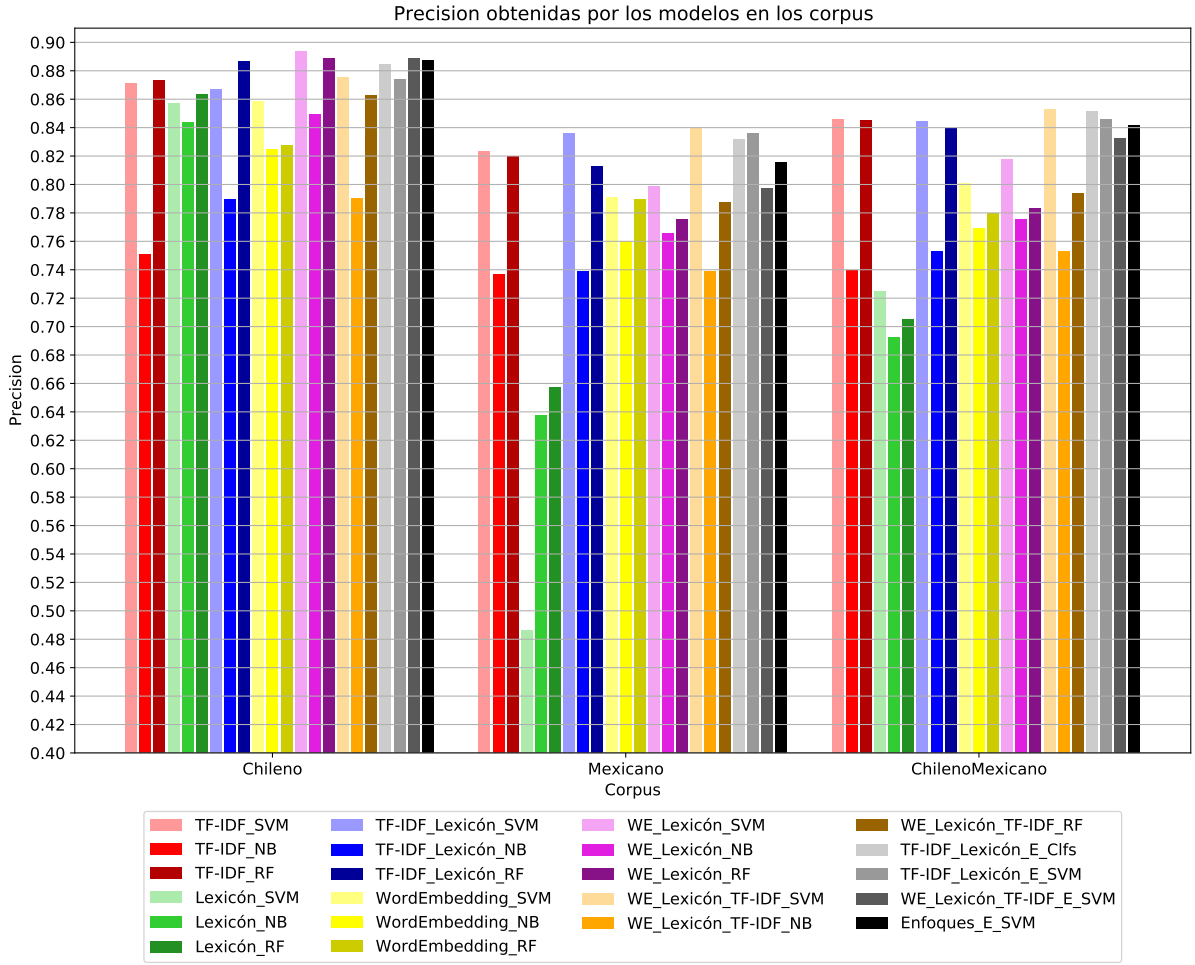


Figura 6.3: Precision obtenidas por los modelos en los corpus

El rendimiento de los modelos en la métrica Recall sobre los corpus es presentado en la Figura 6.4. Se puede observar que el modelo que obtiene mejor rendimiento en esta métrica en el corpus Chileno es WE_Lexicón_SVM con 0.892. En los corpus Mexicano y ChilenoMexicano es WE_Lexicón_TF-IDF_SVM con 0.8431 y 0.8548, respectivamente.

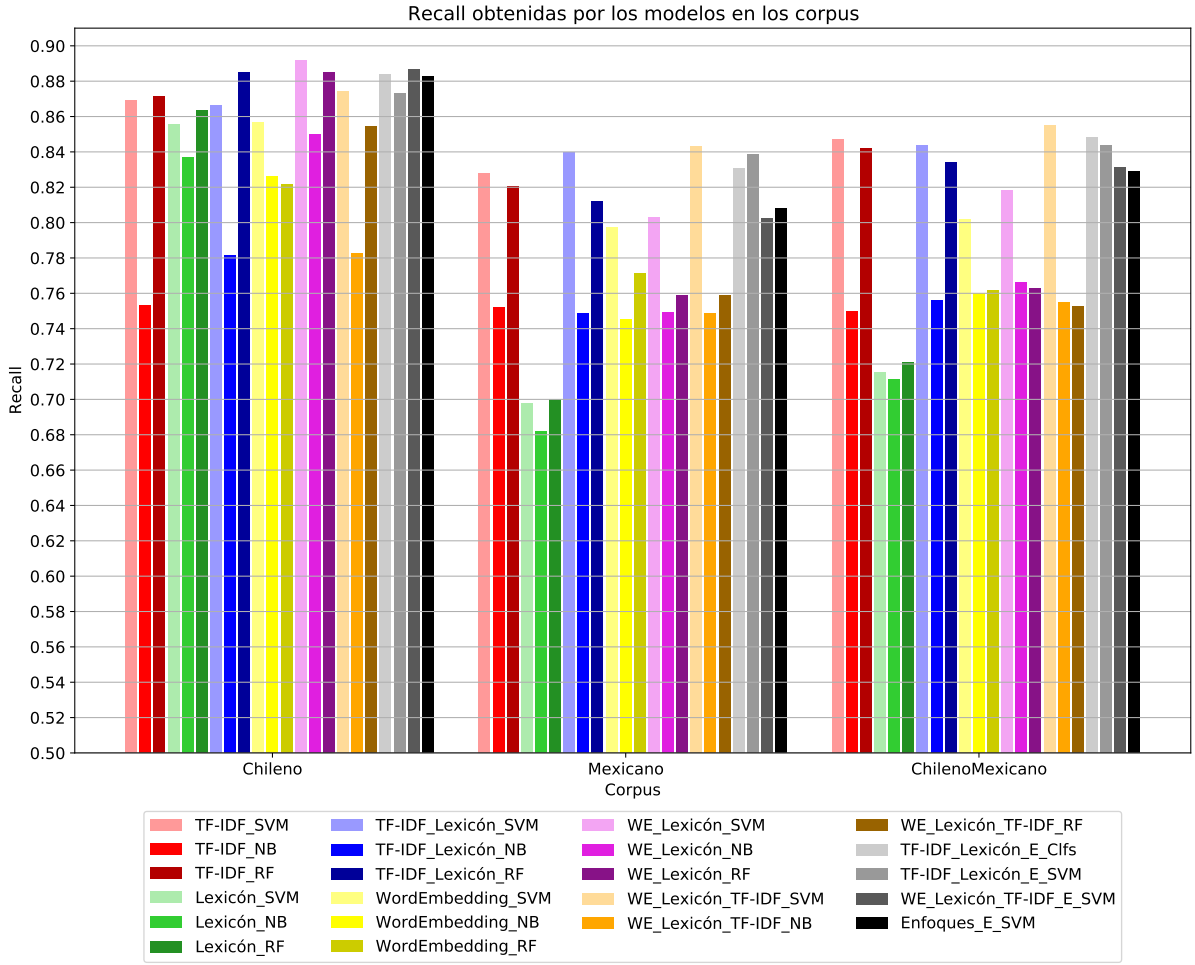


Figura 6.4: Recall obtenidas por los modelos en los corpus

En términos generales, se puede observar que en las distintas métricas los modelos tienen un comportamiento similar. Por otra parte, de forma general, los modelos tienen un mejor rendimiento en el corpus Chileno, seguido por el ChilenoMexicano y, por último, el Mexicano.

Desde los gráficos se puede concluir que los modelos con enfoque híbrido se comportan de mejor manera comparados con los enfoques que no usan Lexicones en el corpus Chileno, seguido por el Mexicano, y por último, el ChilenoMexicano. Como se observa en la Tabla 6.15, en el corpus Chileno 8 modelos híbridos superan el rendimiento del mejor modelo que no usa Lexicones. En el caso del corpus Mexicano 3 son los modelos híbridos que obtienen mejores rendimientos que el mejor modelo que no usa Lexicones, como se observa en la Tabla 6.16. Por último, en la Tabla 6.17 se observa que solo 1 modelo supera el rendimiento del mejor modelo híbrido que no usa Lexicones en el corpus ChilenoMexicano.

| **Modelo** | **F-measure** |
|---|---|
| WE_Lexicón_SVM | 0.8908 |
| WE_Lexicón_TF-IDF_E_SVM | 0.8851 |
| TF-IDF_Lexicón_RF | 0.8839 |
| WE_Lexicón_RF | 0.8833 |
| TF-IDF_Lexicón_E_Clf | 0.8828 |
| Enfoques_E_SVM | 0.8804 |
| WE_Lexicón_TF-IDF_SVM | 0.8731 |
| TFIDF_Lexicón_E_SVM | 0.8716 |
| TF-IDF_RF (No usa Lexicones) | 0.8701 |

Tabla 6.15: Modelos híbridos que superan al mejor modelo que no usa Lexicones en el corpus Chileno

| **Modelo** | **F-measure** |
|---|---|
| WE_Lexicon_TF-IDF_SVM | 0.8394 |
| TF-IDF_Lexicón_SVM | 0.833 |
| TFIDF_Lexicón_E_SVM | 0.8308 |
| TF-IDF_SVM (No usa Lexicones) | 0.8225 |

Tabla 6.16: Modelos híbridos que superan al mejor modelo que no usa Lexicones en el corpus Mexicano

| **Modelo** | **F-measure** |
|---|---|
| WE_Lexicón_TF-IDF_SVM | 0.8507 |
| TF-IDF_SVM (No usa Lexicones) | 0.8424 |

Tabla 6.17: Modelos híbridos que superan al mejor modelo que no usa Lexicones en el corpus ChilenoMexicano

## 6.10. Mejores modelos por corpus

A modo de resumen, en la Tabla 6.18 se muestran los modelos que obtienen mejores resultados según las métrica F-measure y Accuracy por cada corpus utilizado. Se observa que para el corpus Chileno es el modelo WE_Lexicón_SVM y para el corpus Mexicano y ChilenoMexicano es WE_Lexicón_TF-IDF_SVM. Estos dos modelos usan Word Embedding y Lexicones para la extracción de características de los textos, esto demuestra que al incorporar estas técnicas se obtienen buenos resultados.

Por otra parte, se observa que en el corpus Chileno se obtiene el mejor resultado pasando los 0.89 de F-measure y Accuracy, seguido por el corpus ChilenoMexicano, y por el último, el Mexicano. Esto se puede dar por la falta de palabras propias mexicanas en los diferentes corpus utilizados.

Por último, todos los modelos ganadores utilizan Support Vector Machine como clasificador de Machine Learning, con esto se reafirma que es el mejor algoritmo para realizar clasificación de texto de los tres algoritmos probados.

| **Corpus** | **Modelo** | **F-measure** | **Accuracy** |
|---|---|---|---|
| Chileno | WE_Lexicón_SVM | 0.8908 | 0.892 |
| Mexicano | WE_Lexicón_TF-IDF_SVM | 0.8394 | 0.8431 |
| ChilenoMexicano | WE_Lexicón_TF-IDF_SVM | 0.8507 | 0.8548 |

Tabla 6.18: Mejores modelos por corpus

La Tabla 6.19 muestra los resultados que obtienen en la métrica F-measure los mejores modelos de cada corpus (Tabla 6.18) y los modelos de los enfoques base que obtienen mejores resultados. Se puede observar que la diferencia más amplia se encuentra en el corpus Chileno seguido por el Mexicano, y por último, el ChilenoMexicano. También se observa que la diferencia es más amplia con los modelos que están bajo el enfoque base Word Embedding.

| **Corpus** | **Mejor Modelo** | **Mejor modelo enfoque TF-IDF** | **Mejor modelo enfoque WordEmbedding** |
|---|---|---|---|
| Chileno | 0.8908 | 0.8701 | 0.8547 |
| Mexicano | 0.8394 | 0.8225 | 0.7831 |
| ChilenoMexicano | 0.8507 | 0.8424 | 0.7900 |

Tabla 6.19: Comparación de F-measure

# Capítulo 7

# Aplicación web desarrollada

Se desarrolla una aplicación web[1] para darle aplicabilidad a los distintos modelos creados y evaluarlos con distintos corpus etiquetados. A continuación, se presenta de forma general la aplicación web implementada.

Para el backend se utiliza el framework FastAPI[2] que permite contruir APIs mediante el lenguaje de programación Python, además de las siguientes librerías:

- Pandas 1.1.4: Librería destinada al análisis de datos, que proporciona estructuras de datos flexibles y que permite trabajar con ellas de forma fácil y eficiente. Específicamente, se utiliza para cargar y manejar los corpus y Lexicones.
- NLTK 3.5: Conjunto de bibliotecas y programas para el Procesamiento del Lenguaje Natural simbólico y estadísticos para el lenguaje de programación Python. Se utiliza específicamente para eliminar puntación, caracteres no legibles,tokenizar etc.
- Scikit-learn (sklearn) 0.22: Librería de Machine Learning para Python de fácil uso y eficiente.
- Numpy 1.19: Librería de funciones matemáticas de alto nivel para operar de forma sencilla con vectores o matrices. Se utiliza para manejar de forma eficiente los vectores de características que reciben los algoritmos de Machine Learning.
- Tweepy 3.10: Librería que permite la conexión a la API pública de Twitter para realizar las consultas a esta de forma sencilla y eficiente.

El frontend se implementó utilizando el framework de javascript Vue.js[3], además de la librería de interfaz de usuario Vuetify[4] para Vue.js.

Por último, se utiliza Docker[5] y Docker-compose[6] para desplegar la web.

[1] http://35.192.83.211/
[2] https://fastapi.tiangolo.com/
[3] https://vuejs.org/
[4] https://vuetifyjs.com
[5] https://docs.docker.com/
[6] https://docs.docker.com/compose/

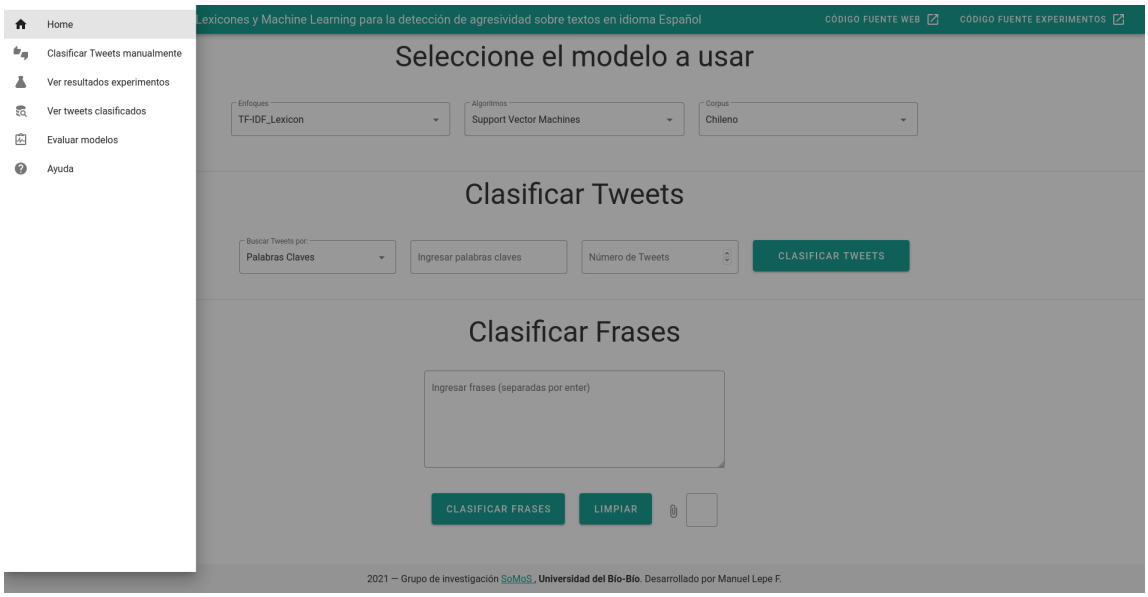


Figura 7.1: Inicio y menú de la aplicación web

En la Figura 7.1 se muestra la interfaz de usuario de la página de inicio y el menú de la aplicación web. A continuación, se detalla cada una de las funcionalidades de la web.

## 7.1. Clasificar tweets

El objetivo de este módulo es poder clasificar la agresividad de los tweets más recientes, que contengan algunas palabras claves o hashtag y los tweets escritos por un usuario en específico. El módulo entrega la opción de elegir entre estas dos opciones descritas anteriormente, las palabras claves o nombre de usuario y la cantidad de tweets. Antes de clasificar se debe elegir el modelo a usar, esto se realiza escogiendo enfoque, algortimo de Machine Learning (si corresponde) y corpus donde fue entrenado. Además, el módulo entrega la opción de descargar en un archivo excel los tweets clasificados.

En la Figura 7.2 se observa un ejemplo utilizando el hashtag #Piñera y clasificando 50 tweets con el modelo TF-IDF_Lexicón_SVM entrenado con el corpus Chileno, el resultado nos muestra cada tweet clasificado y el porcentaje total de tweets por cada clase. Por último, en la Figura 7.3 se muestra un ejemplo clasificando los tweets escritos por el usuario @sebastianpinera. Es importante señalar que no se clasifican los retweets evitando que aparezcan tweets repetidos.

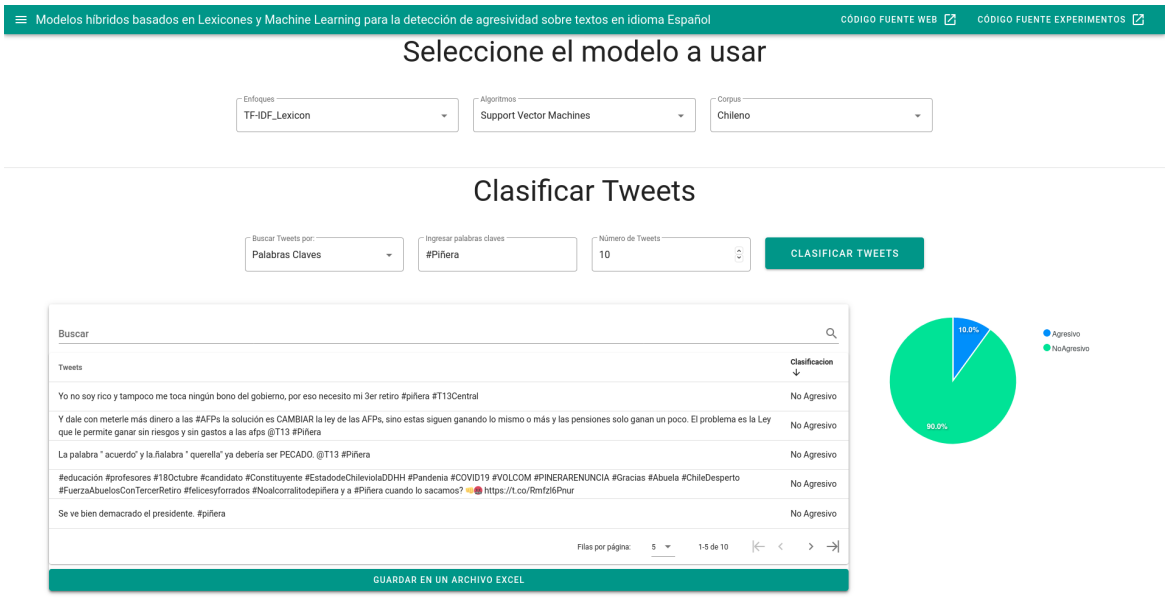


Figura 7.2: Clasificación hashtag #Piñera

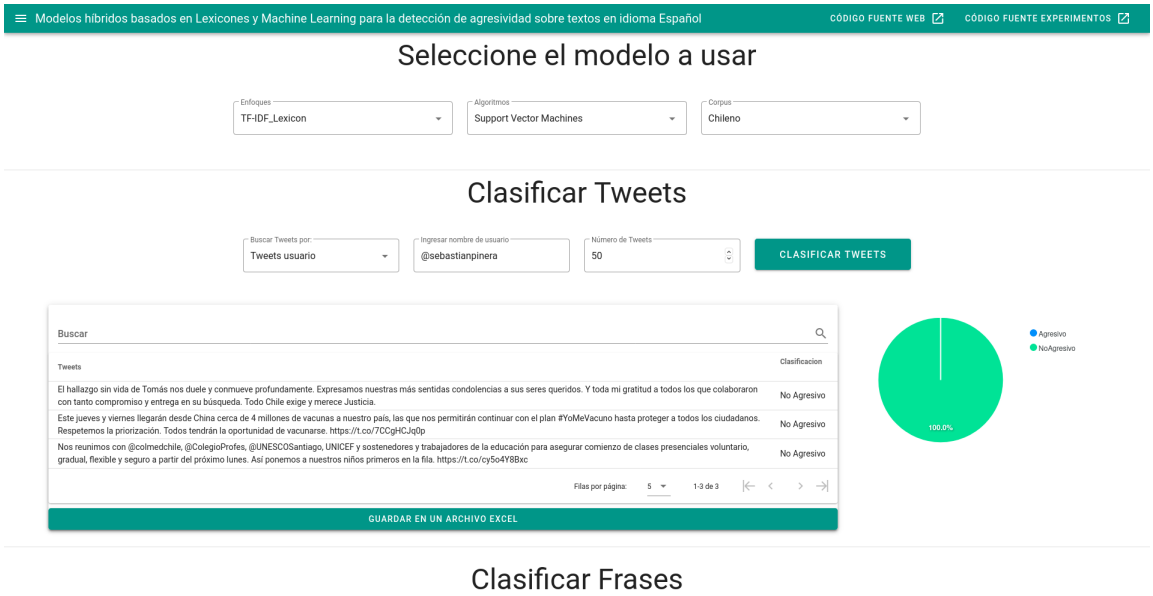


Figura 7.3: Clasificación tweets usuario @sebastianpinera

## 7.2. Clasificar frases

Este módulo permite clasificar frases separadas por un salto de línea (presionando enter) con el modelo seleccionado, asimismo permite cargar las frases mediante un archivo txt. Además, entrega la opción de descargar en un archivo excel los tweets clasificados. La Figura 7.4 muestra la clasificación de dos frases usando el modelo WE_Lexicón_TF-IDF entrenado en el corpus Chileno.

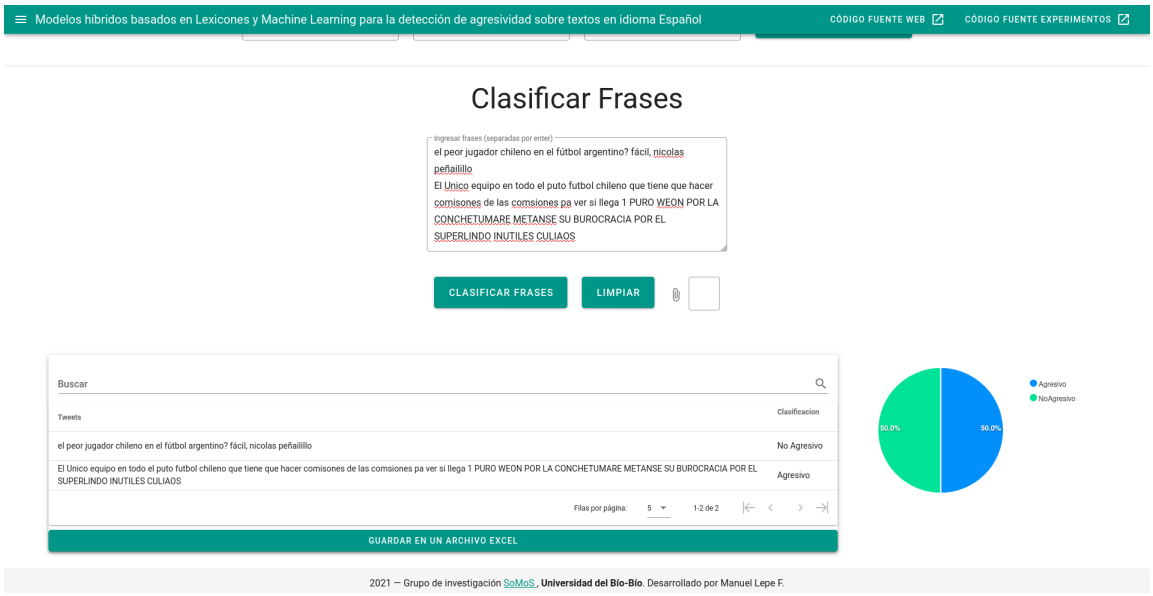


Figura 7.4: Clasificación frases

## 7.3. Clasificar tweets manualmente

Permite retroalimentar a los modelos clasificando manualmente los tweets como se observa en la Figura 7.5. La clasificación manual y la del modelo quedan guardadas en una base de datos para posteriores investigaciones.

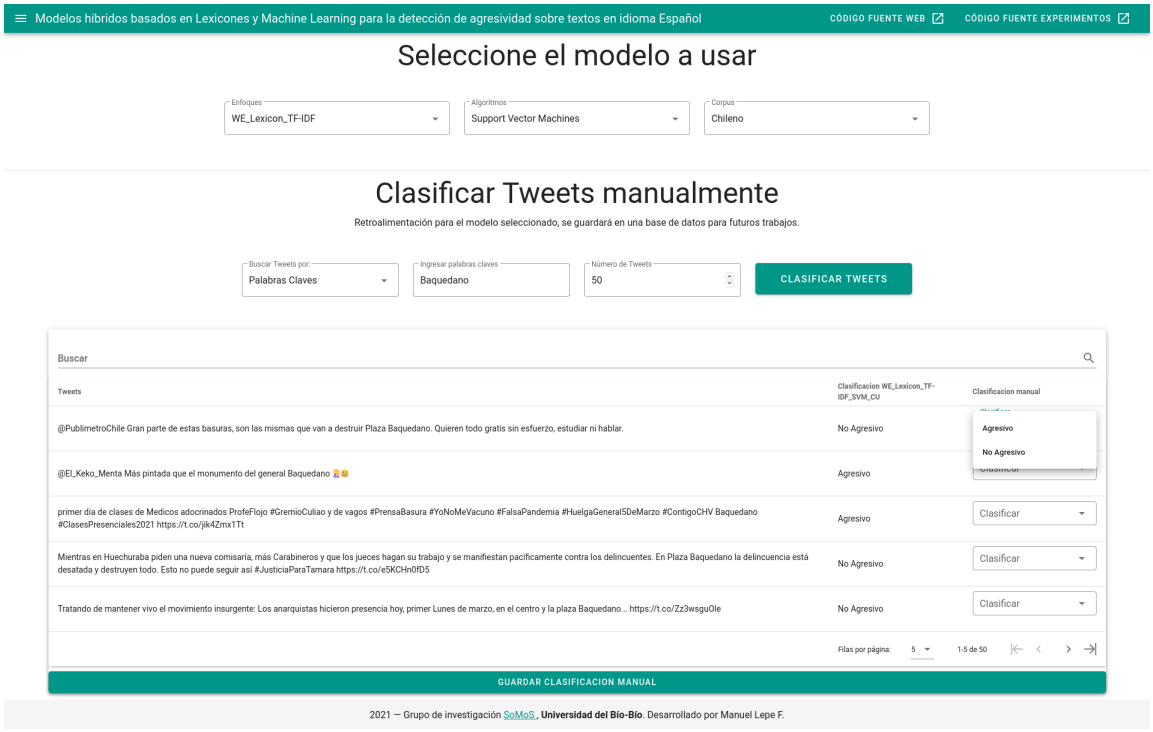


Figura 7.5: Clasificación de tweets manual

## 7.4. Ver resultados experimentos

Permite ver los resultados de los experimentos sobre los corpus de prueba de todos los modelos como se observa en la Figura 7.6, los resultados se encuentran guardados en una base de datos. Además, se permite descargar los resultados en un archivo excel y los corpus utilizados.

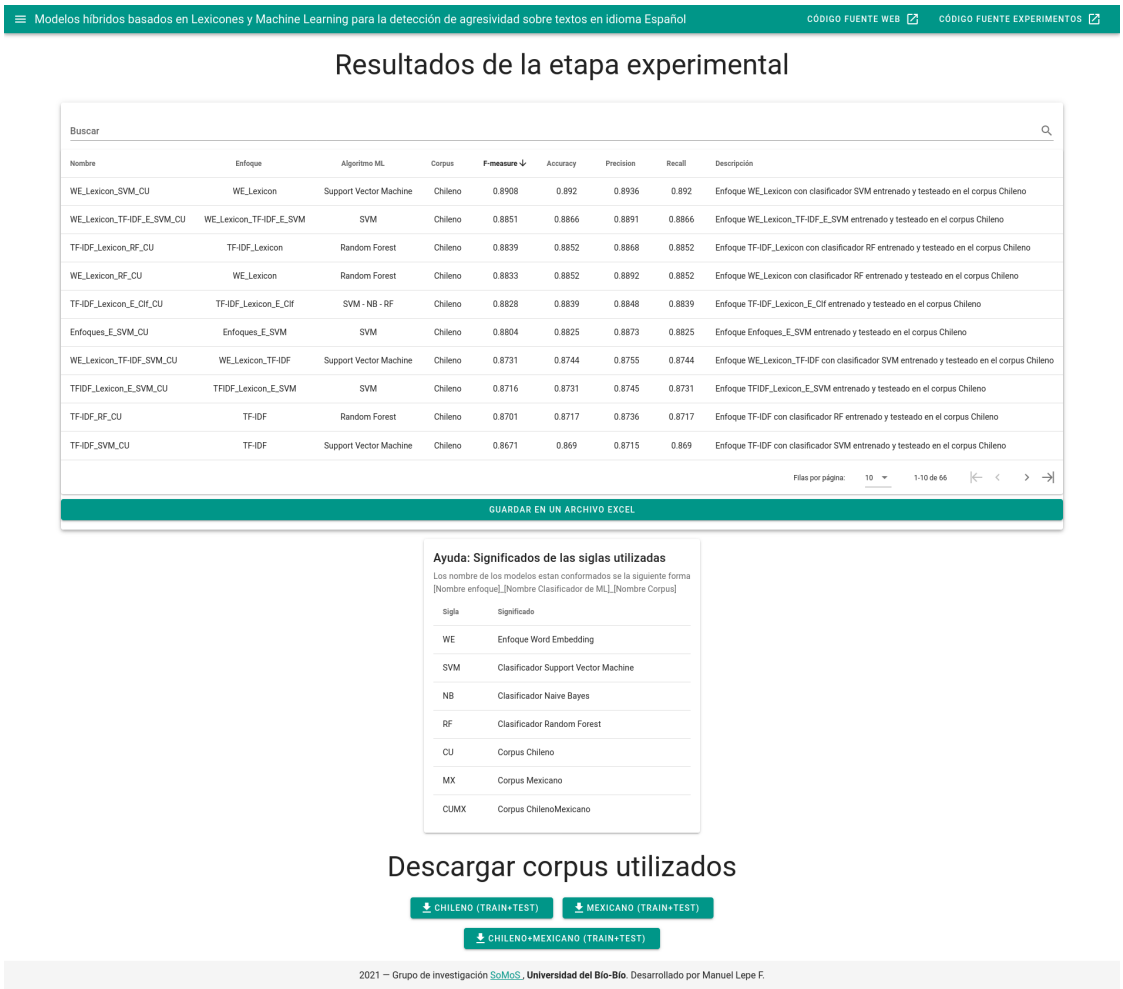


Figura 7.6: Resultados de los modelos

## 7.5. Ver tweet clasificados

Permite ver los tweets clasificados manualmente guardados en la base de datos como se observa en la Figura 7.7, además entrega la posibilidad de descargarlos en un archivo .xls.


Figura 7.7: Tweets clasificados manualmente

## 7.6. Evaluar modelos

Este módulo permite evaluar el despeño del modelo seleccionado con corpus de pruebas que estén estructurados de la misma forma que la plantilla descargable. Entrega resultados para las métricas de F-measure, accuracy, precision y recall, como se muestra en la Figura 7.8

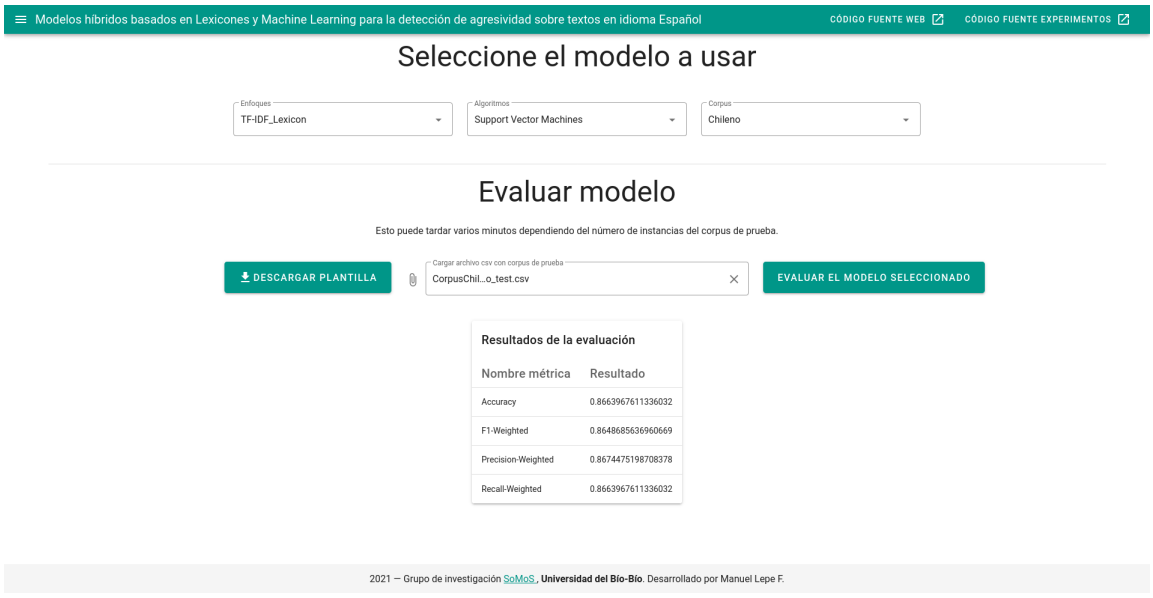


Figura 7.8: Modulo evaluar modelos

# Capítulo 8

# Conclusiones y trabajos futuros

## 8.1. Conclusiones generales

El presente trabajo entrega un aporte a la investigación con el desarrollo de nuevos modelos para detectar agresividad sobre textos en idioma español. Estos modelos se basan en la idea de utilizar el análisis de emociones con Lexicones afectivos en conjunto con el análisis de Machine Learning.

Se proponen 5 enfoques para crear diferentes modelos; Lexicón, TF-IDF_Lexicón, WE_Lexicón, WE_Lexicón_TF-IDF y enfoque Ensemble, estos enfoques principalmente se diferencian en la forma de extraer el vector de características del texto como se explica en el capítulo 5. Además, se implementan 2 enfoques base TF-DF y Word Embedding que no usan Lexicones para comparar con los demás modelos implementados.

En cada uno de los modelos creados se buscan los mejores hiperparámetros sobre el dataset de entrenamiento de cada corpus usando GridSearchCv, para luego realizar la experimentación sobre el dataset de prueba y con esto comparar los resultados obtenidos en cada modelo y seleccionar los mejores modelos en cada uno de los corpus. Todos los modelos que obtuvieron mejores resultados en los corpus usan enfoques que mezclan Word Embedding, Lexicones y clasificadores de Machine Learning como se muestra en la Tabla 6.18 superando los modelos base. Cabe destacar que los modelos híbridos tienen un mejor rendimiento en el corpus Chileno porque los Lexicones tienen mayor cobertura o coincidencia con el español de Chile que con el español de México.

Por último, se realiza una aplicación web que permite darle aplicabilidad a los distintos modelos implementados, permitiendo clasificar tweets, clasificar frases, evaluar los modelos implementados y recibir retroalimentación de los usuarios sobre la predicción de los modelos que quedan guardados en una base de datos para próximas investigaciones. Además, el backend de la aplicación web se implementa como una API, lo que permite ser consumida por servicios externos.

## 8.2. Conclusiones hipótesis y objetivos

A continuación, se detallan las conclusiones sobre la hipótesis y objetivos declarados en el capítulo 2. Para la hipótesis definida como **Los modelos híbridos que mezclan el enfoque de Lexicones y Machine Learning permiten mejorar el rendimiento de la predicción de agresividad presente en textos en idioma español**, se concluye que se cumple, ya que los modelos que obtuvieron los mejores resultados en cada uno de los corpus usan enfoques que mezclan Lexicones y clasificadores de Machine Learning como se muestra en la Tabla 6.18.

Las conclusiones de los objetivos específicos se detallan a continuación:

1. **Revisar el estado del arte de los distintos trabajos que tengan como objetivo predecir agresividad en texto usando modelos de Machine Learning y Lexicones, principalmente en idioma español:** Se cumple el objetivo de revisar los diferentes trabajos en idioma español como se presenta en la sección 3.2.4, donde se muestra un resumen de los trabajos más importantes.

2. **Crear diferentes modelos híbridos; usando el enfoque de Lexicones y Machines Learning, principalmente para la extracción de característica del texto:** Se implementan diferentes modelos usando 5 enfoques que mezclan Lexicones y clasificadores de Machine Learning; Lexicón_TF-IDF, WE_Lexicón, WE_Lexicón _TF-IDF y enfoque Ensemble, además de implementar enfoques base que no usan Lexicones; TF-IDF y WordEmbedding para comparar los resultados. En total se implementaron 22 modelos y cada uno de ellos se entrenaron y probaron usando 3 corpus: Chileno, Mexicano y ChilenoMexicano.

3. **Comparar el rendimiento de los distintos modelos creados en diferentes corpus en idioma español mediante una herramienta web que, además, quedará disponible para darle aplicabilidad a los modelos creados y recibir retroalimentación de los usuarios.** Se logró desarrollar una aplicación web[1] que cumple los objetivos de darle aplicabilidad a los modelos, recibir retroalimentación de los usuarios y evaluar los modelos implementados.

4. **Analizar los resultados obtenidos por los distintos modelos para generar conclusiones objetivas y proponer trabajos futuros.** Dado los experimentos realizados para medir el rendimiento de los modelos en diferentes corpus usando distintas métricas, se llegaron a conclusiones sobre los modelos implementados y trabajos futuros a realizar.

Al cumplir todos los objetivos específicos definidos se cumple el objetivo general de esta tesis definido como **Crear y evaluar distintos modelos híbridos para identificar agresividad en textos en idioma español, los que quedarán disponibles en una plataforma web que permitirá recibir retroalimentación de los distintos**

[1] http://35.192.83.211/

**usuarios**, la hipótesis se prueba y confirma con los resultados de los experimentos y queda disponible una aplicación web para el uso del público general e investigadores.

## 8.3. Conclusiones sobre los modelos implementados

A continuación, se presentan conclusiones sobre los modelos implementados en base a los resultados:

- El enfoque Lexicón no logra superar los enfoques base, por lo tanto, es mejor usarlo en conjunto con otros enfoques.
- Los modelos que obtienen un mejor resultado usan enfoques que mezclan Word Embedding, Lexicones y clasificadores de Machine Learning como se muestra en la Tabla 6.18, superando los modelos base. Por lo tanto, se concluye que es mejor mezclar mezclar diferentes enfoques en el proceso de extracción de características.
- Todos los modelos que obtienen mejores resultados en los corpus usan como clasificador de Machine Learning Support Vector Machine. Se concluyendo, por lo tanto, que este es el mejor algoritmo para la clasificación de agresividad considerando los clasificadores implementados.
- Los modelos de Ensemble implementados no obtienen los resultados esperados, en la mayoría de los corpus no supera al modelo base TF-IDF. Sin embargo, sería una buena estrategia implementarlos usando mayor número de clasificadores distintos, más que centrarse en mezclar los enfoques para extraer las características de los textos, dado que el modelo que se implementó bajo esta metodología obtiene mejores resultados, como se muestra en la Tabla 6.14.
- Como se observa en los resultados, los modelos se comportan de mejor forma en el corpus Chileno y ChilenoMexicano comparados con el corpus Mexicano, esto se puede dar por las pocas palabras mexicanas que se encuentran en los Lexicones, se puede volver a experimentar agregando más palabras mexicanas.

## 8.4. Trabajos futuros

Para trabajos futuros, se podría considerar la inclusión de palabras mexicanas en los distintos Lexicones utilizados, especialmente en el de malas palabras, para verificar si los rendimientos de los modelos implementados sobre el corpus mexicano mejora. Así mismo, utilizar distintos Lexicones tipo diccionarios, ya que estos incluyen más palabras que el Lexicón utilizado en este trabajo. También se puede implementar el manejo de cuantificadores, negaciones y emojis en el preprocesamiento del texto ya que este trabajo no lo considera. Por otra parte, sería adecuado experimentar con más modelos Ensemble usando distintos clasificadores de Machine Learning. Por último, experimentar con distintos modelos que usen redes neuronales para clasificar y mezclarlos con Lexicones. En especial los

modelos de transformer del tipo BERT ya entrenados, como es el caso de BETO para el idioma español, ya que a diferencia de las demás redes neuronales, no requiere gran cantidad de datos para tener un buen rendimiento, esto se demuestra para el caso particular de clasificar agresividad en los trabajos del workshop del año 2020, visto en las sección 3.2.4.

# Referencias


Mohammed Ali Al-garadi, Kasturi Dewi Varathan, y Sri Devi Ravana. Cybercrime detection in online communications: The experimental case of cyberbullying detection in the twitter network. *Computers in human behavior*, 63:433–443, 2016. ISSN 07475632. doi: 10.1016/j.chb.2016.05.051. URL http://linkinghub.elsevier.com/retrieve/pii/S0747563216303788.

Mario Ezra Aragón, Miguel Ángel Álvarez Carmona, Manuel Montes-y Gómez, Hugo Jair Escalante, Luis Villaseñor Pineda, y Daniela Moctezuma. Overview of MEX-A3T at IberLEF 2019: Authorship and aggressiveness analysis in mexican spanish tweets. En *IberLEF@ SEPLN*, págs. 478–494. 2019.

Vimala Balakrishnan, Shahzaib Khan, Terence Fernandez, y Hamid R. Arabnia. Cyberbullying detection on twitter using big five and dark triad features. *Personality and Individual Differences*, 141:252–257, 2019. ISSN 01918869. doi:10.1016/j.paid.2019.01.024. URL https://linkinghub.elsevier.com/retrieve/pii/S0191886919300364.

Bill Belsey. Cyberbullying. 2004. URL http://www.cyberbullying.ca.

Anselm Blumer, A. Ehrenfeucht, David Haussler, y Manfred K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989. ISSN 00045411. doi:10.1145/76359.76371. URL http://portal.acm.org/citation.cfm?doid=76359.76371.

Piotr Bojanowski, Edouard Grave, Armand Joulin, y Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. ISSN 2307-387X. doi:10.1162/tacl_a_00051.

Leo Breiman. Random forests. *Springer Science and Business Media LLC*, 2001. doi:10.1023/a:1010933404324. URL http://link.springer.com/10.1023/A:1010933404324.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen,

Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, y Dario Amodei. Language models are few-shot learners. 2020.

Marilyn Campbell y Natalie Morgan. Adults'perceptions of bullying in early childhood. En Phillip T. Slee, Grace Skrzypiec, y Carmel Cefai, eds., *Child and adolescent well-being and violence prevention in schools*, págs. 101–108. Routledge, 2017. ISBN 9781315102047. doi:10.4324/9781315102047-10. URL https://www.taylorfrancis.com/books/9781351590884/chapters/10.4324/9781315102047-10.

Cristian Cardellino. Spanish {B}illion {W}ords {C}orpus and {E}mbeddings. 2019. URL https://crscardellino.github.io/SBWCE/.

Marco Casavantes, Roberto López, y Luis Carlos González. Uach at mex-a3t 2019: Preliminary results on detecting aggressive tweets by adding author information via an unsupervised strategy. 2019.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, y Jorge Pérez. Spanish pre-trained bert model and evaluation data. En *PML4DC at ICLR 2020*. 2020.

Pancracio Celdrán. *El gran libro de los insultos: tesoro crtico, etimológico e histórico de los insultos españoles*. La Esfera de los Libros, 2009.

Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, y Athena Vakali. Mean birds: detecting aggression and bullying on twitter. En *Proceedings of the 2017 ACM on Web Science Conference - WebSci '17*, págs. 13–22. ACM Press, New York, New York, USA, 2017. ISBN 9781450348966. doi:10.1145/3091478.3091487. URL http://dl.acm.org/citation.cfm?doid=3091478.3091487.

Carlos Enrique Muñiz Cuza, Gretel Liz De, Peña Sarracén, y Paolo Rosso. Attention mechanism for aggressive detection. 2018. URL https://mexa3t.wixsite.com/home.

Laura P. Del Bosque y Sara Elena Garza. Aggressive text detection for cyberbullying. En Alexander Gelbukh, Félix Castro Espinoza, y Sofía N. Galicia-Haro, eds., *Human-Inspired Computing and Its Applications*, tomo 8856 de *Lecture notes in computer science*, págs. 221–232. Springer International Publishing, Cham, 2014. ISBN 978-3-319-13646-2. doi:10.1007/978-3-319-13647-9_21. URL http://link.springer.com/10.1007/978-3-319-13647-9_21.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, y Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. En *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, págs. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota, 2019. doi:10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Elgueta. Comparación de rendimiento de técnicas de aprendizaje automático para análisis de afecto sobre textos en español. 2017. URL http://repobib.ubiobio.cl/jspui/handle/123456789/1772.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, y Chris Welty. Building watson: an overview of the deepqa project. *AI Magazine*, 31(3):59, 2010. ISSN 0738-4602. doi:10.1609/aimag.v31i3.2303. URL https://aaai.org/ojs/index.php/aimagazine/article/view/2303.

Maite Garaigordobil, Juan Pablo Mollo-Torrico, y Enara Larrain. Prevalencia de bullying y cyberbullying en latinoamérica: una revisión. *Research in Pharmacy*, 11(3):1–18, 2019. ISSN 2500-6517. doi:10.33881/2027-1786.rip.11301. URL https://reviberopsicologia.ibero.edu.co/article/view/rip.11301.

Denis Gordeev. Automatic detection of verbal aggression for russian and american imageboards. *Procedia - Social and Behavioral Sciences*, 236:71–75, 2016. ISSN 18770428. doi:10.1016/j.sbspro.2016.12.022. URL http://linkinghub.elsevier.com/retrieve/pii/{S187704281631655X}.

G Grefenstette, Y Qu, JG Shanahan, y DA Evans. Coupling niche browsers and affect analysis for an opinion mining application. *In Proceedings of the 12th International Conference Recherche dÍnformation Assistee par Ordinateur*, págs. 186–194, 2004. URL http://scholar.google.com/scholar?q=Coupling+Niche+Browsers+and+Affect+Analysis+for+an+Opinion+Mining+Application.#0.

Mario Guzman-Silverio, Ángel Balderas-Paredes, y Adrián Pastor López-Monroy. Transformers and data augmentation for aggressiveness detection in mexican spanish. 2020. URL https://www.cimat.mx/es/adri.

Janet Hicks, Lynn Jennings, Stephen Jennings, Stephan Berry, y Dee-Anna Green. Middle school bullying: student reported perceptions and prevalence. *Journal of Child and Adolescent Counseling*, 4(3):195–208, 2018. ISSN 2372-7810. doi:10.1080/23727810.2017.1422645. URL https://www.tandfonline.com/doi/full/10.1080/23727810.2017.1422645.

Chih-Wei Hsu, Chih-Chung Chang, y Chih-Jen Lin. A practical guide to support vector classification. 2008. URL https://towardsdatascience.com/support-vector-machines-for-classification-fc7c1565e3.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, y Jeffrey Dean. Google multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. ISSN 2307-387X. doi:10.1162/tacl_a_00065. URL https://www.mitpressjournals.org/doi/abs/10.1162/tacl_a_00065.

Barbara Kitchenham. Procedures for performing systematic literature reviews. *Joint Technical Report, Keele University TR/SE-0401 and NICTA TR-0400011T.1*, pág. 33, 2004.

Vijay Kotu y Bala Deshpande. Data mining process. En *Predictive analytics and data mining*, págs. 17–36. Elsevier, 2015. ISBN 9780128014608. doi:10.1016/B978-0-12-801460-8.00002-1. URL https://linkinghub.elsevier.com/retrieve/pii/B9780128014608000021.

Hitesh Kumar Sharma, K Kshitiz, y Shailendra. NLP and machine learning techniques for detecting insulting comments on social networking platforms. En *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, págs. 265–272. IEEE, 2018. ISBN 978-1-5386-4485-0. doi:10.1109/{ICACCE}.2018.8441728. URL https://ieeexplore.ieee.org/document/8441728/.

Gabriel A. Leon-Paredes, Wilson F. Palomeque-Leon, Pablo L. Gallegos-Segovia, Paul E. Vintimilla-Tapia, Jack F. Bravo-Torres, Liliana I. Barbosa-Santillan, y Maria M. Paredes-Pinos. Presumptive detection of cyberbullying on twitter through natural language processing and machine learning in the spanish language. En *2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, págs. 1–7. IEEE, 2019. ISBN 978-1-7281-3185-6. doi:10.1109/{CHILECON47746}.2019.8987684. URL https://ieeexplore.ieee.org/document/8987684/.

Edward Loper y Steven Bird. NLTK: the natural language toolkit. En *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics* -, págs. 63–70. Association for Computational Linguistics, Morristown, NJ, USA, 2002. doi:10.3115/1118108.1118117. URL http://portal.acm.org/citation.cfm?doid=1118108.1118117.

Rolfy Nixon Montufar Mercado, Hernan Faustino, y Eveling Gloria. Automatic cyberbullying detection in spanish-language social networks using sentiment analysis techniques. *International Journal of Advanced Computer Science and Applications*, 9(7), 2018. ISSN 21565570. doi:10.14569/{IJACSA}.2018.090733. URL http://thesai.org/Publications/{ViewPaper}?Volume=9&Issue=7&Code=ijacsa&{SerialNo}=33.

Saif Mohammad. From once upon a time to happily ever after: Tracking emotions in novels and fairy tales. En *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, págs. 105–114. Association for Computational Linguistics, Portland, OR, USA, 2011. URL https://www.aclweb.org/anthology/W11-1514.

Carlos Molina Beltrán, Alejandra Andrea Segura Navarrete, Christian Vidal-Castro, Clemente Rubio-Manzano, y Claudia Martínez-Araneda. Improving the affective analysis in texts. *Ecos de Linguagem*, 37(6):984–1006, 2019. ISSN 0264-0473. doi:10.1108/EL-11-2018-0219. URL https://www.emerald.com/insight/content/doi/10.1108/EL-11-2018-0219/full/html.

Masanori Morise, Fumiya Yokomori, y Kenji Ozawa. WORLD: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE transactions on information and systems*, E99.D(7):1877–1884, 2016. ISSN 0916-8532. doi:10.1587/transinf.{2015EDP7457}. URL https://www.jstage.jst.go.jp/article/transinf/E99.D/7/E99.{D_2015EDP7457}/_article.

Shane Murnion, William J. Buchanan, Adrian Smales, y Gordon Russell. Machine learning and semantic analysis of in-game chat for cyberbullying. *Computers & Security*, 76:197–213, 2018. ISSN 01674048. doi:10.1016/j.cose.2018.02.016. URL https://linkinghub.elsevier.com/retrieve/pii/S0167404818301597.

Klaus Nordhausen. The elements of statistical learning: data mining, inference, and prediction, second edition by trevor hastie, robert tibshirani, jerome friedman. *International Statistical Review*, 77(3):482–482, 2009. ISSN 03067734. doi:10.1111/j.1751-5823.2009.00095_18.x. URL http://doi.wiley.com/10.1111/j.1751-5823.2009.00095_18.x.

D. Opitz y R. Maclin. Popular ensemble methods: an empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999. ISSN 1076-9757. doi:10.1613/jair.614. URL https://jair.org/index.php/jair/article/view/10239.

Rohit Pawar, Yash Agrawal, Akshay Joshi, Ranadheer Gorrepati, y Rajeev R. Raje. Cyberbullying detection system with multiple server configurations. En *2018 IEEE International Conference on Electro/Information Technology (EIT)*, págs. 0090–0095. IEEE, 2018. ISBN 978-1-5386-5398-2. doi:10.1109/{EIT}.2018.8500110. URL https://ieeexplore.ieee.org/document/8500110/.

ROBERT PLUTCHIK. Chapter 1 - a GENERAL PSYCHOEVOLUTIONARY THEORY OF EMOTION. En Robert Plutchik y Henry Kellerman, eds., *Theories of Emotion*, págs. 3–33. Academic Press, 1980. ISBN 978-0-12-558701-3. URL http://www.sciencedirect.com/science/article/pii/B9780125587013500077.

Michal Ptaszynski, Fumito Masui, Taisei Nitta, Suzuha Hatakeyama, Yasutomo Kimura, Rafal Rzepka, y Kenji Araki. Sustainable cyberbullying detection with category-maximized relevance of harmful phrases and double-filtered automatic optimization. *International Journal of Child-Computer Interaction*, 8:15–30, 2016. ISSN 22128689. doi:10.1016/j.ijcci.2016.07.002. URL http://linkinghub.elsevier.com/retrieve/pii/{S221286891630054X}.

Andrew J. Reagan, Brian Tivnan, Jake Ryland Williams, Christopher M. Danforth, y Peter Sheridan Dodds. Benchmarking sentiment analysis methods for large-scale texts: A case for using continuum-scored words and word shift graphs. 2015.

Ricardo Riquelme. Detección de violencia verbal hacia las mujeres en redes sociales mediante técnicas de aprendizaje automático. 2019.

MGDA Rıos y A Gravano. Spanish DAL: A spanish dictionary of affect in language. *Wassa 2013*, (June):21–28, 2013. URL http://www.aclweb.org/anthology/W/W13/W13-16.pdf#page=33.

A.L Samuel. Some studies in machine learning using the game of checkers. II progress. *Annual Review in Automatic Programming*, 6:1–36, 1969. ISSN 00664138. doi: 10.1016/0066-4138(69)90004-4. URL https://linkinghub.elsevier.com/retrieve/pii/0066413869900044.

Klaus R Scherer. Emotion as a multicomponent process: A model and some cross-cultural data. *Review of Personality & Social Psychology*, 5:37–63, 1984. ISSN 0270-1987(Print).

Alejandra Segura Navarrete, Claudia Martinez-Araneda, Christian Vidal-Castro, y Clemente Rubio-Manzano. A novel approach to the creation of a labelling lexicon for improving emotion analysis in text. *Ecos de Linguagem*, ahead-of-print(ahead-of-print), 2021. ISSN 0264-0473. doi:10.1108/{EL}-04-2020-0110. URL https://doi.org/10.1108/EL-04-2020-0110.

Ahmed Serhrouchni. Multilingual cyberbullying detection system. págs. 1–8, 2014.

Carlo Strapparava y Rada Mihalcea. Learning to identify emotions in text. En *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*, pág. 1556. ACM Press, New York, New York, USA, 2008. ISBN 9781595937537. doi:10.1145/1363686.1364052. URL http://portal.acm.org/citation.cfm?doid=1363686.1364052.

Claudia Nallely Sánchez Gómez. INGEOTEC at MEX-A3T: Author profiling and aggressiveness analysis in twitter using uTC and EvoMSA. CEUR-WS, Seville, Spain, 2018.

Mircea-Adrian Tanase, George-Eduard Zaharia, Dumitru-Clementin Cercel, y Mihai Dascalu. Detecting aggressiveness in mexican spanish social media content by fine-tuning transformer-based models. 2020. URL https://www.facebook.com/communitystandards/hate_speech.

Freddy Tapia, Cristina Aguinaga, y Roger Luje. Detection of behavior patterns through social networks like twitter, using data mining techniques as a method to detect cyberbullying. En *2018 7th International Conference On Software Process Improvement (CIMPS)*, págs. 111–118. IEEE, 2018. ISBN 978-1-7281-0158-3. doi:10.1109/{CIMPS}.2018.8625625. URL https://ieeexplore.ieee.org/document/8625625/.

Nancy Willard. An educator's guide to cyberbullying and cyberthreats. URL www.safestates.org/resource/resmgr/imported/educatorsguide.pdf.

Theresa Wilson, Janyce Wiebe, y Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. En *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*, págs. 347–354. Association for Computational Linguistics, Morristown, NJ, USA, 2005. doi:10.

3115/1220575.1220619. URL http://portal.acm.org/citation.cfm?doid=1220575.1220619.

Ian H. Witten, Eibe Frank, y Mark A. Hall. Credibility. En *Data mining: practical machine learning tools and techniques*, págs. 147–187. Elsevier, 2011. ISBN 9780123748560. doi:10.1016/B978-0-12-374856-0.00005-5. URL https://linkinghub.elsevier.com/retrieve/pii/B9780123748560000055.

Dong Yu y Li Deng. *Automatic Speech Recognition*. Signals and communication technology. Springer London, London, 2015. ISBN 978-1-4471-5778-6. doi:10.1007/978-1-4471-5779-3. URL http://link.springer.com/10.1007/978-1-4471-5779-3.

Harry Zhang. *The Optimality of Naive Bayes*, tomo 2. 2004.

Miguel Á Álvarez Carmona, Estefana Guzmán-Falcón, Manuel Montes-y Gómez, Hugo Jair Escalante, Luis Villasenor-Pineda, Verónica Reyes-Meza, y Antonio Rico-Sulayes. 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*, tomo 6. 2018.

# Anexo A

# Documentación código

## A.1. Código de los experimentos

El código de los experimentos esta disponible en el siguiente repositorio: https://gitlab.com/ManuelLepeF/lexicon_ml_agresividad/. A continuación se presenta una pequeña documentación para su mejor comprensión.

### A.1.1. Requisitos

Para correr el código se debe tener instalado Python 3.8 y virtualenv.

### A.1.2. Instalar ambiente y requerimientos

Para instalar los requerimientos necesarios (librerías y demás) se debe ejecutar los siguientes comandos en la terminal:

1. Crear ambiente:

```
virtualenv ambiente
source ambiente/bin/activate
```

2. Instalar requerimientos:

```
pip install -r requerimientos.txt
```

3. Desempacar NLTK:

```
python
import nltk
nltk.download('punkt')
exit()
```

### A.1.3. Entrenar y probar modelos

Para entrenar y porbar los con los hiperparámetros encontrados con GridSearchCv en cada uno de los corpus se debe tener la siguiente consideración:

```
python Experimentos_CON_WE.py <Nombre modelo> <corpus completo
(ver carpeta corpus)>
```

Ejemplo:

```
python Experimentos_CON_WE.py WE_Lexicon_NB_CU Corpus/CorpusChileno.csv
```

- Experimentos_CON_WE.py : Para los modelos que usan Word Embedding.
- Experimentos_SIN_WE.py : Para los modelos que no usan Word Embedding

Además, se puede utilizar los sh RunExperimentos_Con_WE.sh y RunExperimentos_SIN_WE.sh para ejecutar el entrenamiento y las pruebas de los modelos solo pasando por parametro el nombre del enfoque (No acepta modelos Ensemble)

## A.2. Código de la aplicación web

El código de la aplicación web está disponible en el siguiente repositorio: https://gitlab.com/ManuelLepeF/lexicon_ml_agresividad_web. A continuación se presenta unapequeña documentación para su mejor comprensión.

### A.2.1. Requisitos

Para desplegar la web se debe tener instalado Docker y Docker-compose.

### A.2.2. Desplegar la aplicación web

Para poder despegar la web en un servidor se deben seguir los siguientes pasos:

1. Editar el archivo Frontend/app/.env con la ip donde se montará la API.
2. Ejecutar en la raíz:

   ```
   docker-compose up -d
   ```

3. Insertar resutados en la base de datos usando el archivo Backend/sql_Resultados.sql y pgAdmin4 (ip:5555) con las credenciales definidas en el archivo docker-compose.yml (Para conectarse a la BD se debe usar el host: lexicon_ml_agresividad_web_db_1)

Los datos de la BD quedan guardados en la carpeta /Backend/db_data como se define en el archivo docker-compose.yml.

# Anexo B

# Esquema base de datos

La Figura B.1 presenta el esquema de la base de datos utilizada en la aplicación web.

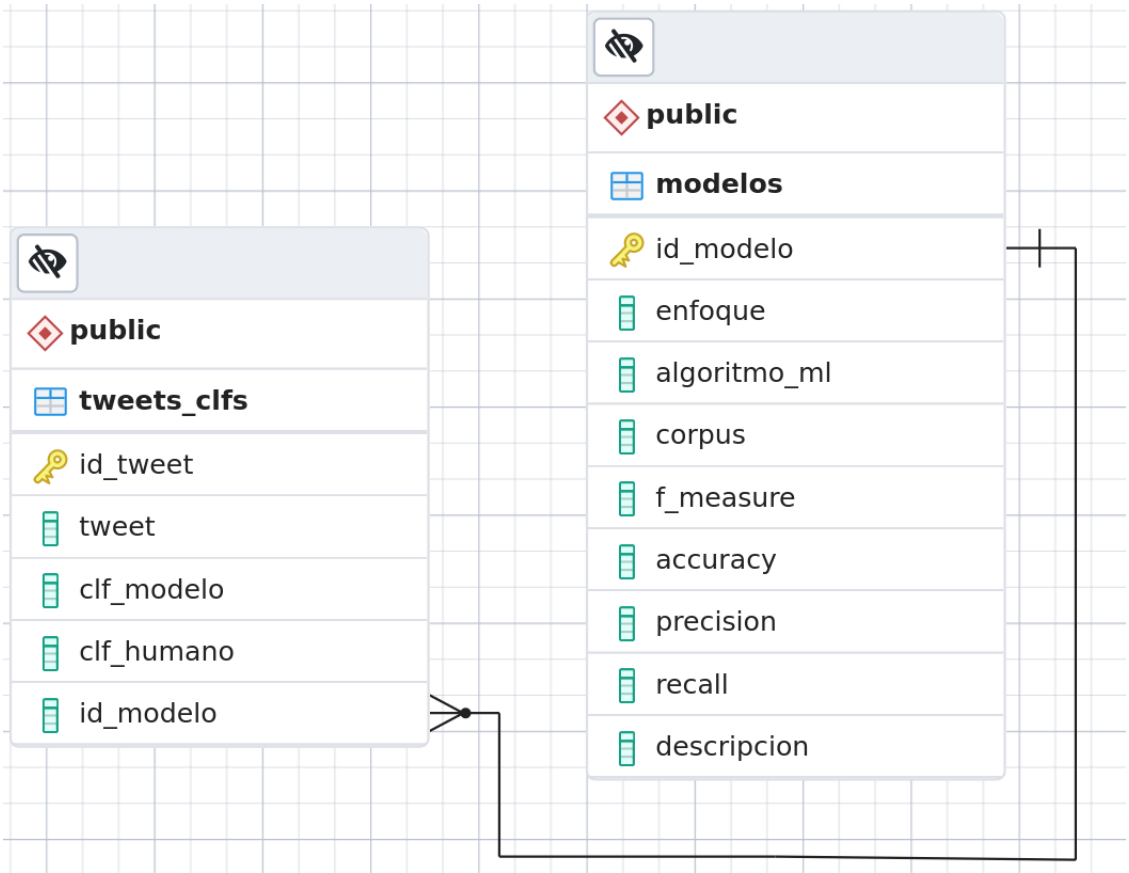


Figura B.1: Modelo relacional base de datos