

## 1. INTRO

El discurso de odio o discriminatorio<sup>1</sup> puede describirse como un discurso en clave de intenso aborrecimiento, denigración y enemistad que ataca a un individuo o un grupo de individuos por poseer –o aparentar poseer– cierta característica protegida por tratados internacionales como el sexo, el género, la etnia, la creencia religiosa, etc. Si bien no hay un consenso generalizado sobre qué configura exactamente discurso de odio [4], un punto de contacto entre las distintas definiciones es que su tendencia a generar un ambiente de hostilidad contra grupos o individuos, incitando a la violencia colectiva contra ellos.

En los últimos años, este tipo de discurso ha tomado gran relevancia en redes sociales y otros medios virtuales debido a su intensidad y a su relación con actos violentos contra miembros de estos grupos. A raíz de esto, estados y organizaciones supranacionales como la Unión Europea han sancionado legislación que insta a las empresas de redes sociales a moderar y eliminar contenido discriminatorio, con particular foco de aquel que incita a la violencia. Para citar un ejemplo, desde 2016 Twitter tiene en sus términos y condiciones:

Conductas de incitación al odio: No se permite fomentar la violencia contra otras personas ni atacarlas o amenazarlas directamente por motivo de su raza, origen étnico, nacionalidad, pertenencia a una casta, orientación sexual, género, identidad de género, afiliación religiosa, edad, discapacidad o enfermedad grave. Tampoco permitimos la existencia de cuentas cuyo objetivo principal sea incitar la violencia contra otras personas en función de las categorías antes mencionadas.

Imágenes y nombres de usuario que incitan al odio: No puedes usar imágenes o símbolos de incitación al odio en la imagen o el encabezado de tu perfil. Tampoco puedes usar tu nombre de usuario, nombre visible o biografía de perfil para participar en comportamientos abusivos, como realizar acosos dirigidos o expresar odio contra una persona, un grupo o una categoría protegida. (Política relativa a las conductas de incitación al odio, Twitter)

La enorme cantidad de texto generado por usuarios en las redes sociales –alrededor de los 500M de tweets por día a nivel mundial– hace<sup>1</sup> imposible que el análisis de este contenido sea realizado de manera enteramente manual. En este escenario de creciente preocupación que genera la proliferación de este discurso, se hace necesario el desarrollo de herramientas que automaticen la detección de discurso de odio en redes sociales, bien sea para el estudio y monitoreo o bien para la moderación.

Desde el procesamiento de lenguaje natural, la detección de discurso de odio puede entenderse en su forma más básica como un problema de clasificación de texto: dado un texto generado por un usuario, predecir si es o no contenido discriminatorio. Así mismo, puede ser de interés predecir otras características: por ejemplo, si el texto contiene un llamado a la acción violenta, si está dirigido contra un individuo o un grupo, el tipo de característica ofendida, entre otras. Poder identificar estas características puede ayudar a delimitar las formas más peligrosas de este fenómeno, como incitaciones a la violencia contra un grupo o individuo.

<sup>1</sup> Usaremos intercambiablemente estas expresiones. Para una discusión sobre sus diferencias, ver la sección 4.1

Así mismo  
también  
que a  
más adelante

## 1.1. Algunos casos resonantes

Para hacer énfasis en la necesidad de desarrollar herramientas que puedan ayudar a la detección de contenido discriminatorio, comentaremos algunos casos puntuales que han tenido lugar en los últimos años, en los cuales han co-ocurrido<sup>2</sup> picos de discurso de odio en redes sociales –mayormente racista o xenófoba– con eventos de extrema violencia en la vida real. Aún cuando estos ejemplos relatan escenarios en sus formas más brutales, la mera exposición a este discurso en medios virtuales tiene impactos negativos en la psiquis de sus objetivos [129] y prepara un terreno hostil y de deshumanización contra un grupo vulnerado, como inmigrantes, minorías religiosas y sexuales [15], algo que ya ha sido estudiado a lo largo de décadas antes de la aparición de las redes sociales e Internet.

### 1.1.1. Atentados en Charlottesville

En Agosto del 2017, una gran movilización organizada por varios movimientos de ultraderecha y supremacistas blancos tuvo lugar en la ciudad de Charlottesville, Virginia, Estados Unidos. Esta concentración fue llamada en el medio del intento de universitarios y el movimiento Black Lives Matter (BLM) de remover estatuas de militares conferados pro-esclavitud de la Guerra de Secesión a lo largo de todo el territorio de Estados Unidos. En este caso puntual, se intentaba remover la estatua de Robert Lee ubicada en el campus de la Universidad de Virginia, durante los primeros meses de mandato de Donald Trump.

Numerosos grupos de ultraderecha, neonazis, neo-confederados, entre otros, convocaron a la marcha “Unite the Right”(UtR), diseñada como una campaña militar y organizada durante varios meses antes de su concreción. Blout and Burkart [18] describen la experiencia de Charlottesville como la de un “terrorismo inmersivo” ya que generaron un ámbito de terror en varios “teatros” (como lo llaman los autores, usando jerga militar). Principalmente, el teatro físico, con la marcha y enfrentamientos con las contra-movilizaciones, la intimidante marcha de antorchas, y el asesinato de Heather Heyer atropellada por un manifestante neo-nazi. Así mismo, el teatro “virtual” situado en las redes sociales, sirvió para generar un clima de intimidación antes, durante, y luego del evento mencionado, como una campaña judeófoba contra el alcalde de Charlottesville, de ascendencia judía, y el vicemayor, de ascendencia afroamericana.

Blout and Burkart [18] llegan a la conclusión de que el evento fue organizado de manera centralizada, tanto en su planificación como despliegue en un intento de ejercicio militar. También concluyen que la propaganda y la información diseminada por los organizadores sirvió para publicitar y reclutar a simpatizantes como también para aterrorizar a la población de Charlottesville. Esta propaganda se difundió tanto por medios impresos (por ejemplo, posters pegados en las calles) como por redes sociales como Facebook, Twitter o Discord. Klein [75] analiza los intercambios en Twitter entre los dos bandos (manifestantes de ultraderecha y los contramanifestantes) y muestra que, en el caso de quienes se encontraban del lado de la marcha de UtR, se identifica como enemigos a los musulmanes, liberales o izquierdistas, a miembros de la comunidad LGBTQ, judíos, entre otros, dando cuenta del sesgo discriminatorio de este grupo.

Sobre? —

re pag. 8



Fig. 1.1: Último post de Robert Bowers, tirador en la masacre de Pittsburgh, en la red social Gab.

### 1.1.2. Matanza en Sinagoga de Pittsburgh

En Octubre de 2018, un hombre fuertemente armado entró a la sinagoga “El Árbol de la Vida” en Pittsburgh, Pensilvania, Estados Unidos. Luego de gritar “muerte a los judíos”, abrió fuego contra la multitud matando 11 personas y dejando decenas de heridos, la matanza más grande de judíos en EEUU de la que se tenga registro.

El tirador, Richard Bowers, era usuario activo de Gab<sup>3</sup>, una red social que nació en 2016 bajo la égida de la defensa de la “libertad de expresión” a raíz de la creciente moderación de Twitter y Facebook a discursos discriminatorios. Desde entonces, ha sido el refugio de activistas de la derecha alternativa, supremacistas raciales, grupos conspiracionistas y otros elementos reaccionarios. El asesino en cuestión posteaba frecuentemente contenido antisemita en dicha red social [93], particularmente contra la HIAS (Sociedad Hebrea de auxilio de inmigrantes). En su último post en dicha red social, horas antes de la masacre, Bowers postcó una amenaza (ver figura 1.1) diciendo que no podía tolerar ver a su gente ser asesinada (por judíos) y que iba a tomar acciones al respecto.

A raíz de esto, Gab –llamada popularmente como el “Twitter racista”– estuvo de baja durante cierto tiempo al serle negado alojamiento web debido a este atentado. Desde entonces, diversos trabajos han recopilado y analizado el contenido discriminatorio en esta red social [72, 93].

### 1.1.3. Masacre Rohingya en Myanmar

Entre 2016 y 2017, fue perpetrada una matanza de la etnia Rohingya, un grupo étnico musulmán, en la República de Myanmar (ex Birmania). Cerca de 25 mil personas fueron masacradas y un éxodo de más de 700 mil personas tuvo lugar hacia la lindante Bangladesh, conformando el campamento de refugiados más grande del mundo en la actualidad. La ONU y algunos estados nacionales han calificado lo ocurrido como un “genocidio” y como una “limpieza étnica”.

Si bien el sometimiento de este pueblo tiene lugar hace décadas, en los últimos años

<sup>2</sup> Nótese que utilizamos la palabra co-ocurrir y no causar

<sup>3</sup> <https://gab.com/>

por qué oca?  
debería en la  
página anterior

tuvo un gran recrudecimiento motorizado desde las altas esferas gubernamentales y militares birmanas, que niegan cualquier estatus legal a la población rohingya. En ese punto, las redes sociales han jugado un rol de difusor y catalizador de incitaciones a la violencia y noticias falsas alrededor de esta etnia. Según un informe solicitado por Facebook acerca de la situación en Myanmar [145], gran parte de este problema se debe a un déficit en el “alfabetismo digital”(sic) de la población de este país, que usa casi exclusivamente Internet a través de dicha red social. Enviados de las Naciones Unidas han acusado directamente a Facebook de haber servido como intermediario de discurso de odio a través de su plataforma<sup>4</sup>, y que ha tenido un “rol determinante” en este genocidio.

Grupos de derechos humanos de ese país han instado a la empresa de Mark Zuckerberg a invertir recursos en el control del discurso de odio, particularmente aquel que insta a la violencia física [104]. A finales de 2021, un grupo de refugiados rohingya denunció a Facebook por 150 mil millones de dólares<sup>5</sup> por haber promovido la violencia contra este grupo, luego de que en 2018 responsables de la empresa admitieran que no se hizo lo suficiente para detener la proliferación del discurso xenófobo contra esa minoría.

Este hecho cuenta con una particularidad: apunta a un idioma –el birmano, idioma oficial en Myanmar– que dispone de pocos recursos en el área del Procesamiento del Lenguaje Natural. La mayoría de los recursos y estudios están dedicados al idioma inglés, ignorando las particularidades de cada idioma y el componente cultural de algunas tareas, como en este caso la detección de discurso de odio. Además, según Reuters para finales de 2018, Facebook no contaba con ningún empleado en Myanmar<sup>6</sup> ni tampoco quedaba claro que alguno de sus empleados dedicados a la tarea del monitoreo sea hablante nativo de birmano.

## 1.2. Avances en IA y NLP

En los últimos 10 años, el área de la Inteligencia Artificial ha sido sacudida por la irrupción de las redes neuronales. Desde el campo de Visión por Computadora, un conjunto de factores han potenciado el éxito de esta técnica de aprendizaje estadístico: datasets de gran tamaño como ImageNet [34], la utilización de dispositivos de gran poder de cómputo como las GPUs, y el desarrollo de mejores algoritmos para su entrenamiento (de optimización, funciones de activación, entre otras cosas). Esta combinación posibilitó que las redes neuronales obtengan mejoras considerables en el desempeño de tareas de reconocimiento de imágenes, trasladándose esto a otras áreas como procesamiento de habla, y a todas las áreas de aprendizaje automático en general, con particular foco de aquellos datos no estructurados como imágenes, sonido, y otras señales.

Este boom inicial tuvo su primera repercusión de magnitud en NLP cerca del año 2013 con el desarrollo de los word-embeddings. La técnica de *word2vec* [96] permitió generar representaciones de palabras de manera eficiente sobre grandes cantidades de datos no etiquetados. Estas representaciones de las palabras (podemos pensarlas como vectores de largo fijo asignadas a cada token) han sido la “salsa secreta” que permitió el éxito de las redes neuronales en NLP, permitiendo una mejora en las tareas de reconocimiento de entidades nombradas (NER), POS tagging, parsing, clasificación de textos, entre otras. Otro componente de este éxito de las redes neuronales ha sido el uso de redes recurrentes

<sup>4</sup> <https://www.reuters.com/article/us-myanmar-rohingya-facebook/u-n-investigators-cite-facebook-role-in-mya>

<sup>5</sup> <https://www.bbc.com/news/world-asia-59558090>

<sup>6</sup> <https://www.reuters.com/investigates/special-report/myanmar-facebook-hate/>

como las Long Short-Term Memory (LSTM) [64] o las Gated Recurrent Units (GRU) [27], que permiten codificar secuencias de manera autorregresiva. Un caso de éxito particular utilizando estas redes recurrentes ha sido el de la traducción automática mediante la arquitectura sequence-to-sequence (seq2seq) [139]. Estas redes permitieron atacar los problemas de aprendizaje de secuencia a secuencia, como la traducción automática o summarización de texto, reemplazando sistemas realmente complejos y de difícil mantenimiento (como los de Statistical Machine Translation) por diseños más simples y con una muy superior performance.

En 2017, Vaswani et al. [142] propusieron una arquitectura que elimina la estructura recurrente: los *Transformers*. Este modelo utiliza únicamente múltiples capas de autoatención para el problema de traducción automática. Al eliminar los pasos recurrentes, permitió la parallelización del cálculo y el entrenamiento de arquitecturas verdaderamente profundas para el área de NLP, como ya hace tiempo se utilizaban en el área de Visión por Computadora. En conjunto a la aplicación del pre-entrenamiento utilizando la tarea de modelado de lenguaje (que introdujeron Howard and Ruder [67] con ULMFiT, entre otros trabajos) supusieron un cambio rotundo en el modo en que hacemos NLP hoy día: en lugar de entrenar una red neuronal casi desde cero (quizás sólo con una capa de embeddings con pesos iniciales pre-calculados) sólo ajustamos (*fine-tune*) una gran red neuronal pre-entrenada sobre un dataset de entrenamiento con alguna tarea de modelado de lenguaje. *BERT*, *GPT* y otros personajes de Plaza Sésamo son algunos de los rutilantes nombres en el zoológico de modelos pre-entrenados. Esta nueva forma de atacar los problemas de aprendizaje automático de NLP supusieron un breakthrough en el área, mejorando la performance sensiblemente en benchmarks de tareas como GLUE [143], RACE [79], entre otras tareas.

Estos avances han permitido atacar numerosas tareas que quizás parecían fuera de alcance para el área de NLP o bien tenían performances relativamente pobres. Una de estas tareas es la detección de discurso de odio en redes sociales.

*desempeños*

A esto le falta un poquito de desarrollo S.

(último parrafo)

### 1.2.1. Asimetría de recursos

Un problema no menor en el área de NLP es la enorme asimetría de recursos entre idiomas. La inmensa mayoría de recursos –tanto en forma de datasets como modelos– están diseñados para el inglés. Con el advenimiento de los modelos pre-entrenados basados en *Transformers*, este problema se agravó ya que estos modelos necesitan muchos recursos computacionales para ser generados. Para atacar este problema, es necesario desarrollar recursos y estudios para los distintos idiomas. Citando a la “Regla de Bender” *“M”*:

“Do state the name of the language that is being studied, even if it’s English. Acknowledging that we are working on a particular language foregrounds the possibility that the techniques may in fact be language specific. Conversely, neglecting to state that the particular data used were in, say, English, gives [a] false veneer of language-independence to the work.” *[M]*

Si bien el español puede considerarse de los idiomas dentro del grupo de los de “altos recursos” [10], aún así la disparidad de recursos al ~~al~~ inglés es abrumadora. En particular, para el área de interés de esta tesis –la detección de discurso de odio–, los recursos son muy escasos y, en la mayoría de los casos, “réplicas” de trabajos hechos en inglés.

*Con el*



Fig. 1.2: Tweets y respuestas discriminatorias. Leyendo únicamente el texto de los comentarios resulta difícil descifrar su sentido.

### 1.3. Detección de discurso

Como dijimos anteriormente, la detección del discurso de odio puede pensarse como una tarea de clasificación sobre un texto generado por un usuario. Muchos trabajos en los últimos años han abordado la tarea desde esa perspectiva, desarrollándose numerosos recursos en workshops para varios idiomas, y herramientas muy utilizadas para la moderación de contenido tóxico<sup>7</sup> como Perspective API<sup>8</sup>, desarrollada por Jigsaw y Google.

Si bien ~~esta~~ este enfoque tiene una innegable utilidad, adolece de ciertas limitaciones. Uno de sus problemas es la falta de contexto en los mensajes analizados. Los seres humanos no solo recibimos mensajes de manera aislada sino que los entendemos ~~a estos~~ situados de acuerdo a varios factores: su emisor, la situación y el medio en el que se lo emite, a quién está dirigido, sobre qué hace mención, entre otras cosas. La mayoría de los estudios y recursos, sin embargo, están hechos sobre datos fuera de contexto: posteos en redes sociales sin ningún tipo de información conversacional o de otra índole. Para ilustrar el problema de la falta de contexto, la figura 1.2 muestra un tweet de un medio periodístico que habla sobre una actriz y respuestas a esa noticia<sup>9</sup>. Leyendo los tweets por fuera del contexto es difícil comprender el mensaje y su contenido transfóbico.

La falta de contexto restringe la información disponible –tanto para un humano como para un algoritmo– para poder discernir si un texto social es discriminatorio. Otra información usualmente faltante y que puede ayudar a enriquecer la detección de discurso de odio es la característica atacada en un texto: es común que los datasets estén anotados de manera poco granular –casi siempre de manera binaria– no brindando información acerca de si la agresión es por motivos de género, religión, etnia, etc.

Por último, una limitación puntual del español es la poca disponibilidad de recursos para esta tarea, algo que mencionamos en la anterior sección como un problema general de

<sup>7</sup> El contenido tóxico o abusivo es una categoría un poco más general que la de discurso de odio

<sup>8</sup> <https://www.perspectiveapi.com/>

<sup>9</sup> El hilo completo puede encontrarse en <https://twitter.com/infobae/status/1242506130213015552>

No me queda  
claro wá! es  
'interfogue'.

NLP. A este problema se le suma que los pocos datasets disponibles tanto para español como otros idiomas—suelen estar generados por anotadores que no son hablantes de las variedades dialectales de los textos utilizados, lo cual genera un déficit en su calidad al ser el lenguaje discriminatorio altamente dependiente de la jerga específica de cada región y de su contexto sociocultural.

#### 1.4. Aportes de este trabajo

En esta tesis nos proponemos hacer un aporte en el sentido de desarrollar mejores mecanismos automáticos de detección de discurso de odio. Si bien el área de NLP ha avanzado enormemente en los últimos años —y esta subdisciplina en particular ha recibido un gran interés— creemos que muchos de los enfoques actuales inhiben un avance cualitativo en la detección de este pernicioso fenómeno en medios sociales.

Para ello, en primer lugar estudiamos técnicas de detección sobre datasets ya existentes, utilizando técnicas del estado del arte. En base a la observación de algunos datasets y la literatura en general, plantearemos un nuevo problema: la detección *contextualizada* de discurso de odio. Para ello, construimos un corpus de discurso de odio sobre comentarios en noticias de medios gráficos argentinos en Twitter, siendo este conjunto de datos etiquetado por hablantes nativos. Este dataset es un aporte importante en sí ya que es uno de los primeros datasets que incluyen información contextual, y es el único a nuestro conocimiento en español que tiene esta información. Otras características que lo distinguen es que es uno de los primeros datasets de la variedad dialectal rioplatense a nuestro conocimiento recolectado durante la pandemia de COVID-19.

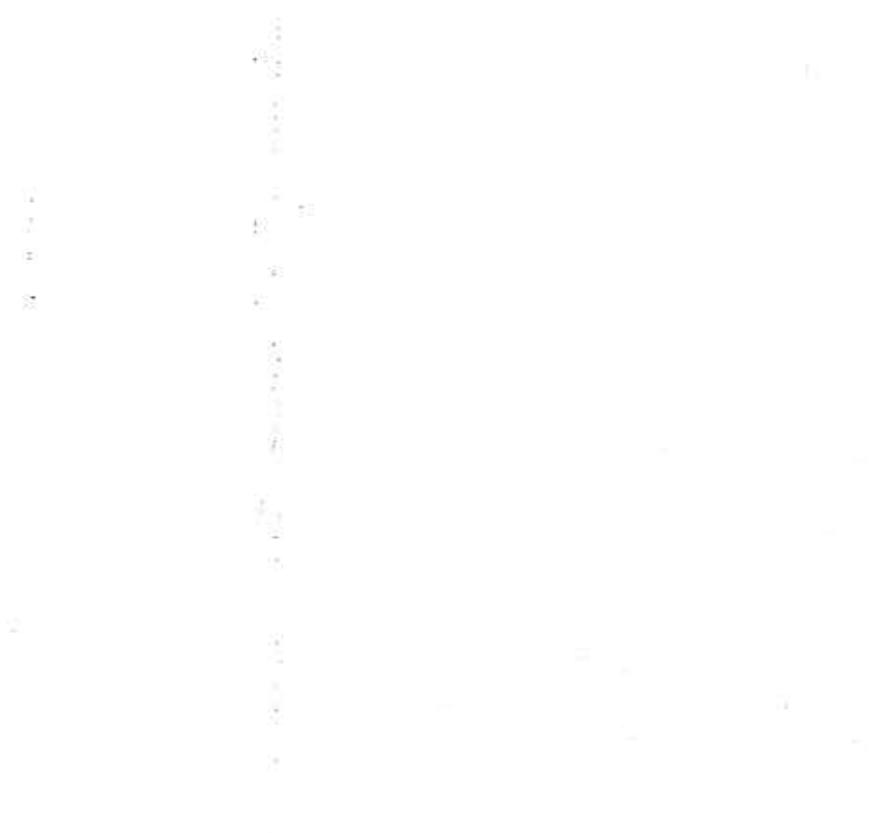
Con este recurso, exploramos la siguiente pregunta: ¿pueden los métodos actuales basados en modelos pre-entrenados aprovechar información adicional de contexto para mejorar la detección de discurso de odio? Este punto ha sido poco estudiado en la literatura y consideramos que es una pregunta de interés para atravesar los límites de la clasificación basada en una única fuente de información (el comentario analizado). En base a los experimentos realizados, encontramos evidencia que el contexto puede brindar información útil para detectar este fenómeno. Particularmente, observamos que para los mensajes de odio contra ciertos grupos —por ejemplo, contra la comunidad LGBTI— el contexto puede ser aún más útil para su detección.

Finalmente, realizamos un estudio más en general sobre la *adaptación de dominio* en tareas de clasificación de redes sociales. Para ello, generamos un modelo de lenguaje pre-entrenado sobre textos sociales en español al que bautizamos *RoBERTuito*, el primero disponible y a gran escala en este idioma. Comparamos la performance de este modelo contra otros modelos pre-entrenados sobre textos formales, y contra modelos ajustados al dominio social. Esta comparación es de interés ya que el ajuste de dominio es relativamente económico frente al enorme costo de entrenamiento que tiene construir modelos como *RoBERTuito*. Observamos que para todas las tareas, *RoBERTuito* obtiene una performance del estado del arte, pero el ajuste de dominio recorta considerablemente el gap de performance contra otros modelos.

Un aporte en general de esta tesis es que todos los estudios y recursos han sido realizados en español. Vista la enorme asimetría que hay con otros idiomas, y teniendo en cuenta que el español es el segundo idioma en hablantes nativos del mundo, consideramos necesario mitigar este desbalance de recursos.

¿Por qué es relevante esto?

ver pág. 4



## 2. PRELIMINARES

En esta sección haremos una breve introducción a técnicas de Machine Learning y NLP. Describiremos las principales técnicas de clasificación utilizadas durante los últimos años en el área, las cuales usaremos en las secciones subsiguientes.

### 2.1. Redes Neuronales

#### 2.1.1. Multi-layer perceptron

Los perceptrones multi-capa (MLP) o redes feed-forward (FFN) son la “quintesencia” de los métodos modernos de Deep Learning, como describen Goodfellow et al. [53]. La idea detrás de estos modelos es encontrar una función  $f : \mathbb{R}^m \rightarrow O$  que aproxime a otra  $f^*$  con misma aridad. Podremos notar  $y = f(x; \Theta)$ , siendo  $\Theta$  los parámetros de dicha función. Uno de los primeros acercamientos a este problema es la neurona de McCulloch-Pitts [92], que intenta modelar parte del funcionamiento de las neuronas mediante una función:

$$y = H(\Theta^T x)$$

← donde  $H$  es la función de Heaviside o función escalón. Esta función permite aproximar a dos valores (0 ó 1) a partir de una entrada  $x$  y un parámetro  $\Theta$ . El perceptrón, desarrollado en 1958 en Rosenblatt [126], es el primer modelo que utiliza este tipo de modelo de cómputo cuyos parámetros  $\Theta$  se encontraban mediante un algoritmo. Minsky and Papert [97] demostraron que este tipo de modelos sólo pueden ajustarse a datos linealmente separables, provocando el primer “invierno” de las redes neuronales.

Una forma de sortear estas dificultades planteadas es “apilar” (stack en inglés) varias de estas funciones para poder ajustar a más tipos de funciones. En términos matemáticos, esto es tan sólo una composición de funciones, tomando ahora  $f = f_3 \circ f_2 \circ f_1$ , donde  $f_1$  es la primera “capa” de nuestra función correspondiente a la entrada,  $f_2$  es la capa intermedia u oculta, y  $f_3$  es la capa de salida. Si bien este ejemplo consta de 3 capas, se puede generalizar a arbitrarias capas ocultas. Este modelo es el que conocemos como Perceptrón Multicapa o Multi-Layer Perceptron (MLP por sus siglas en inglés), y provocó el resurgir conexionista de las redes neuronales en los años 80s mediante el desarrollo de algoritmos que permitieron entrenar estos modelos mediante backpropagation [128].

#### 2.1.2. Redes neuronales recurrentes

Los problemas descriptos de NLP suelen constar de procesar una secuencia de palabras o tokens  $x_1, x_2, \dots, x_k$  de longitud variable, de manera de ajustar a una función

$$y = f([x_1, \dots, x_k])$$

Una manera de abordar una función de este tipo (usando una entrada de largo fijo) es convertir este problema a ajustar una función autorregresiva

$$y_k = f(x_k, y_{k-1})$$

donde tenemos una salida para cada paso  $k$  de tiempo. Si  $f$  es una red neuronal, llamamos a este tipo de redes neuronales recurrentes, ya que la salida a cada paso ( $y_k$ ) depende de la salida del paso anterior,  $y_{k-1}$ <sup>1</sup>.

Una primera aproximación a este problema es la red recurrente de Elman [38] definida por las siguientes ecuaciones

$$h_t = \sigma(W_h x_t + U_h h_{t-1} + b_h) \quad (2.1)$$

$$y_t = \sigma(W_y h_t + b_y) \quad (2.2)$$

$h_t$  es normalmente llamado el **estado oculto** en las redes neuronales recurrentes. Los parámetros a ajustar son  $W_h, U_h$  (matrices) y  $b_h, b_y$  (escalares). Podemos ver que, a grandes rasgos, este tipo de red recurrente no es más que un perceptrón multicapa cuya entrada consta de  $x_t$ , la entrada original en el tiempo actual  $t$ , y el estado oculto anterior,  $h_{t-1}$ .

Para entrenar este tipo de redes recurrentes utilizamos back-propagation through time (BPTT), que consta en desplegar la relación recurrente y aplicar back-propagation de manera normal. Este tipo de redes recurrentes sufren de varios problemas: entre ellos, **vanishing gradient** y **exploding gradient**. Estos problemas pueden observarse ya que el cálculo del gradiente de las ecuaciones 2.2 usando BPTT induce la potencia a la  $n$  (donde  $n$  es el largo de la secuencia) de las matrices  $W_h$  y  $U_h$ . Usando alguna descomposición de la matriz en valores singulares (como la forma normal de Jordan) podemos ver que sus elementos en la diagonal que sean distintos de 1, o bien tienden a infinito o a cero.

El problema de **exploding gradient** puede solucionarse mediante la técnica de **gradient clipping**, que consta de reajustar la norma del gradiente. Sin embargo, nos queda aún el problema de **vanishing gradient**. Para ello, se han propuesto otras arquitecturas recurrentes. Hochreiter and Schmidhuber [64] propusieron las **Long Short-Term Memory (LSTM)** como solución a estos problemas. Para solucionar los problemas mencionados, proponen una arquitectura basadas en compuertas (gates) que regulan los cambios en el estado oculto y en la salida. La arquitectura de las LSTMs<sup>2</sup> modifica la arquitectura de Elman mediante la inserción de compuertas para la entrada y salida, evitando algunos de los problemas antes mencionados. Otras arquitecturas como las Gated Recurrent Units [27] usan menor cantidad de compuertas reduciendo la cantidad de parámetros a entrenar.

## 2.2. Técnicas de representación

Una de las necesidades que tienen las redes neuronales para poder trabajar con textos es el de tener representaciones continuas de cada token o palabra. Las representaciones utilizadas en la época previa de los modelos lineales –bolsas de palabras/caracteres ponderadas con esquemas como TF/IDF– adolecen de varios problemas: tienen una altísima dimensionalidad; no tienen representación semántica de la similaridad de las palabras; están concentradas en una o pocas dimensiones y suelen ser discretas.

<sup>1</sup> No confundir con las redes neuronales recursivas

<sup>2</sup> Para una muy buena explicación de las LSTMs sugerimos este artículo: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

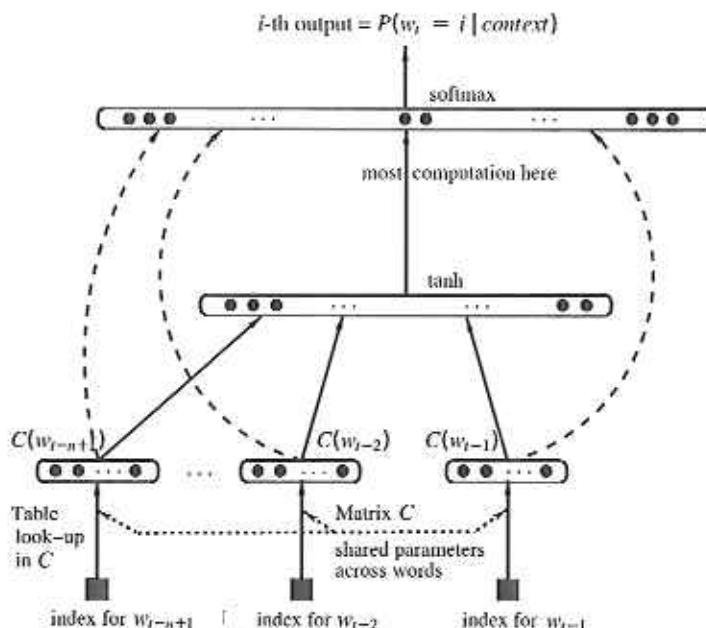


Fig. 2.1: Ilustración del modelo de lenguaje neuronal de Bengio et al. [13]. La entrada consta de  $n - 1$  palabras, que primero pasan por una lookup table (o capa de embeddings), una función de activación y son colapsadas para luego ser utilizadas como entrada de una función softmax.

Latent Semantic Analysis (LSA) [81] es una de las primeras técnicas populares de representación continua utilizadas. Plantearon el problema de obtener representaciones continuas como el de factorizar una matriz de co-ocurrencia entre tokens y documentos (o contextos). LDA (Latent Dirichlet Allocation) [16] es otra técnica basada en modelos gráficos entrenados mediante métodos variacionales, muy utilizada aún en la actualidad ya que genera representaciones latentes de los tópicos de los textos.

Dentro de los métodos neuronales, uno de los más populares ha sido el de Bengio et al. [13], que propone una arquitectura neuronal para un modelo de lenguaje markoviano. La arquitectura de esta red está ilustrada en la figura 2.1. En la capa intermedia contiene una tabla de lookup de vectores de las diferentes palabras (también conocido como capa de embeddings) donde se generan las representaciones de las palabras. Trabajo posterior (con diferentes variaciones de esta misma idea) como el de Collobert et al. [30] ha demostrado que la utilización de este tipo de representaciones es útil para diversas tareas de NLP como POS Tagging, NER, y otras. Más aún, este trabajo tiene una idea que fue utilizada muchos años después con éxito rotundo: la utilización de la tarea de modelado de lenguaje como base para el pre-entrenamiento de los modelos.

Uno de los problemas de los métodos vistos hasta el momento es que sufrían problemas de eficiencia, sólo pudiéndose entrenar con pocos millones de palabras y con dimensiones reducidas. La técnica *word2vec* [95] permite entrenar representaciones de palabras de mayor dimensión y sobre grandes cantidades de textos de manera eficiente. Los vectores de palabras guardan cierta estructura lineal y semántica, como ilustran los autores con algunos ejemplos, como el ya clásico  $v(\text{rey}) - v(\text{hombre}) + v(\text{mujer}) \approx v(\text{reina})$ .

Para generar los vectores de *word2vec*, los autores plantean una relajación del problema

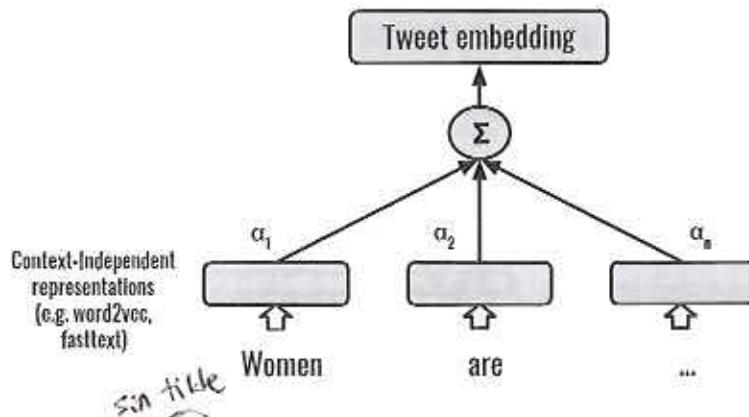


Fig. 2.2: Representación continua de un tweet mediante combinación lineal de las representaciones de cada palabra.

de modelado de lenguaje mediante dos alternativas: Continuous Bag of Words (CBOW) y Skip-Gram. En CBOW intentamos predecir la palabra faltante dada una bolsa de palabras del contexto, y en skip-gram intentamos predecir las palabras del contexto dada la palabra central. Para ambos problemas, se generan representaciones intermedias ricas para las distintas palabras. Mikolov et al. [95] extiende la idea del anterior trabajo proponiendo plantear el problema de skip-gram como uno de distinguir palabras ruido de palabras efectivamente del contexto, haciendo mucho más eficiente el cálculo de estas representaciones. GloVe [112] es otra técnica de representación de palabras que combina las ideas de factorización de matrices de LSA mediante un problema de optimización distinto y generando representaciones que superan ligeramente en algunos benchmarks de tareas a los de *word2vec*.

Los métodos mencionados de representación calculan vectores de tamaño fijo sobre cada una de las distintas palabras. En español, por ejemplo, las palabras gato, gata, gatito, gatuno, todas tienen representaciones independientes en *word2vec*, a pesar de tener información morfológica en común. Esto es un problema en varios escenarios: idiomas con muchas inflexiones o aglutinantes (como el turco, alemán o finés) o –lo que es de nuestro interés– texto altamente desnormalizado como el de redes sociales. La técnica *fastText* [19] extiende la idea de *word2vec* mediante la asignación de vectores a secuencias de 3 caracteres (subpalabras), capturando así cierta información morfológica. La representación de una palabra se obtiene mediante una combinación lineal de los vectores de las subpalabras que la componen.

### 2.2.1. Embeddings a nivel documento

Una forma relativamente simple de obtener una representación de un documento<sup>3</sup> es realizar una combinación lineal de las representaciones obtenidas para cada palabra. Es decir, dada una oración  $s = w_1 w_2 \dots w_n$ , y representaciones  $\bar{w}_1, \bar{w}_2, \dots, \bar{w}_n \in \mathbb{R}^m$ , podemos

<sup>3</sup> en nuestra tesis, esto será casi siempre un tweet

obtener una representación

$$\bar{s} = \sum_{i=1}^n \alpha_i \bar{w}_i \quad (2.3)$$

con  $\alpha_1, \dots, \alpha_n \in \mathbb{R}$  escalares (dependientes de la oración). De esta manera, obtenemos de  $n$  representaciones independientes del contexto una representación para el tweet, sin tener en cuenta posibles interacciones entre los distintos componentes. La figura 2.2 ilustra esta metodología simple para obtener representaciones de oraciones.

Tenemos entonces dos posibilidades para determinar la combinación lineal: la forma de obtener las representaciones, y la forma de calcular los coeficientes. Para las representaciones, podemos usar varias de las técnicas que ya vimos como word2vec, GloVe, o fastText. Para calcular los coeficientes, consideramos en nuestro trabajo dos formas. La primera, la forma canónica, calculando un promedio de las representaciones, es decir, tomando  $\alpha_i = \frac{1}{n}$ . Otra es realizar una ponderación distinta, como por ejemplo el de frecuencia inversa suave (SIF) [3], inspirado en TF-IDF. Cada palabra  $w$  se pondera con  $\frac{a}{a+p(w)}$ , donde  $p(w)$  es la palabra probabilidad unigrama y  $a$  es un hiperparámetro de suavizado. Los valores altos de  $a$  significan más suavizado hacia el promedio simple.

## 2.3. Transfer Learning y modelos pre-entrenados

### 2.3.1. ELMo y ULMFiT

Hasta cerca de 2018, la forma canónica de abordar un problema de NLP era entrenar una red neuronal recurrente que consumiera embeddings no contextualizados de los tokens de entrada. Esta arquitectura tiene algunas limitaciones; una de ellas es que, dados dos o más problemas distintos (por ejemplo, análisis de sentimientos e inferencia de lenguaje natural –NLI–) lo único compartido por ambas redes es la capa más baja –la capa de embeddings– teniendo que entrenar desde cero todo el resto de los parámetros del modelo. En términos coloquiales, cada red debe “aprender a leer” sobre cada tarea, ignorando muchas construcciones sintácticas y semánticas comunes del lenguaje.

Uno de los primeros esfuerzos exitosos en superar los embeddings no contextualizados es ELMo [116]. Este modelo aprende embeddings ya no sobre una única palabra como word2vec sino sobre toda una oración, generando representaciones contextualizadas de cada palabra. Para aprenderlas, ELMo se entrena sobre un modelo de lenguaje bidireccional<sup>4</sup> recurrente de varias capas sobre grandes cantidades de texto. En dicho trabajo, utilizan luego una combinación lineal de la salida de cada capa para obtener representaciones contextualizadas de cada token. Esta misma idea es una continuación de Peters et al. [115], y también parcialmente de McCann et al. [91]; en este último trabajo abordan la construcción de representaciones contextualizadas mediante la tarea de traducción automática.

Alrededor de 2018, este paradigma de entrenar una red desde cero compartiendo su capa más baja –word2vec o bien ELMo– comenzó a cambiar hacia un esquema donde se entrena una red neuronal sobre una tarea genérica para luego ajustarla a la tarea específica, una práctica muy común en el área de Computer Vision. Howard and Ruder [67] introdujeron la técnica de ULMFiT (Universal Language Modeling for Fine-tuning for text classification), uno de los trabajos fundamentales de este nuevo enfoque en NLP. ULMFiT

<sup>4</sup> En realidad no es estrictamente bidireccional, sino dos modelos de lenguaje concatenados

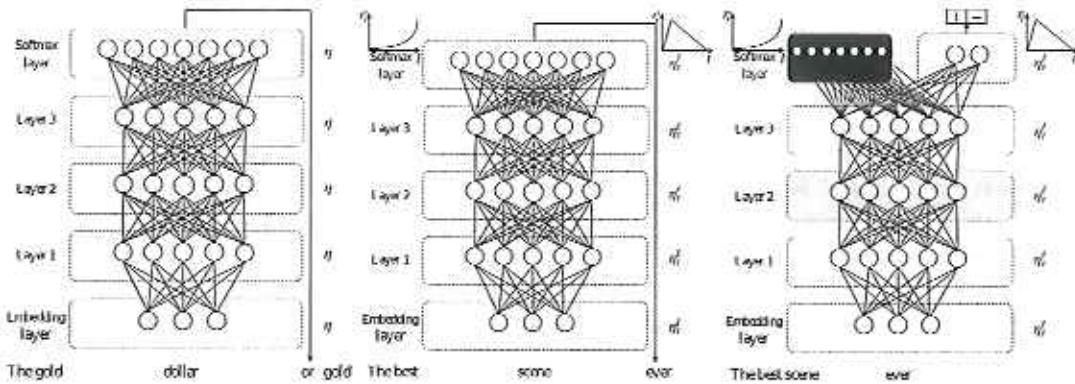


Fig. 2.3: Universal Language Modeling for Text Classification (ULMFiT). Esquema del método planteado: primero se pre-entrena sobre la tarea de modelado de lenguaje sobre un dataset no supervisado. Luego, se corre la misma tarea pero sobre el texto de la tarea ignorando las etiquetas (ajuste de dominio). Finalmente, se agrega una capa de parámetros particulares de la tarea y se entrena la red completa descongelando gradualmente. Fuente: Howard and Ruder [67]

? entrando ?

consta de pre-entrenar en primer lugar un modelo de lenguaje sobre un gran dataset no etiquetado, y luego utilizar esa misma red (cambiándole la última capa) ajustándola a una tarea específica. El primer paso, el **pre-entrenamiento** es realizado una única vez, y sus pesos son luego utilizados para realizar el ajuste en cada tarea distinta. Este es uno de los primeros esquemas **transfer learning** exitosos sobre NLP: transferimos conocimiento de la tarea de modelado de lenguaje a las distintas tareas finales que realizamos como POS tagging, análisis de sentimientos, detección de entidades nombradas, etc.

*Tres* Los autores proponen 3 etapas: primero, el pre-entrenamiento sobre la tarea de modelado de lenguaje en un gran dataset de texto (e.g. Wikipedia o Common Crawl); segundo, un ajuste de la tarea de modelado de lenguaje sobre el texto de la tarea en cuestión (LM fine-tuning); y finalmente, el entrenamiento sobre las etiquetas de la tarea (Classifier fine-tuning). La figura 2.3 ilustra las 3 etapas para el problema de clasificación de sentimientos. Entre varias técnicas que utilizan para entrenar estos modelos y evitar el olvido catastrófico, vale destacar el uso de *slanted triangular learning rates*, donde el learning rate tiene una etapa de *warmup* donde sube hasta el pico y luego una etapa de *annealing* donde se reduce linealmente hasta 0 por el resto del entrenamiento. Esta técnica es también utilizada por *BERT* y otros modelos de lenguaje basados en transformers.

El modelo de lenguaje utilizado por los autores de ULMFiT utiliza una arquitectura AWD-LSTM [94]. Estas arquitecturas recurrentes fueron el estado del arte hasta el momento, pero fueron superados a los pocos meses por los modelos de lenguaje basados en *transformers*.

### 2.3.2. Modelos basados en atención

Antes de detallar los modelos basados en transformers, hacemos una pequeña digresión sobre traducción automática y los modelos de atención, conceptos necesarios para explicar adecuadamente esta arquitectura. Una de las limitaciones de los modelos basados en redes recurrentes es que sufren sesgo de **localidad** o **secuencialidad** (locality bias) [8]. En palabras coloquiales, las redes recurrentes tienen problemas para aprender dependencias

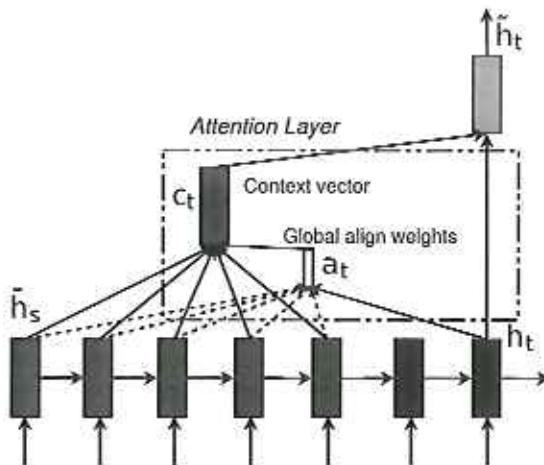


Fig. 2.4: Mecanismo general de atención. En azul, la salida del encoder recurrente de la entrada. En rojo, la salida del decoder recurrente. Fuente: Luong et al. [85]

de largo rango en las oraciones, siendo esto producto de su arquitectura autorregresiva donde se construye la salida  $y_t$  en base a  $y_{t-1}$ . Este sesgo es particularmente dañino en tareas de transducción de secuencias con la arquitectura encoder-decoder básica ya que a esto se le suma un cuello de botella forzoso por la compresión de toda la secuencia de entrada en un vector de longitud fija.<sup>5</sup>

Una de las formas de mitigar este sesgo es la utilización de mecanismos de *atención* [6]. Suponiendo una arquitectura recurrente de encoder y decoder, y siguiendo la notación de Luong et al. [85], para la tarea de traducción automática de una secuencia  $(x_1, \dots, x_n)$  a  $(y_1, \dots, y_m)$ , con estados ocultos  $(\bar{h}_1, \dots, \bar{h}_n)$  para la entrada y  $(h_1, \dots, h_m)$  y para la salida, el mecanismo de atención<sup>6</sup> consta de calcular para cada paso  $t$  un vector de contexto

$$c_t = \sum_{i=1}^n \alpha_i^{(t)} \bar{h}_i$$

donde  $\alpha^{(t)}$  es el vector de alineamiento, calculado como

$$\alpha^{(t)} = \text{softmax(score}(\bar{h}_1, h_t), \text{score}(\bar{h}_2, h_t), \dots, \text{score}(\bar{h}_n, h_t))$$

Cada  $\text{score}(\bar{h}_i, h_t)$  marca una similaridad no normalizada entre sus argumentos. Las alternativas planteadas en Luong et al. [85] son:

$$\text{score}(\bar{h}_i, h_t) = \begin{cases} \bar{h}_i^T h_t & \text{dot} \\ \bar{h}_i^T W h_t & \text{general} \\ v^T \tanh(W[\bar{h}_i^T; h_t]) & \text{concat} \end{cases} \quad (2.4)$$

<sup>5</sup> Sutskever et al. [139] en su trabajo observa que invertir la oración de entrada obtiene mejores resultados para la tarea de traducción automática. Esto también es un síntoma de este problema

<sup>6</sup> global, en Luong et al. [85] se menciona el mecanismo local que no consideramos

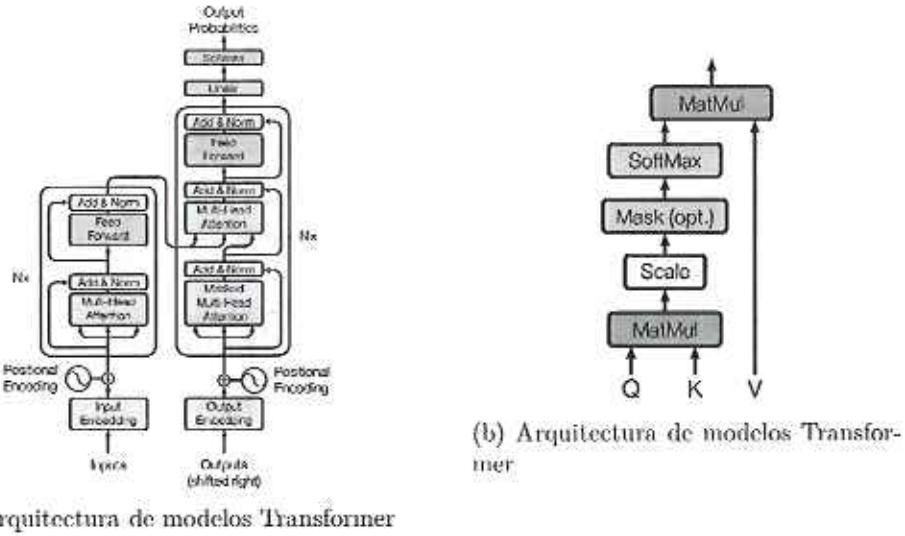


Fig. 2.5: Modelo de transformador y su versión de auto-atención. La subfigura 2.5a muestra la arquitectura de los codificadores y decodificadores. Fuente: Vaswani et al. [142]

con  $W$  y  $v$  parámetros adicionales. En el caso de la atención producto interno podemos reescribir todas las ecuaciones como:

$$C = \text{softmax}(H\tilde{H}^T)\tilde{H} \quad (2.5)$$

donde  $\tilde{H}, H$  son los vectores que tienen  $(\tilde{h}_1, \dots, \tilde{h}_n)$  y  $(h_1, \dots, h_m)$  como filas respectivamente, y softmax se calcula fila a fila.

Finalmente, el vector  $\tilde{h}_t$  es calculado como una transformación del estado oculto del decoder  $h_t$  y el vector contextual  $c_t$ :

$$\tilde{h}_t = \tanh(W_h[h_t; c_t])$$

El vector  $\tilde{h}_t$  codifica información de manera global de todos los estados ocultos del codificador, atenuando los problemas de localidad de las redes recurrentes. Esta técnica se convirtió en parte integral de los modelos seq2seq como ser traducción automática, sumarización, entre otras. La figura 2.4 ilustra esta arquitectura.

La técnica de auto-atención o intra-atención [107] consiste en aproximadamente la misma idea que la atención sólo que teniendo una única secuencia; podemos asumir ecuaciones similares con  $\tilde{h}_i = h_i$ . La auto-atención genera representaciones de los distintos vectores de entrada observando la totalidad de la secuencia, a diferencia de las redes recurrentes que sólo construyen una representación en base al paso anterior. Esta capa se utiliza en arquitecturas para clasificación de texto encima de una capa recurrente para generar representaciones con dependencias sin distinción de la distancia entre los distintos tokens.

## 2.4. Transformers

Mencionamos el sesgo de la secuencialidad como uno de los problemas de los modelos recurrentes. Otro de los grandes obstáculos para las arquitecturas autorregresivas es la parallelización. El modelo de cómputo secuencial donde  $h_t$  se calcula en base a  $h_{t-1}$  inhibe un

cálculo paralelo, donde las diferentes representaciones puedan ser generadas simultáneamente. Parikh et al. [107] es uno de los primeros trabajos que proponen una arquitectura para el problema de inferencia (NLI) enteramente basada en modelos de atención, sin modelos recurrentes.

Vaswani et al. [142] introdujeron la arquitectura **Transformer** para la tarea de traducción automática. Esta arquitectura no utiliza capas recurrentes ni convolucionales, basándose enteramente en el mecanismo de auto-atención. La figura 2.5a muestra la arquitectura de los modelos basados en Transformer, organizado en forma de encoder-decoder, con 6 capas de cada uno.

Cada capa del encoder utiliza un mecanismo de auto-atención múltiple seguido de una capa feed-forward punto a punto. Las dos capas de auto-atención o feed-forward están sucedidas por conexiones residuales [62] para facilitar el flujo del gradiente en una arquitectura profunda y una capa de normalización, de manera que la salida se expresa como:

$$\text{Layer}(x) = \text{Norm}(x + \text{subLayer}(x))$$

Las capas decodificadoras son similares, salvo que se les agrega una capa extra de auto-atención donde se combinan las salidas del encoder con las representaciones que genera el decoder. A su vez, las capas de multi-atención están enmascaradas para no poder “ver” las representaciones que se generan en pasos posteriores para guardar su naturaleza secuencial en la tarea.

El cálculo de atención utilizado en este trabajo es similar al visto en la ecuación 2.5, aunque normalizado por  $\sqrt{d_k}$ , donde  $d_k$  es la dimensión de los vectores de entrada:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q^T K}{\sqrt{d_k}}\right)V$$

Cada capa utiliza varias cabezas de auto-atención, cuyas salidas son concatenadas y proyectadas. A su vez, la salida de cada una de las capa pasa por una regularización de tipo dropout [137].

Un punto no menor es que el modelo Transformer, siendo que no tiene ningún tipo de recurrencia y convolución, carece de cualquier ordenamiento de la secuencia de tokens. Para inyectar ese conocimiento en la red, utilizan *vectores de posicionamiento* (positional embeddings) que se suman a los vectores de entrada de la capa de embeddings, como se ilustra en la figura 2.5a. Estos vectores no son parámetros entrenados (como sí lo son en *BERT*) sino que se calculan mediante funciones sinusoidales.

No nos extenderemos más en la explicación de esta arquitectura, y referimos para más información a los excelentes artículos *Transformers from Scratch*<sup>7</sup>, *Annotated Transformer*<sup>8</sup> y *The Illustrated Transformer*<sup>9</sup>.

## 2.5. GPT, BERT y amigos

Combinando las ideas de ULMFit –entrenaje semi-supervisado sobre la tarea de modelado de lenguaje– y la arquitectura Transformer –removiendo las redes recurrentes y

<sup>7</sup> <http://peterbloem.nl/blog/transformers>

<sup>8</sup> <https://nlp.seas.harvard.edu/2018/04/03/attention.html>

<sup>9</sup> <https://jalammar.github.io/illustrated-transformer/>

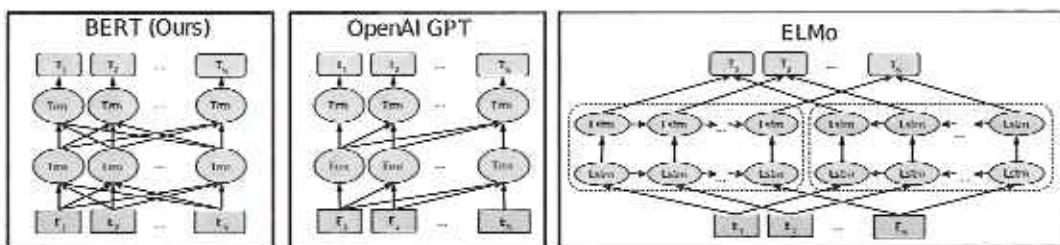


Fig. 2.6: Comparación entre ELMo, GPT y BERT. ELMo genera vectores contextualizados mediante dos modelos de lenguaje recurrentes (uno de izquierda a derecha y el otro al revés), . GPT pre-entrena un modelo de lenguaje basado en Transformers. BERT genera representaciones bidireccionales

facilitando la paralelización del cálculo—en Radford et al. [122] se introduce GPT (*generative pre-training*). Esta técnica consiste de un pre-entrenamiento sobre un gran corpus no etiquetado seguido de un fine-tuning discriminativo para cada tarea, muy en la línea de Howard and Ruder [67], introduciendo unos pocos parámetros específicos para cada una de estas. El modelo que usa esta tarea es el de **modelado de lenguaje causal** —es decir, de izquierda a derecha. Este modelo obtuvo el estado de arte para el benchmark GLUE [143], superando otros modelos como ELMo.

**BERT** [35] (Bidirectional Encoder Representations from Transformers) plantea una modificación sobre GPT: en lugar de pre-entrenar el modelo sobre la tarea de modelado de lenguaje causal —de izquierda a derecha— hacerlo sobre la tarea de modelado de lenguaje enmascarado. Esta tarea (usualmente llamada *Cloze task* [140]) consta de enmascarar una cierta cantidad de palabras de una frase, y luego intentar predecir las palabras faltantes. Por ejemplo, en la siguiente frase, consta de reemplazar los dos tokens [MASK]:

El [MASK] es celeste y el pasto [MASK]

A diferencia de la tarea de modelado de lenguaje causal, los autores argumentan que esta tarea permite generar representaciones bidireccionales ricas. La figura 2.6 muestra una comparación entre los distintos tipos de pre-entrenamiento de GPT.

BERT es pre-entrenado conjuntamente sobre dos tareas: una, la ya mencionada tarea de modelado de lenguaje enmascarado; la otra, la tarea de *predicción de próxima oración* (Next Sentence Prediction, NSP). Esta tarea consiste en predecir si, dado un par de oraciones, la segunda es la que sigue a la primera. El 50% de las ocasiones, las dos oraciones de entrada son contiguas en el texto de origen, y el 50% restante son dos oraciones aleatorias concatenadas. Esta tarea debiera guardar cierta relación con la semántica de las oraciones y su interrelación, necesaria en tareas como NLI y Question Answering.

Dos caracteres especiales son utilizados en **BERT**: [CLS] y [SEP]. Estos caracteres se utilizan para delimitar las oraciones en la entrada de la red, y también para separar las dos oraciones de entrada. Durante el pre-entrenamiento, la representación generada por [CLS] es utilizada como la predicción de la tarea NSP y la representación de cada token enmascarado es utilizado como entrada de una capa softmax para predecir el token faltante. Para la mayoría de las tareas de clasificación de texto, [CLS] es usado como la representación de la (o las) oraciones de entrada. La figura 2.7 ilustra esta metodología para la tarea de QA, ligeramente más compleja que la clasificación de texto.

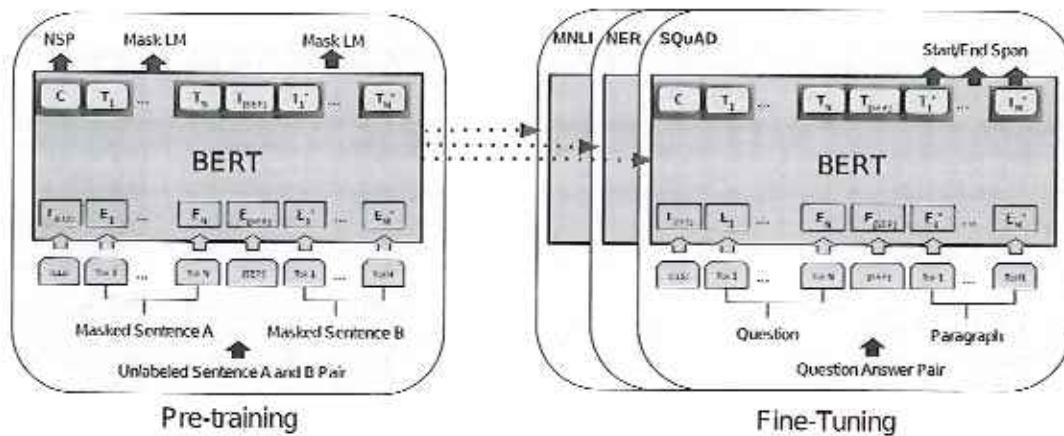


Fig. 2.7: Pre-entrenamiento y fine-tuning de BERT para distintas tareas. Fuente: Devlin et al. [35]

*BERT* utiliza WordPiece [149], un algoritmo de tokenización muy similar a BPE [133], con un vocabulario de largo 30,000 para separar su entrada en subpalabras de manera eficiente. A su vez, para cada posición de su entrada entrena embeddings posicionales (a diferencia de los embeddings posicionales fijos originales de Vaswani et al. [142]) con un límite de 512, y dos embeddings especiales para la primera oración y la segunda oración de la entrada. Los vectores que ingresan a la primera capa de transformers son la suma de los embeddings de cada token, los embeddings posicionales, y los embeddings de oración.

El proceso de pre-entrenamiento es realizado sobre la concatenación de dos corpora: BooksCorpus [159] y la versión en inglés de Wikipedia. Estas dos fuentes son utilizadas ya que permiten extraer pares de palabras contiguas, algo necesario para la tarea de NSP.

Liu et al. [84] propone dos modificaciones al pre-entrenamiento de *BERT*: en primer lugar, remover la tarea de NSP; y en segundo lugar, realizar un pre-entrenamiento más extenso y con batch sizes más grandes, pasando de lotes de 512 a 8,192 oraciones. Este modelo pre-entrenado (al cual se denomina *RoBERTa*) obtuvo mejor desempeño que *BERT* en el dataset de GLUE y otras tareas.

Luego de estos modelos de lenguaje, una suerte de guerra armamentística tuvo lugar para entrenar modelos más grandes y con más parámetros al observar que aumentando la cantidad de estos mejoraba la performance en distintas tareas –sin observarse aún un techo más que los recursos computacionales y energéticos disponibles en el planeta. Sólo para ilustrar el punto, la versión *base* de *BERT* tiene 110M parámetros, su versión larga 330M, GPT-2 1,500M, Turing NLG de Microsoft 17,000M y finalmente GPT-3 [23] tiene la asombrosa cantidad de 175,000M parámetros.



### 3. EXTRACCIÓN DE OPINIONES DE TEXTOS SOCIALES

La extracción de opiniones en distintos espacios virtuales ha atraído mucho interés desde los comienzos de la Web 2.0. Inicialmente motivados por fines puramente comerciales, diferentes intereses y oportunidades han surgido debido al desarrollo de la técnica y la proliferación de las redes sociales: desde fines sociológicos (como el análisis de discurso de odio o las reacciones a la pandemia) hasta políticos (como observar cuál es la opinión general sobre tal o cual candidato o sobre un tema candente). Desde comienzos de los años 2000, y debido a la combinación del desarrollo de métodos de aprendizaje estadístico y la cantidad creciente de datos disponibles generados por usuarios en Internet, numerosos trabajos han analizado este tipo de textos para poder extraer conocimiento **subjetivo** de quienes vuelcan sus pensamientos en las redes sociales y otros espacios virtuales.

Debido a la inmensa cantidad de contenido generado en diversos sitios y redes sociales (se estima que en el mundo se generan 500M tweets por día para 2021<sup>1</sup>), esta tarea es difícil de realizar sin algún tipo de automatización. Para ello, muchísimo esfuerzo se ha volcado en utilizar técnicas de aprendizaje automático para atacarla. El avance de las técnicas de NLP –como hemos descrito en el capítulo anterior– han permitido avanzar sobre este terreno; sin embargo, muchas de las limitaciones actuales del área ~~en conjunto con~~ las dificultades particulares de las interacciones en medios sociales hacen esta tarea difícil.

En este capítulo haremos una breve introducción al análisis de sentimientos o extracción de opiniones sobre textos de redes sociales. Esto es, dado un texto generado por un usuario (un post en Facebook, Instagram, un tweet, etc) predecir alguna característica discreta de éste, como por ejemplo si es un texto positivo o negativo, si tiene algún tipo de emoción de ira, alegría, u otra; si contiene discurso de odio contra algún grupo o no; si es irónico; entre otras. En base a datasets en español para distintas tareas, presentaremos modelos de clasificación basados en técnicas del estado del arte.

Analizaremos también algunas cuestiones relacionadas a la adaptación de dominio y representaciones generadas sobre dominios de textos generados por usuarios, plantando algunas líneas que retomaremos en capítulos posteriores.

#### 3.1. Motivación

Las motivaciones para extraer opiniones subjetivas de usuarios en Internet son múltiples, aunque intentaremos categorizarlas en algunos grupos de notable interés. Dado el aumento considerable de contenido generado por usuarios desde la popularización de la WWW luego del año 2000<sup>2</sup> y subsiguientemente con la explosión de las Redes Sociales –una de las cuestiones que motoriza este área es netamente comercial: ¿qué opinan los usuarios sobre este nuevo producto? ¿cuáles creen que son sus falencias? ¿qué tal es el servicio en tal o cual Restaurant? Desde ya más de 20 años, numerosos sitios de e-commerce brindan la posibilidad de que los clientes vuelquen sus opiniones al respecto de los productos que consumen en sus plataformas, como así también pueden incorporarse en otras aplicaciones que brindan esta posibilidad de expresar comentarios sobre productos, servicios u otros lugares. Para citar unos ejemplos, IMDb permite agregar comentarios sobre películas, Google Maps sobre distintos sitios –tanto turísticos como locales comerciales–, o los

<sup>1</sup> Fuente: <https://www.internetlivestats.com/twitter-statistics/>

distintos sitios de venta minorista como MercadoLibre, eBay, o Amazon. Sobre esta información disponible, una encuesta de 2008 [65] reportó que cerca del 81 % de los usuarios de Internet de entonces (60 % de los habitantes de Estados Unidos) realizaron investigación online antes de sus compras, aunque también reportaban problemas a la hora de encontrar información valiosa para sus fines.

Con la explosión de las redes sociales se abrieron nuevas oportunidades y posibilidades de preguntas a contestar mediante la extracción de opiniones en este medio<sup>2</sup>. Uno de estos horizontes, que es de interés particular para esta tesis, es el de las preguntas de carácter sociológico y político. Preguntas que pueden suscitar interés dentro de este punto pueden ser:

- ¿cuál es la opinión de los usuarios acerca de la legalización del aborto en cierto país que tiene en tratamiento este tema? ¿es representativo de la población en general? [54]
- ¿cómo se ha modificado el humor social de acuerdo a crisis económicas o pandemias como la del COVID-19? [90]
- ¿qué características tiene el discurso de odio contra los inmigrantes en España en el medio del auge de la ultraderecha en dicho país? [25]
- ¿qué artículos periodísticos suscitan la mayor cantidad de discurso discriminatorio en las redes sociales?
- ¿cuáles son las principales vulnerabilidades e intereses de ciertos sectores de la población? [160]<sup>3</sup>

entre otras. Estos tópicos son de gran interés para investigadores y políticos. Usualmente, la forma más estandarizada de acceder a la opinión de distintos actores sociales ha sido la de encuestas; sin embargo, la recolección y extracción automática de opiniones de medios virtuales brinda una alternativa (a veces) más económica y masiva aunque con un sesgo poblacional distinto al de otras metodologías.

### 3.2. Clasificación de textos sociales

Muchos de estos problemas de extracción de opiniones se pueden plantear como problemas de clasificación [106]: dado un contenido social, queremos predecir una clase  $c \in \mathcal{C}$ , con  $\mathcal{C}$  un conjunto finito de clases. El análisis de polaridad, por ejemplo, se puede plantear como un problema de clasificación en el que dado un texto en una red social, predecir si tiene un sentimiento positivo, negativo, o neutro. El problema de análisis de emociones se puede plantear como, dado un texto, seleccionar la emoción predominante en él de un conjunto de 6 emociones y una neutral.

Algunas variantes de estos problemas se pueden dar en el contenido analizado. Por ejemplo, el *Análisis de Sentimiento basado en aspectos* (usualmente denominada *ABSA*) en

Esta es una  
definición parcial.  
Hay variantes más...

<sup>2</sup> Si bien algunas preguntas de carácter sociológico tuvieron lugar con anterioridad, podemos marcar el uso intensivo de Facebook y Twitter como el comienzo de un estudio más sistemático de ellas dado el enorme volumen de datos accesibles para los investigadores

<sup>3</sup> Esto, según parece, fue utilizado en el affaire de Cambridge Analytica en las elecciones de 2016 que consagraron a Donald Trump en EEUU

la literatura por sus siglas en inglés) es una variante de la clasificación de polaridad en la que queremos predecir el sentimiento de un texto para cierto aspecto [108]; por ejemplo, en la oración “lindo lugar, la comida está muy bien pero la cerveza es horrible” (en una posible reseña de un restaurante) podemos identificar dos sentimientos distintos: uno positivo para la comida y otro negativo para la cerveza. Dentro de estos problemas que complejizan la entrada, podemos contar algunos de carácter multimodal. En Sharma et al. [134] se plantea un problema de análisis de emociones para memes donde la entrada (el contenido social) consta de imágenes y texto, y se intenta predecir la emoción predominante.

Análogamente, se puede agregar cierta complejidad en la salida. Por ejemplo, el Stanford Sentiment Treebank (SST) [136] plantea una tarea de análisis de polaridad asignando una escala de Likert [83]: cada comentario está etiquetado como muy negativo, algo negativo, neutral, algo positivo o muy positivo. Así mismo, otra posibilidad es la de predecir conjuntamente varias variables; por ejemplo, predecir si un comentario es discriminatorio, si es dirigido a un grupo o una persona, y si es agresivo, como el dataset de hatEval [7]; o bien, dado un comentario de una nota periodística, predecir las características que discrimina si es que hay alguna (como ser a las mujeres, al colectivo LGTBI, por motivos raciales, etc). De este último ejemplo hablaremos en los capítulos 5 y 6.

### 3.3. Trabajo previo

El análisis de sentimientos, opinion mining u opinion extraction suscitó interés casi desde el comienzo de la generación masiva de contenido de parte de los usuarios en la WWW. Particularmente desde la eclosión de las redes sociales, la inmensa cantidad de contenido generado por usuarios ha sido una fuente de información sin antecedentes para la extracción de todo tipo de opiniones. La bibliografía que comprende este tema es demasiado extensa y escapa los objetivos de esta tesis centrada en la detección de discurso de odio. Mencionamos algunos surveys relevantes y una pequeña selección de trabajos a continuación.

Pang and Lee [106] ofrecen un amplio repaso sobre los usos, aplicaciones, técnicas y dificultades de este problema en la era pre-deep learning and pre-redes sociales, mencionando cuestiones como los problemas que los diferentes dominios presentan a las técnicas del entonces (y también de ahora). Pak and Paroubek [105] es uno de los trabajos pioneros en plantear Twitter como una fuente de mensajes para la extracción de opiniones – en particular, para analizar polaridad de mensajes – proponiendo una metodología para recolectar y etiquetar un dataset sobre esta red social. Yue et al. [153] presenta un recuento de los diversos problemas de análisis de sentimientos aplicados a textos de redes sociales y las técnicas para atacarlo.

Dentro de los recursos para tareas de opinion mining, Maas et al. [87] es un dataset de análisis de polaridad con dos etiquetas (positivo y negativo) sobre películas en la plataforma de IMDb, extensamente utilizado para la tarea de análisis de sentimientos. Socher et al. [136] presenta el Stanford Sentiment Treebank (SST), un dataset que contiene información granular en una escala similar Likert de polaridad sobre las distintas subpartes de cada oración. Esta tarea forma parte del benchmark de General Language Understanding Evaluation (GLUE) [143]. El workshop SemEval<sup>4</sup> ha generado numerosos recursos para tareas de opinion mining en redes sociales, como Análisis de Polaridad, Análisis de Polaridad basado en aspecto, análisis de emociones, entre otras.

<sup>4</sup> <https://semeval.github.io/>

(en Masa)

revisar  
consistencia en  
uso de  
mayúsculas

Tarea	Datasct	Tamaño	Clases
Análisis de Sentimientos	TASS 2020	14,509	Neg (39.80 %)
			Neu (29.52 %)
			Pos (30.68 %)
Análisis de emociones	TASS 2020/EmoEvent	8,409	Otra (49.08 %)
			Alegria (21.58 %)
			Tristeza (12.00 %)
			Ira (10.19 %)
			Sorpresa (4.10 %)
			Disgusto (1.91 %)
			Miedo (1.14 %)
Detección de ironía	IroSVA 2019	9,000	No irónico (66.68 %)
			Irónico (33.32 %)

Tab. 3.1: Tareas evaluadas en este capítulo, junto a datos estadísticos de los datasets utilizados

Centrándonos en los recursos y tareas en español, uno de los principales polos de esto es el Taller de Análisis de Sentimientos (TASS) [31, 47, 89] organizado por la Sociedad Española de Procesamiento Natural (SEPLN) y a partir de 2020 en el marco del evento Iberian Languages Evaluation Forum (IberLEF). En este foro se presentaron tareas y datasets de análisis de sentimiento [47], de emociones[118], de toxicidad [151], entre otras.

### 3.4. Tareas analizadas

Analizamos 3 tareas de extracción de opiniones sobre redes sociales y las usamos como benchmark para las diferentes técnicas de clasificación. La tabla 3.1 contiene información sobre las distintas tareas analizadas y los datasets utilizados para ellas. Una de las tareas analizadas es la de **análisis de polaridad**: dado un tweet, detectar si tiene una polaridad general positiva, negativa, o neutra. Utilizamos el dataset de TASS 2020 [47], anotado con estas 3 clases y con información de las diferentes variedades dialectales del español a la que pertenece cada tweet. Para nuestro análisis, ignoramos estas distinciones y fusionamos todos los datos en un solo conjunto de datos (con las 3 particiones correspondientes de entrenamiento, validación, y test).

Para el análisis de emociones, también usamos el conjunto de datos de TASS 2020 *EmoEvent* [118]. Este dataset multilingüal (español e inglés) contiene tweets etiquetados con las seis emociones básicas de Ekman [37]: *ira*, *disgusto*, *miedo*, *alegría*, *tristeza*, *sorpresa* y también una emoción *neutral*. El dataset fue recolectado en base a ocho eventos globales diferentes de diferentes dominios (políticos, entretenimiento, catástrofes o incidentes, commemoraciones globales, etc.) por lo que las emociones siempre están relacionadas con un fenómeno en particular. Solo conservamos la parte en español, que contiene 8.409 tweets.

La **detección de ironía** también es una tarea que ha ganado popularidad recientemente. Algunos trabajos han mostrado que tiene importantes implicaciones en otras tareas de NLP de carácter semántico, como por ejemplo Gupta and Yang [57] muestran que el uso de funciones derivadas de la detección de sarcasmo mejora

el rendimiento en la tarea de análisis de sentimientos. Además de esto, el contenido generado por los usuarios es una rica y vasta fuente de ironía, por lo que esta tarea es de particular importancia para el dominio de las redes sociales. IroSVA [103] es un dataset en Español (publicado en el contexto de TASS 2019) que tiene la particularidad de considerar los mensajes no como textos aislados sino con un contexto dado (un titular o un tema). Consta de 7.200 instancias y 1.800 ejemplos de prueba divididos en tres variantes geográficas de Cuba, España y México, cada una con una etiqueta binaria que indica si el comentario contiene ironía o no. A diferencia de las tres tareas anteriores mencionadas aquí, este conjunto de datos contiene no solo mensajes de Twitter, sino también de comentarios de noticias y foros de debate como 4forums.com y Reddit.

La tabla 3.2 ilustra algunos ejemplos seleccionados para las distintas tareas y sus clases. En el caso de la tarea de detección de emociones, podemos ver que algunos ejemplos han sido preprocesados por sus autores para ocultar los hashtags y urls.

### 3.5. Normalización y preprocesamiento

Una de los pasos más importantes para la manipulación de texto social es el preprocesamiento. Con esto nos referimos al conjunto de técnicas dedicadas a disminuir la variabilidad del texto y aproximarla a una forma lo más normal posible<sup>5</sup>. El texto generado por usuarios en medios informales suele ser más irregular que el texto proveniente de otras fuentes, con errores ortográficos y usos coloquiales que hacen difícil el tratamiento por algoritmos de NLP. Para poner un ejemplo, la frase “¡qué lindo día, loco!” puede ser representada de las siguientes maneras:

- q lindo dia loco
- k lindo diaaaaaaaa loco
- ke lendo diaa lk

entre otras formas posibles. Uno de los primeros trabajos que aborda este problema para el dominio de redes sociales es el de Han and Baldwin [60]. En base a un dataset de tweets, observaron que las palabras fuera de vocabulario (OOV en inglés)<sup>6</sup> en dicha red social tienen una alta frecuencia de incidencia. Ejemplos de estas palabras son neologismos, errores ortográficos, typos, contracciones típicas de esta red social (lk), sustituciones fonéticas (wacho en vez de guacho), entre otras. Este problema de la desnormalización del texto generado por usuarios planteaba un serio inconveniente para los métodos del estado del arte de ese entonces basados en bolsas de palabras o representaciones sobre palabras aisladas<sup>7</sup>. Para mitigar el problema de la alta dimensionalidad que generan estas palabras fuera de vocabulario, los autores proponen diversas estrategias para normalizar las palabras y testean sus métodos sobre datasets de Twitter y SMS.

<sup>5</sup> Eisenstein [36] discute siquiera que exista tal forma normal

<sup>6</sup> fuera del vocabulario de un diccionario estándar de GNU en inglés

<sup>7</sup> Recordemos que para el momento de la publicación de este trabajo aún no se usaban redes neuronales, word embeddings, ni mucho menos métodos más avanzados como fasttext, que ayuda mucho en las palabras OOV

Tarea	Clase	
Emociones	Neutral	Espectantes para ver el tercer capitulo de HASHTAG. El principio empieza bien.
	Alegria	Lo de Messi ha sido increíble! HASHTAG
	Tristeza	Un día lamentable. Se perdieron años de historia, de cultura, de arquitectura... Me siento devastada al ver las imágenes del incendio de la catedral de HASHTAG en HASHTAG URL
	Ira	URL que discurso para ponerlo a llorar, Putos humanos, para cuando la extinción??
	Sorpresa	Santa Maria Madre de Dios IIASITAG URL
	Miedo	Joder, izquierda venció ¿y qué? A mi me preocupa y mucho que Vox haya pasado de 0 a 24!! ¡a nadie le parece un montón? No sé, debo ser idiota. IIASITAG
	Disgusto	Como se nota que HASHTAG ya no tiene el apoyo de los libros. Vaya mierda de temporada se están sacando.
Sentimientos	Negativo	que triste es la realidad
	Positivo	Hola a todos corazones, buen dia Y FELIZ NAVIDAD A TODOS USTEDES! DIOS ME LOS BENDIGA Y ME LOS LLENE DE TODO SU AMOR
	Neutral	@AlfonsoEmilioL La próxima vez que no vea lo entrevistado y preguntó que escucha.
Ironía	Irónico	Pues, noticia de última hora, eres un anormal que no sabe escribir. Es LESIONARAN. El juez paraliza la exhumación de Franco alegando peligro para los operarios. Pues me parece muy bien, porque con las ratas hay que ser muy precavido. @CristinaSegui_ @Pablo_Iglesias_ Doctorado en cambiar pañales....
	No irónico	No me sirvió para nada todo fue en vano. Espero mejore la calidad.

Tab. 3.2: Tareas evaluadas en este capítulo, junto a datos estadísticos de los datasets utilizados

Actualizar

Eisenstein [36] trata este problema desde una perspectiva más amplia con los enfoques utilizados hasta el momento. El autor plantea dos posibilidades para las tareas de NLP en el medio social: la **normalización** sería una forma de adaptar el texto a las herramientas, mientras que la **adaptación de dominio** sería adaptar las herramientas al texto. Con lo primero nos referimos al conjunto de técnicas que podemos utilizar para acercar la distribución del texto lo más posible a un dominio formal (por ejemplo, sobre el que entrenamos un POS tagger), mientras que por adaptación de dominio nos referimos a la construcción de datasets y algoritmos particulares para distintas tareas de NLP en redes sociales, como por ejemplo POS tagging [50], NER [125], entre otros.

Con el advenimiento de las redes neuronales y los word-embeddings como GloVe o word2Vec, esto siguió siendo un problema ya que cada representación se calcula sobre cada una de estas palabras. En el caso de una palabra fuera de vocabulario, un mecanismo habitual es asignarles un token especial "`<unk>`", y por lo tanto, una única representación para todas esas palabras OOV. Sin embargo, esto puede ser problemático ya que elimina muchas palabras similares a otras que sí tenemos en el vocabulario. Bojanowski et al. [19] propuso una solución a esto al permitir formar la representación de cada palabra mediante una combinación lineal de las representaciones de las "subpalabras" de cada una (ver sección 2.2).

Con el advenimiento de los modelos basados en transformers, otros tipos de tokenización fueron propuestos que permiten reducir las palabras OOV. BPE [133], Word Piece, Sentence Piece [78] utilizan, en lugar de separar el texto mediante los espacios, en la utilización de distintos tipos de subpalabras. Esta técnica permite reducir la incidencia de palabras OOV notablemente.

En el caso de modelos basados en transformers entrenados sobre texto social,<sup>4</sup> Nguyen et al. [102] plantearon experimentos en tareas sociales usando dos formas de normalización: una débil, donde sólo convierten nombres de usuario en un token especial `@USER` y a las urls en otro token especial `HTTPURL`, y otra estrategia fuerte donde usan diccionarios de normalización. Para un conjunto de tareas de clasificación sobre Twitter y distintos modelos pre-entrenados, los resultados de los experimentos arrojaron que la normalización fuerte empeora ligeramente la performance.

En base a esto, adoptamos una estrategia similar a la normalización débil mencionada en Nguyen et al. [102]:

- Convertimos los handles a un token especial `@usuario`
- Convertimos URLs a un token especial URL
- Convertimos los emojis a representaciones textuales usando la librería emoji<sup>8</sup>
- Normalizamos risas ("jajajajjjajaja" lo convertimos a "jaja")

● Procesamos hashtags: #EsteHayQueNormalizar lo convertimos a *hashtag esto hay que normalizar*

- Limitamos repeticiones de caracteres a 3 ocurrencias → Ejemplo?

<sup>8</sup> <https://pypi.org/project/emoji/>

¿  
este ~~que~~ se hace en base a las mayúsculas? Activar.

Si bien en algunas secciones (como en 4.5.1) usaremos variaciones de estos métodos –particularmente para modelos lineales–, en general seguiremos esta estrategia de normalización.

### 3.6. Modelos de clasificación

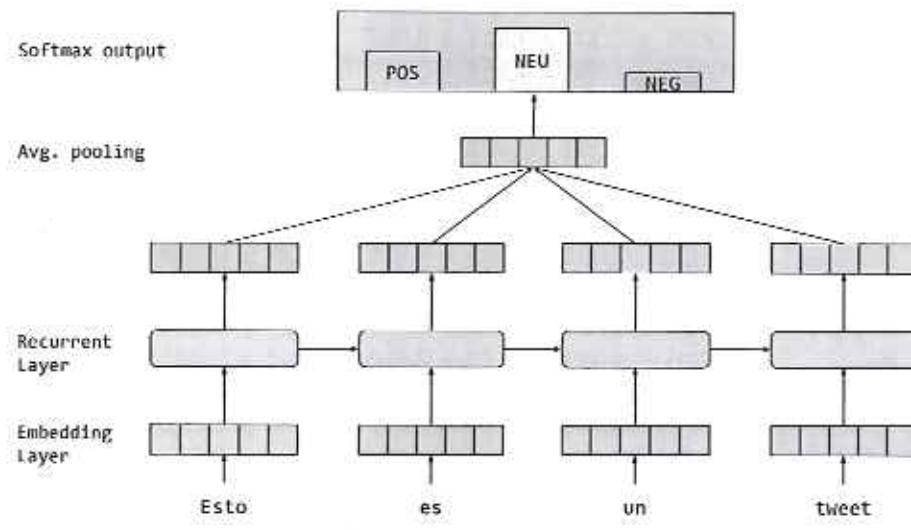
Describimos a continuación los clasificadores utilizados para las tareas. Podemos, a grandes rasgos, describir a todos nuestros clasificadores como compuestos de dos partes: un *codificador* (o *encoder* en inglés) que genera una representación *continua* de longitud fija del texto de entrada, y un *decodificador* que toma esa codificación y la convierte a la salida deseada.

Todos nuestros problemas analizados son de clasificación multiclase: elegir exactamente una clase entre varias. Para ello, nuestro decodificador será de la forma  $\text{softmax}(Wx + b)$ , donde  $W$  es una matriz de pesos y  $b$  un vector de bias, ambos parámetros de nuestra red neuronal. Los modelos planteados diferirán entonces en los codificadores. Proponemos las siguientes variantes:

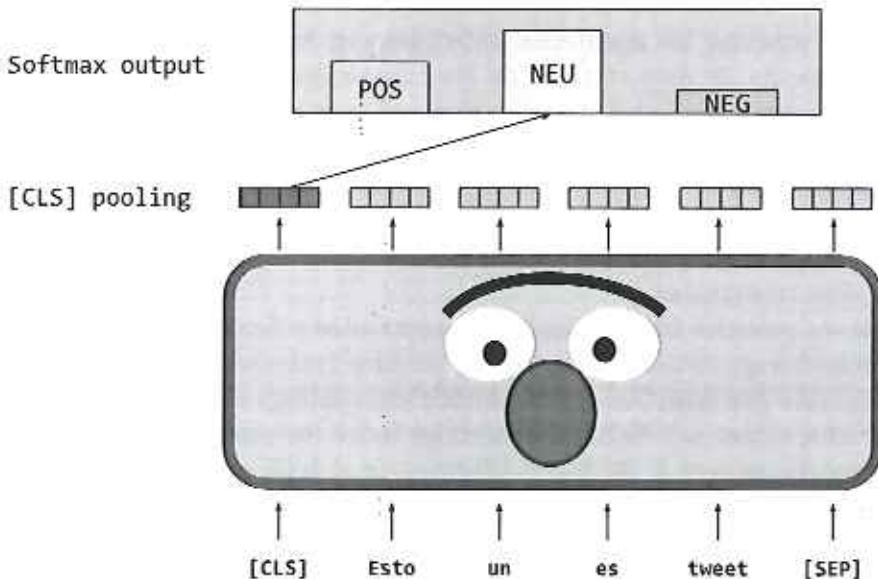
- **FFN**: Un perceptrón multicapa (feed-forward network) con una función de activación intermedia *ReLU*
- **GRU/biGRU**: Una red neuronal recurrente. La capa oculta es una Gated Recurrent Unit (GRU) unidireccional o bidireccional.
- **BETO|mBERT|RoBERTa|BERTin**: un modelo pre-entrenado de lenguaje basado en transformers.

En el caso de la FFN, la codificación se da luego de aplicar la función ReLU a la capa intermedia de la red. En el caso de la GRU/biGRU, la codificación se da como el promedio de los vectores salida de cada paso (podemos pensar cada uno de estos como representaciones contextualizadas de cada palabra). En el caso de los modelos basados en transformers, la codificación se da tomando la salida del carácter de inicio ([CLS] en el caso de BERT pero puede variar para RoBERTa u otros modelos)<sup>9</sup>. La figura 3.1 ilustra la arquitectura de los clasificadores recurrentes y basados en transformers.

Usamos un tamaño oculto de 512 para los modelos recurrentes y FFN, y para todos los casos un dropout [137] de 0.1 sobre la codificación del tweet. Para los dos primeros modelos, los entrenamos usando Adam [74] con un learning rate de 0.001 y un decay de 0.01. Para los modelos basados en transformers, usamos Adam con un learning rate triangular de  $10^{-5}$  y un warmup del 10 % de los pasos. Entrenamos todos los modelos por 5 epochs y nos quedamos con los modelos que mejor performance tengan sobre el split de validación en términos de la métrica correspondiente a la tarea.



(a) Clasificador basado en redes recurrentes



(b) Modelos de clasificación basado en BERT y similares

Fig. 3.1: Clasificadores propuestos para las tareas de Análisis de Polaridad, Análisis de Emociones y Detección de Ironía. La subfigura 3.1a muestra la arquitectura del modelo recurrente, que usa una capa de embeddings basados en *fasttext* y codifica el tweet como el promedio de las salidas de la capa recurrente. La subfigura 3.1b muestra un clasificador basado en BERT, donde tomamos la salida del token [CLS] como la codificación del tweet. Ambos usan un decodificador softmax.

Modelo	Polaridad	Emociones	Ironía	Puntaje
RoBERTa	0,670 ± 0,006	0,527 ± 0,015	0,721 ± 0,008	0,665
BERTin	0,666 ± 0,005	0,524 ± 0,007	0,713 ± 0,012	0,660
BETO <sub>U</sub>	0,651 ± 0,006	0,532 ± 0,012	0,701 ± 0,007	0,653
BETO <sub>C</sub>	0,662 ± 0,005	0,516 ± 0,012	0,705 ± 0,009	0,652
mBERT	0,617 ± 0,003	0,493 ± 0,010	0,681 ± 0,010	0,627
biGRU <sub>TW</sub>	0,585 ± 0,011	0,264 ± 0,007	0,631 ± 0,011	0,518
biGRU <sub>CC</sub>	0,553 ± 0,008	0,231 ± 0,006	0,625 ± 0,009	0,486
GRU <sub>TW</sub>	0,602 ± 0,004	0,269 ± 0,003	0,628 ± 0,014	0,509
GRU <sub>CC</sub>	0,564 ± 0,004	0,237 ± 0,005	0,581 ± 0,016	0,474
ffn <sub>TW</sub>	0,516 ± 0,004	0,203 ± 0,003	0,627 ± 0,004	0,433
ffn <sub>CC</sub>	0,509 ± 0,003	0,179 ± 0,001	0,481 ± 0,003	0,393

Tab. 3.3: Resultados de la evaluación de los distintos modelos para las 3 tareas analizadas (Análisis de Emociones, Análisis de Emociones, Detección de Ironía). Los 3 resultados están dados en Macro F1, y expresados como la media de diez corridas junto a su desviación estándar. El puntaje de cada modelo es el promedio de las métricas para las 3 tareas

### 3.7. Resultados

No entiendo  
test estadístico para esto?

La tabla 3.3 muestra los resultados obtenidos con los distintos modelos, expresados como la media de diez corridas de los experimentos de clasificación junto a sus desviaciones estándar; ~~estos~~ lo realizamos con motivo de que el entrenamiento es estocástico a diferencia de otros modelos clásicos de Machine Learning. Podemos observar que los mejores resultados se obtienen con el modelo ~~roberta-base~~ para las 3 tareas, aunque las diferencias son pequeñas. ~~Todos los modelos basados en modelos pre-entrenados de transformers obtienen mejor performance que los basados en redes neuronales recurrentes.~~

función

Dentro de los modelos basados en redes recurrentes y feed-forward, aquellos que consumen embeddings entrenados en textos sociales (marcadas como *tw*) obtienen mejor performance que aquellos que consumen embeddings entrenados en Common Crawl (marcados como *cc*). Esta diferencia, en todos los casos, es estadísticamente significativa corriendo un test U de Mann-Whitney ( $p \leq 0,05$  para el caso de ironía y *biGRU*, para todas las demás comparaciones  $p \leq 0,001$  ).

### 3.8. Librería de análisis de sentimientos

Algo que suele obstaculizar la utilización de herramientas de extracción de opinión (como las que acabamos de ver en este capítulo pero así mismo las que veremos más adelante) con fines de investigación es la dificultad a su acceso. O bien estos servicios están detrás de APIs pagas con precios demasiado altos para los presupuestos académicos o están disponibles pero no en español u otros idiomas de bajos

<sup>9</sup> Si bien podría también tomarse el promedio como en el caso de las redes recurrentes, los modelos basados en transformers no sufren el cuello de botella que se genera tomando la última representación de la red recurrente.

i te referís > "out-of-the-box"?

recursos<sup>10</sup>. En otros casos, estos recursos están disponibles pero no para ser usados de forma de caja negra, lo cual es un escollo para investigadores que no sean expertos en NLP.

Como una pequeña contribución de esta tesis y con el objetivo de facilitar el acceso de estos recursos para la investigación, creamos el paquete *pysentimiento*<sup>11</sup>. Esta biblioteca provee modelos pre-entrenados y herramientas de preprocesado para textos sociales. Si bien tiene soporte multilingual tanto en español como inglés, su eje original es el de proveer recursos para el español que tiene una disparidad importante en recursos.

*pysentimiento* utiliza el model hub de *huggingface*<sup>12</sup>, un repositorio de modelos pre-entrenados basados en transformers. Allí es donde colocamos todos los modelos que entrenamos, tanto de sentimientos, emociones, y los que mostraremos más adelante como detección de discurso de odio. Cada tweet que es analizado por la librería pasa primero por una etapa de preprocesamiento (siguiendo el proceso explicado en la sección 3.5), y luego analizado por el modelo que nos brinda la salida correspondiente.

### 3.9. Discusión

Para las 3 tareas planteadas, los clasificadores basados en modelos pre-entrenados de transformers obtuvieron mejores resultados que los basados en redes recurrentes y feed-forward. Como es esperable (y se observa en la literatura) los modelos monolingüales (*RoBERTa*, *BERTin* y *BETO*) tienen una performance sensiblemente mejor el modelo multilingual *mBERT*. Dentro de los modelos de mejor performance, *RoBERTa* obtiene la mejor performance, aunque su mejora es pequeña respecto de *BETO*.

Algo que observamos es que, entre los modelos recurrentes y feed-forward que consumen word-embeddings, la utilización de representaciones entrenadas directamente sobre textos generados por usuario resultan en una mejor performance de los clasificadores. Si bien puede pensarse que el entorno pequeño o el texto ruidoso de los textos pueden ser un problema a la hora de construir representaciones, los experimentos realizados indican lo contrario. Retomaremos esta idea en el capítulo 7, donde por un lado generaremos un modelo basado en *RoBERTa* entrenado sobre tweets, y por otro lado observaremos si podemos replicar su performance intentando “adaptar” un modelo *BETO* a este nuevo dominio.

Finalmente, como un pequeño aporte – principalmente a la comunidad académica, y puntualmente aquella hispanoparlante – creamos una librería de análisis de sentimientos *pysentimiento* que provee modelos pre-entrenados y herramientas de preprocesado para textos sociales. En esta herramienta quedarán volcados todos los modelos entrenados de esta tesis.

<sup>10</sup> La definición de bajos recursos es subjetiva, pero tomando en cuenta la cantidad de hablantes nativos de español hay una desproporción abismal con otros idiomas

<sup>11</sup> <https://github.com/pysentimiento/pysentimiento>

<sup>12</sup> <https://huggingface.co/models>

### 3.10. Conclusiones

En este capítulo hemos hecho una introducción a la extracción de opiniones usando técnicas de clasificación basadas en redes neuronales. Analizamos tres problemas de extracción de opiniones en Español –análisis de polaridad, análisis de emociones y detección de ironía– y utilizamos modelos basados en redes neuronales y otros basados en modelos pre-entrenados. Presentamos el andamiaje básico para tareas de clasificación que utilizaremos en el resto de la presente tesis. En los siguientes capítulos centraremos nuestra atención en una tarea particular: la detección de discurso de odio.

Este sería bueno expandirlo.

### 3.11. Notas

Gran parte de este trabajo está basado en nuestra participación en TASS 2018[89] resumida en Luque and Pérez [86]. Los resultados en esta sección no son comparables con los de ese trabajo ya que decidimos utilizar la versión del dataset de TASS 2020[47], que por un lado unifica las dos posibles clases neutrales de TASS 2018 y además brinda el nuevo dataset de análisis de emociones. Omitimos el análisis de data augmentation y nos centramos en dar una breve introducción al tema y en analizar el impacto de los word-embeddings generados en textos sociales.

Dar más sobre cómo se relacionan estos estudios con el lenguaje de odio: similitudes y diferencias, por ejemplo

brinda? No entiendo.

Respecto de quella publicación,

## 4. DETECCIÓN DE DISCURSO DE ODIO

El discurso de odio contra mujeres, inmigrantes y otros grupos protegidos es un fenómeno generalizado en la Internet, y que resulta importante monitorizar dada su potencial relación con actos violentos, como hemos comentado en la introducción de esta tesis. En los primeros días de la World Wide Web, algunos académicos se aventuraron a decir a que los prejuicios y el odio serían removidos en este espacio mediante la disolución de identidades en el ámbito virtual [25, 82, 124]. Veinte años después de esta hipótesis, podemos decir que no ha sido el caso. La prevalencia del racismo en la "World Wide Web" y la explosión de esta en las redes sociales ha sido estudiada en numerosos trabajos [1, 73], como así también la misoginia en el mundo virtual [42, 88], entre otros ataques discriminatorios.

¿que es esto?

El discurso racista y sexista es una constante en las redes sociales, pero muchos picos se documentan después de eventos detonantes como asesinatos con motivos religiosos o políticos [24]. Debido a esto, algunos estados y organizaciones supranacionales han tomado cartas en el asunto instando a las empresas de redes sociales a que tomen medidas para bajar la incidencia de este fenómeno. Debido a la enorme cantidad de contenido generado por usuarios en las redes sociales, es necesario desarrollar herramientas que faciliten la labor humana en la detección y prevención del discurso de odio, con particular foco de aquel que incita a la violencia física.

En este capítulo haremos una introducción a este problema desde varias ópticas. Analizaremos las diversas definiciones de discurso de odio y haremos una breve reseña de este fenómeno desde un marco legal y de tratados internacionales para luego centrarnos en este problema desde una perspectiva del procesamiento de lenguaje natural. En base al dataset de la competencia hateEval [7], propondremos técnicas de detección de discurso de odio para las tareas propuestas, algunas de ellas presentadas en Luque and Pérez [86]. Finalmente, marcaremos algunos problemas en los enfoques actuales de la detección de discurso discriminatorio y algunas oportunidades de mejora que abordaremos en capítulos subsiguientes.

### 4.1. ¿Qué es el discurso de odio?

No existe una definición universalmente aceptada de lo que configura discurso de odio. En esta sección haremos en primer lugar un repaso muy general de algunos tratados internacionales sobre la materia para intentar aproximarnos a este concepto, a la vez que también haremos un recuento de las definiciones utilizadas en trabajos dedicados a abordar este problema mediante herramientas de NLP.

Una pequeña aclaración: en la normativa sobre derechos humanos muchas veces se encuentra bien delimitado el discurso **discriminatorio** del discurso de odio, siendo este último una subcategoría del primero de mayor intensidad y con incitaciones a la violencia contra grupos protegidos o individuos miembros de estos grupos. En la literatura de NLP sobre el tema se utiliza la expresión discurso de odio (*hate speech*) para referirse indistintamente a ambos fenómenos.

¿Por qué? Dejar algo tan  
confundente regresa cierta  
explicación.

Aún cuando entendemos que la acepción general del discurso de odio es incorrecta desde la perspectiva de tratados internacionales, teniendo en cuenta que esta tesis está centrada en técnicas para su detección automática usaremos esta terminología para plegarnos a los usos y costumbres de la comunidad de NLP.

#### 4.1.1. Abordaje desde una perspectiva legal y de los Derechos Humanos

Un principio general que hace a los derechos más elementales del hombre y a la vida en sociedad es la posibilidad de expresarse libremente, el derecho a la libre expresión. Este derecho está protegido por constituciones nacionales y numerosos tratados internacionales. Uno de estos tratados, el Pacto Internacional de Derechos Civiles y Políticos (*ICCPR* por sus siglas en inglés)<sup>1</sup>, sancionado en 1966 en la Asamblea de las Naciones Unidas y ratificado por 166 países, incluye en su artículo 19:

1. Nadie podrá ser molestado a causa de sus opiniones.
2. Toda persona tiene derecho a la libertad de expresión; este derecho comprende la libertad de buscar, recibir y difundir informaciones e ideas de toda índole, sin consideración de fronteras, ya sea oralmente, por escrito o en forma impresa o artística, o por cualquier otro procedimiento de su elección.
3. El ejercicio del derecho previsto en el párrafo 2 de este artículo entraña deberes y responsabilidades especiales. Por consiguiente, puede estar sujeto a ciertas restricciones, que deberán, sin embargo, estar expresamente fijadas por la ley y ser necesarias para:
  - a) Asegurar el respeto a los derechos o a la reputación de los demás;
  - b) La protección de la seguridad nacional, el orden público o la salud o la moral públicas. (Artículo 19 de la *ICCPR*)

Este artículo que garantiza el derecho a la libertad de expresión da cuenta también de que esta libertad no es completamente irrestricta. El ejercicio de los derechos e igualdad ante la ley de otros marca este límite, no pudiéndose utilizar esta libertad de expresión para avasallar los derechos de terceros. La Convención Internacional sobre toda forma de Discriminación Racial (*ICERD*)<sup>2</sup> dice en su artículo 4 al respecto:

Los Estados partes condenan toda la propaganda y todas las organizaciones que se inspiren en ideas o teorías basadas en la superioridad de una raza o de un grupo de personas de un determinado color u origen étnico, o que pretendan justificar o promover el odio racial y la discriminación racial, cualquiera que sea su forma, y se comprometen a tomar medidas inmediatas y positivas destinadas a eliminar toda incitación a tal discriminación o actos de tal discriminación y, con ese fin, teniendo debidamente

<sup>1</sup> Este pacto desarrolla los derechos civiles y políticos establecidos por la Declaración Universal de los Derechos Humanos de la ONU

<sup>2</sup> <http://servicios.infoleg.gob.ar/infolegInternet/anexos/120000-124999/122553/norma.htm>

en cuenta los principios incorporados en la Declaración Universal de Derechos Humanos, así como los derechos expresamente enunciados en el artículo 5 de la presente Convención, tomarán, entre otras, las siguientes medidas:

- a) Declararán como acto punible conforme a la ley, toda difusión de ideas basadas en la superioridad o en el odio racial, toda incitación a la discriminación racial así como todo acto de violencia o toda incitación a cometer tal efecto, contra cualquier raza o grupo de personas de otro color u origen étnico, y toda asistencia a las actividades racistas, incluida su financiación;
- b) Declararán ilegales y prohibirán las organizaciones, así como las actividades organizadas de propaganda y toda otra actividad de propaganda, que promuevan la discriminación racial e inciten a ella y reconocerán que la participación en tales organizaciones o en tales actividades constituye un delito penado por la ley;
- c) No permitirán que las autoridades ni las instituciones públicas nacionales o locales, promuevan la discriminación racial o inciten a ella. (Artículo 4, ICERD)

Los Estados y otros organismos deben tomar medidas para poder asegurar el libre ejercicio de los derechos y la igualdad de todos sus miembros, aún cuando esto pueda significar una restricción en la libertad de expresión [4]. Entendiendo entonces que el derecho a la expresión tiene sus límites, podemos pensar que el discurso de odio es una de esas fronteras. Si bien el discurso de odio es algo que no está completamente delimitado, repasaremos algunas definiciones de este fenómeno hechas en tratados para acercarnos un poco más a las características comunes que comparten las diferentes definiciones. La Observación General 35 del Comité por la Eliminación de la Discriminación Racial de la ONU (CERD) considera que será discurso de odio, y debe ser tipificado penalmente:

- a) Toda difusión de ideas basada en la superioridad o en el odio racial o étnico, por cualquier medio;
- b) La incitación al odio, el desprecio o la discriminación contra los miembros de un grupo por motivos de su raza, color, linaje, u origen nacional o étnico;
- c) Las amenazas o la incitación a la violencia contra personas o grupos por los motivos señalados en el apartado anterior;
- d) La expresión de insultos, burlas o calumnias contra personas o grupos, o la justificación del odio, el desprecio o la discriminación por los motivos señalados en el apartado b) anterior, cuando constituyan claramente incitación al odio o a la discriminación;
- e) La participación en organizaciones y actividades que promuevan e inciten a la discriminación racial. (Recomendación 35 del Comité por la Eliminación de la Discriminación Racial, CERD)

En líneas generales, como se menciona en el reporte de la CIDH sobre discurso de odio contra lesbianas, gay, trans e intersex en Latinoamérica [29], el concepto usualmente es referido a expresiones que incitan a tomar algún tipo de medida hostil contra una víctima o un grupo de personas, siendo esta perteneciente a un determinado grupo social definido por alguna característica. Dicho esto, podría delimitarse el discurso discriminatorio del discurso de odio por la componente de la promoción e instigación de la violencia; sin embargo, para los fines de este trabajo utilizaremos los términos indistintamente. Aún cuando el discurso no contenga arengas ni incitaciones a cometer actos violentos, puede entenderse ese discurso como generador de un ambiente hostil y de intolerancia que termine promoviendo estos ataques físicos [29].

Article 19 [4] condensa muchas de estas definiciones de una manera succinta, desglosando esto en "odio" y "discurso":

1. Odio: emoción intensa e irracional de oprobio, enemistad y aborrecimiento hacia una persona o grupo de personas, por tener determinadas características protegidas (reconocidas en el derecho internacional), reales o percibidas. El "odio" es más que un mero prejuicio y debe ser discriminatorio. El odio es una muestra de un estado emocional u opinión y, por lo tanto, se diferencia de cualquier acto o acción que se haya llevado a cabo.
2. Discurso: cualquier expresión que vierta opiniones o ideas, que comparte una opinión o una idea interna con un público externo. Puede adoptar muchas formas: escrita, no-verbal, visual o artística y puede ser difundida en los medios, incluyendo Internet, material impreso, radio o televisión. (Article 19: Hate Speech Toolkit)

En base a esta definición, puede entenderse al discurso de odio como un discurso de cierta intensidad e irracionalidad que ataca a una persona o un grupo de personas por alguna característica históricamente vulnerable: por ser mujer, por su género, por su etnia, nacionalidad, religión, idioma, etc. La clave está en la combinación: un discurso irracional e intenso contra alguien que no posea una característica protegida no configura discurso de odio; por ejemplo, ataques a ciertas personas por ser periodistas. La figura 4.1 ilustra esta definición.

No todo ataque a un individuo o una persona de algún colectivo discriminado es discurso de odio. En particular, la CIDH [29] menciona en base al informe de la UNESCO sobre discurso de odio [45] que:

(...) el discurso de odio no puede abarcar ideas amplias y abstractas, tales como las visiones e ideologías políticas, la fe o las creencias personales. Tampoco se refiere simplemente a un insulto, expresión injuriosa o provocadora respecto de una persona. Así definido, el discurso de odio puede ser manipulado fácilmente para abarcar expresiones que puedan ser consideradas ofensivas por otras personas, particularmente por quienes están en el poder, lo que conduce a la indebida aplicación de la ley para restringir las expresiones críticas y disidentes. Asimismo, el discurso de odio tiene que distinguirse de aquellos "crímenes de odio" que se basan en conductas

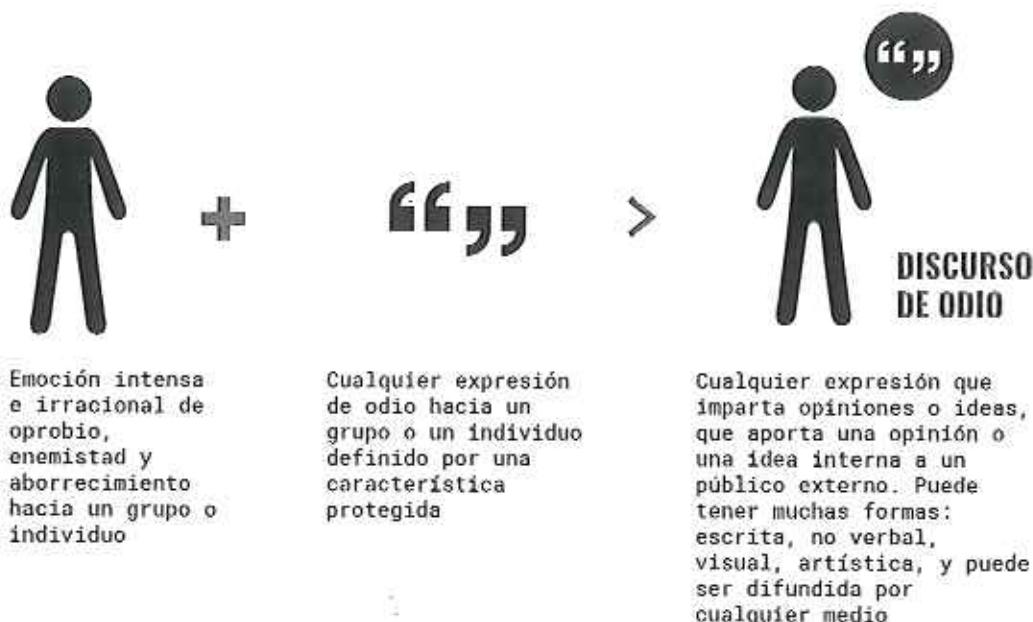


Fig. 4.1: Definición de discurso de odio de acuerdo al Toolkit de Article 19

expresivas, como las amenazas y la violencia sexual, y que se encuentran fuera de cualquier protección del derecho a la libertad de expresión

Como vemos, no sólo es difusa la frontera fijada ~~la característica~~ sobre qué es discurso de odio o insultos, sino que incluso también es difícil definir qué característica es protegida o no. En el siguiente capítulo hablaremos más de esto al describir los criterios utilizados a la hora de anotar un conjunto de datos sobre comentarios en Twitter.

#### 4.1.2. Definiciones utilizadas desde NLP

#### 4.2. Trabajo previo

En esta sección haremos una breve reseña de la literatura de la detección de discurso de odio y otros fenómenos similares. Un análisis exhaustivo de esta subdisciplina escapa al alcance de esta tesis debido a la enorme cantidad de trabajo del área con un ritmo meteórico en los últimos años. Referimos para repasos más extensivos esté interesado a Schmidt and Wiegand [132] y Fortuna and Nunes [43]. Más recientemente, Poletto et al. [119] hace un análisis pormenorizado y actualizado de los recursos existentes para esta tarea.

La detección del discurso del odio es una tarea de clasificación de oraciones relacionada con el análisis de sentimientos y ha sido estudiada para varias redes sociales [105, 130, 141]. Uno de los primeros trabajos al respecto es el de Greevy and Smeaton [55], quienes utilizan bolsas de palabras y Support Vector Machines para detectar

Yo no creo que ~~el discurso~~ escape al alcance realmente.  
Declo de otra forma: "Un análisis exhaustivo sera prácticamente inviable"  
por ejemplo.

6/200

contenido racista en páginas web, utilizando un dataset construido de manera semi-supervisada buscando sitios mediante keywords y sus links en motores de búsqueda. Siguiendo un enfoque similar, Warner and Hirschberg [144] usó unigramas y Brown clusters [22] con SVM para detectar mensajes antisemitas en Twitter.

Waseem and Iovy [147] anotaron un corpus y usaron técnicas basadas en n-gramas de caracteres para detectar comentarios de odio. Badjatiya et al. [5] usaron el mismo conjunto de datos para entrenar modelos de aprendizaje profundo con embeddings ajustados a los datos. Zhang et al. [158] entrenaron una red neuronal profunda que combina CNN con Gated Recurrent Units [28], superando a los sistemas anteriores en varios conjuntos de datos de detección de discurso de odio. Anzovino et al. [2] recopilaron un corpus de tweets misóginos y propuso una taxonomía para distinguirlos en diferentes categorías. Los autores propusieron una serie de técnicas diferentes para clasificarlos, mostrando que enfoques simples (como el uso de modelos lineales junto con n-gramas) logran un rendimiento competitivo en conjuntos de datos de pequeño tamaño.

En cuanto a las tareas compartidas, Fersini et al. [40] presentó un shared task en la detección de misoginia en Twitter, tanto en español como en inglés, mientras que Fersini et al. [41] planteó un desafío similar pero en italiano e inglés. Bosco et al. [21] propuso un concurso de detección automática sobre publicaciones de Twitter y comentarios de Facebook, que incluía discursos de odio en general.

Una de las herramientas más utilizadas, no sólo para la detección de discurso de odio sino para la detección de contenido tóxico en general es Perspective API de Google, desarrollada originalmente por Jigsaw<sup>3</sup>. Esta API de acceso libre brinda un analizador muy potente para la detección de lenguaje tóxico, con información granular sobre los tipos de ataques. Algunos trabajos lo utilizan como algoritmo de detección en modalidad zero-shot, obteniendo mejores resultados que modelos entrenados sobre los propios datos [109]. Sin embargo, algunas de sus debilidades han sido marcadas mediante ejemplos adversariales, algo que obviamente es propio de las actuales limitaciones de las técnicas de NLP [66, 71].

Dentro de los trabajos en español, del Arco et al. [33] evalúa distintos modelos pre-entrenados de lenguaje sobre la tarea de detección de discriminación usando dos datasets: el primero, Pereira-Kohatsu et al. [113] que consta de 6000 tweets, recolectado por el Estado Español para monitorear el discurso de odio en redes sociales; y el segundo, el dataset de SemEval 2019 Task 5 (hatEval) [7], presentado en contexto de una shared-task y que comprende ataques contra inmigrantes y mujeres.

### 4.3. Descripción del dataset

El dataset que utilizamos en este capítulo es el provisto por Basile et al. [7]. Este dataset está orientado a la detección de discurso de odio contra mujeres e inmigrantes en Twitter, tanto en inglés como en español. Nuestro trabajo estará centrado en el dataset en español, aunque también propondremos modelos de clasificación para el problema en inglés. Las instancias del dataset poseen las siguientes etiquetas:

<sup>3</sup> <https://developers.perspectiveapi.com/s/>

Aquí vendría bien contextualizar con respecto al dataset del Capítulo 5: así usaron este dataset, más adelante se mencionan otros datasets más nuevos.

Categoría	Español			Inglés		
	Train	Dev	Test	Train	Dev	Test
No HS	2,643 (58.7 %)	278 (55.6 %)	940 (58.8 %)	5,217 (58.0 %)	573 (57.3 %)	1,740 (58.0 %)
IIS	1,857 (41.3 %)	222 (44.4 %)	660 (41.2 %)	3,783 (42.0 %)	427 (42.7 %)	1,260 (42.0 %)
TR	1,129 (60.8 %)	137 (61.7 %)	423 (64.1 %)	1,341 (35.4 %)	219 (51.3 %)	529 (42.0 %)
AG	1,502 (80.9 %)	176 (79.3 %)	474 (71.8 %)	1,559 (41.2 %)	204 (47.8 %)	594 (47.1 %)
Total	4,500	500	1,600	9,000	1,000	3,000

Tab. 4.1: Números del dataset de Basile et al. [7], por idioma y por partición. No HS representa los tweets que no tienen contenido odioso, HS aquellos que sí, TR aquellos que son individualizados, y AG aquellos que son agresivos. Entre paréntesis encontramos los porcentajes de incidencia, considerando TR y AG dentro de aquellos que son discriminatorios

Texto	HS	TR	AG
Los tomas así puro como si fuera jugo y cuando te querés rescatar estas hablando en árabe URL	0	0	0
Como son españoles nada... sin fueran refugiados...GLORIA # migrates # refugiados # EspañaLoPrimero URL	1	0	0
@OmarPrietoGob "Extranjero sin identificación será puesto en la frontera" ENVÍA AL EJERCITO A TOMAR CONTROL DE LAS PULGAS PLATANEROS Y CURVA DE AHÍ PARA QUE VEAS COMO HAY COLOMBIANOS INDOCUMENTADOS COMO MONTE AHÍ DE BUHONERS PORQUE LA POLICÍA	1	0	1
Immigrante da una brutal paliza a una joven por no dejarse besar en Ciudad Real.#stopinvasion #YoSiTeCreo #NoesNo lo peor que no han salido a la calle las feminas del Twitter que tanto se indignaron con la salida de La Manda a la calle.	1	1	0
@elisacarrio Callate hija de puta gorda falopera	1	1	1

Tab. 4.2: Ejemplos del dataset de SemEval 2019 Task 5: HatEval. HS indica la presencia de discurso de odio, TR la presencia de discriminación individualizada, y AG la presencia de discriminación agresiva

- **HS:** una etiqueta binaria que marca si el tweet tiene contenido discriminatorio (0 si no lo tiene, 1 si hay discurso de odio contra mujeres o inmigrantes)
- **Target:** Si hay IIS, una etiqueta binaria que marca si el objetivo del discurso de odio es un objetivo genérico (0) o si se refiere a un individuo específico (1)
- **Agresividad:** Si hay IIS, una etiqueta binaria que marca si el tweet es agresivo

La tabla 4.1 posee los números del dataset. Podemos observar entre los dos idiomas que, si bien la proporción de discurso de odio se mantiene muy similar en ambos idiomas (58% vs 42% aproximadamente), la proporción de discurso de odio individualizado (TR) y agresivo (AG) es notoriamente más alto para el español. Esto puede deberse, entre otras cosas, a distintas estrategias de recolección de los datasets. La tabla 4.2 posee algunos ejemplos para cada una de las características en cuestión.

#### 4.4. Tareas de clasificación propuestas

Sobre el dataset de hatEval, los autores propusieron dos tareas:

- **Tarea A:** Dado un tweet predecir si contiene discurso de odio contra mujeres o inmigrantes (HS)
- **Tarea B:** Dado un tweet, predecir si contiene discurso de odio (HS), si está dirigido contra un individuo o un grupo (TR), y si es agresivo o no (AG)

La primer tarea es la versión clásica de la detección de discurso de odio, donde predecimos una etiqueta binaria. La segunda es una versión más rica, de grano fino, donde predecimos varias características de particular interés para distinguir algunas formas potencialmente más peligrosas de este fenómeno: por ejemplo, si es agresivo y si es individualizado, lo que puede indicar alguna incitación a un ataque.

La performance de Tarea A es medida mediante la Macro F1 de las 2 clases positiva y negativa. En el caso de Tarea B, se mide por la Macro F1 de las 3 clases (HS, TR, AG) y también por la medida Exact Match Ratio:

$$EMR = \frac{1}{n} \sum_{i=1}^n I(Y_i, Y_i^*)$$

siendo  $Y_i$  las etiquetas respectivas (*HS, TR, AG*),  $Y_i^*$  las etiquetas que predice nuestro sistema, e  $I$  la función indicadora ( $I(x, x) = 1$ ; 0 en cualquier otro caso). Observado más de cerca, esto puede entenderse la accuracy sobre la 3-upla de la salida de los clasificadores, pero usamos la terminología de Exact Match Ratio como Zhang and Zhou [156] para referirnos a esta métrica.

## 4.5. Método

### 4.5.1. Preprocesamiento

Definimos dos niveles de preprocesamiento: preprocesamiento básico y orientado a sentimientos, dependiendo del modelo a utilizar. El preprocesamiento básico de tweets es el mismo que describimos en la sección 3.5, y es el que utilizaremos con los modelos pre-entrenados o modelos neuronales.

El preprocesamiento orientado a sentimientos incluye además lematización (usando TreeTagger [131]) y manejo de negación. Para el manejo de la negación, seguimos un enfoque simple: Buscamos palabras de negación y agregamos el prefijo 'NOT' a los siguientes tokens. Se niegan hasta tres tokens, o menos si se encuentra un token que no sea una palabra. Este preprocesamiento es utilizado en el modelo de SVM.

### 4.5.2. Modelos de clasificación

Para las tareas propuestas, analizamos el desempeño de diversos modelos de clasificación. Algunos de ellos son los presentados para la shared-task HatEyal en Luque and Pérez [86], a las cuales agregamos modelos basados en transformers.<sup>4</sup> Para la tarea de detección binaria (Tarea A) planteamos 3 tipos de clasificadores:

<sup>4</sup> Estos modelos no estaban disponibles al momento de presentar dicho trabajo. El trabajo de *BERT* [35] es de finales de 2018, y hasta finales de 2019 no fue publicada una versión entrenada en español, *BETO*.

- Modelos lineales: regresiones logísticas y SVM con kernel lineal<sup>5</sup>, consumiendo como entrada bolsas de palabras, bolsas de caracteres, y tweet embeddings ( )
- Redes neuronales recurrentes: usando como entrada word-embeddings y embeddings contextualizados (ELMo)
- Modelos pre-entrenados basados en LM: ídem sección 3.6.

Para los modelos lineales, utilizamos tweet embeddings calculados con SIF (ver sección 2.2.1) para más detalles). Utilizamos como base embeddings de *fastText* entrenados sobre tweets utilizando el preprocessado orientado a sentimientos descripto en la sección 4.5.1. Para el resto de los modelos, utilizamos el preprocessado básico.

Para el caso de la tarea de multidetección (**Tarea B**), podemos pensar este problema de dos maneras:

1. Un problema de clasificación múltiple
2. Un problema de clasificación de 5 clases

En el primer caso, el enfoque es el de predecir por separado HS, AG, y TR. La segunda formulación se basa en observar que no todas las 8 combinaciones son permitidas, sino sólo 5: si no hay HS no nos interesa observar las otras dos variables. Tenemos entonces 5 clases a predecir.

Con esta última observación, propusimos en Luque and Pérez [86] un modelo basado en SVM (consumiendo la misma entrada que detallamos anteriormente). No evaluamos en dicho trabajo un modelo recurrente con este esquema de clasificación, ni tampoco lo haremos aquí, considerando que evaluaremos modelos que han demostrado tener mejor performance para numerosas tareas de clasificación de texto.

Así mismo, proponemos para esta subtarea modelos basados en transformers basado en multi-clasificación. La arquitectura usual de modelos basados en BERT para clasificación constan de poner como última capa una “cabeza” que consume la salida del token [CLS]. Esto agrega una matriz de parámetros  $W \in \mathbb{R}^{m \times 768}$  donde  $m$  es la cantidad de clases de nuestro problema y 768 corresponde a la dimensión de cada vector del modelo de transformers, y usando softmax como función de activación.

Para construir un modelo de multi-clasificación, mantenemos la misma arquitectura pero, en lugar de usar como activación la función softmax, utilizamos la función sigmoida elemento a elemento. En el caso de clasificación de  $n$  clases,  $\text{softmax}(Wx + b)$  nos da para el elemento  $i$ -ésimo el score de que la instancia pertenezca a la clase  $i$ . Por otro lado, en el caso de multicasificación de  $n$  variables,  $\sigma(Wx + b)^5$  nos da el score de predecir la etiqueta positiva para el (en nuestro caso,  $\sigma(Wx + b)_1$  nos da  $P(HS = 1 | x)$ ,  $\sigma(Wx + b)_2$  nos da  $P(TR = 1 | x)$ , etc). La figura 4.2 ilustra el modelo utilizado.

Para entrenar el modelo de clasificación, evaluamos dos tipos de funciones de costo. En primer lugar, utilizamos la suma o promedio<sup>6</sup> de las entropías cruzadas

<sup>5</sup> Esta expresión es elemento a elemento

<sup>6</sup> Es equivalente optimizar una u otra

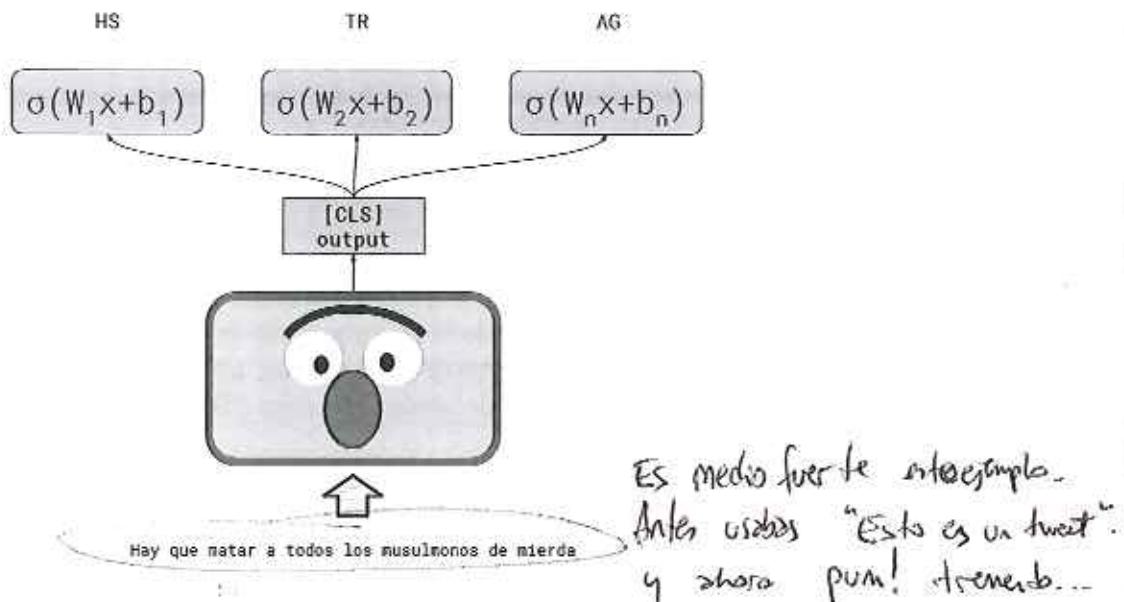


Fig. 4.2: Modelo basado en BERT para la tarea de clasificación múltiple. Cada variable (HS, TR, AG) representa un problema de clasificación en sí mismo

binarias. Concretamente, si  $y = (y_{HS}, y_{TR}, y_{AG})$  son las etiquetas de una instancia e  $\hat{y}$  la predicción del modelo, la función de costo es:

$$L(y, \hat{y}) = \sum_{k \in \{HS, TR, AG\}} J(y_k, \hat{y}_k) \quad (4.1)$$

donde  $J$  es la entropía cruzada. Esta función de costo, sin embargo, ignora cualquier tipo de jerarquía entre las variables; por ejemplo, si para una instancia tenemos  $HS = 0$ , calcula el costo también de las variables  $TR$  y  $AG$ . Contemplamos entonces una variante de esta función para tener en cuenta esta jerarquía:

$$L(y, \hat{y}) = J(y_{HS}, \hat{y}_{HS}) + \beta(y_{HS}) \sum_{k \in \{TR, AG\}} J(y_k, \hat{y}_k) \quad (4.2)$$

donde  $\beta(y_{HS})$  pondera la pérdida de las variables del segundo nivel de nuestra jerarquía. Una opción puede ser considerar  $\beta(1) = 1, \beta(0) = 0$ , donde ignoramos las pérdidas de las variables  $TR, AG$  cuando no hay discurso discriminatorio. Análogamente,  $\beta(y) = 1$  sería el caso descripto en la ecuación 4.1. Una forma de generalizar esto es agregando un hiperparámetro  $\gamma \in [0, 1]$  para escribir  $\beta(y) = (1 - y)\gamma + y$ .

Las evaluaciones de los modelos las realizamos poniendo una máscara por encima de estos modelos de clasificación múltiple de manera de evitar salidas incoherentes (por ejemplo,  $HS = 0, TR = 1, AG = 0$ ).

Modelo	Idioma	Precision	Recall	F1	Macro-F1
SVM*		0.639	0.800	0.711	0.730
ELMO-RNN	es	0.661	0.753	0.704	0.735
<i>BETO</i>		<b>0.674</b>	<b>0.839</b>	<b>0.747</b>	<b>0.764</b>
bert		0.474	<b>0.968</b>	0.637	0.496
roberta	en	0.470	0.967	0.632	0.486
bertweet		<b>0.495</b>	0.959	<b>0.653</b>	<b>0.546</b>

Tab. 4.3: Resultados de la evaluación para la detección de discurso de odio en datasets de desarrollo y test, medidas por precisión y sensitividad sobre la clase positiva (discurso de odio) y por la métrica macro F1. Con \* están marcados los resultados presentados en Luque and Pérez [86]. En negrita, el mejor resultado.

## 4.6. Resultados

La tabla 4.3 muestra los resultados de evaluación para los modelos propuestos en la detección de discurso de odio binaria (Tarea A). Marcamos con un asterisco aquellos modelos presentados en Luque and Pérez [86]. Respecto a los resultados en español, el modelo basado en SVMs obtiene una buena performance, aún contra aquel basado en embeddings contextualizados, habiendo obtenido el mejor desempeño en la competencia con 0,730 de Macro F1. La pobre performance de ELMO contra un modelo mucho más simple puede deberse a un mal pre-entrenamiento del modelo base <sup>7</sup> y también debido al cambio de dominio, a los cuales los modelos pre-entrenados previos a BERT son sumamente sensibles [63].

Para ambos idiomas, los modelos basados en transformers [142] obtienen la mejor performance, con considerables mejoras respecto a los modelos basados en ELMO y a los SVMs. Particularmente, en el caso del inglés, *BERTweet* [102] obtiene la mejor Macro-F1. En el capítulo 7 presentaremos un modelo similar a *BERTweet* para español que mejora la performance sobre *BETO*, a la vez que evaluaremos sobre versiones de *BETO* ajustadas al dominio social <sup>8</sup>.

La tabla 4.4 muestra los resultados de la Tarea B, reportado por las F1 de cada variable predicha (HS, TR, AG), así como por el Macro F1 de HS y el Macro F1 de las 3 variables mencionadas. Los resultados están expresados como la media de 10 corridas independientes del experimento para cada configuración distinta. Consideramos las 3 versiones: *multi* refiere a clasificación múltiple, *hier* a clasificación múltiple con la función de costo jerárquica, y *combi* a la conversión del problema en una clasificación de 5 clases.

Podemos observar que para español, la mejor performance en términos de EMR (la métrica más estricta) es el clasificador entrenada con la función de costo definida en 4.2 (con el hiperparámetro  $\gamma = 0,1$ ); sin embargo, ~~esta diferencia~~ la diferencia entre las performances no es significativa al correr un test de Kruskal-Wallis ( $H(9) = 3,492, p = 0,174$ ). En términos de Macro-F1, la mejor performance es de *BETO* con

<sup>7</sup> No queda claro que en entrenar este modelo sobre 20M palabras sea suficiente, ni que sea un dataset suficientemente general

<sup>8</sup> Un modelo que no evaluamos en el presente trabajo es la versión en español de RoBERTa, recientemente entrenada. En el capítulo 7 evaluaremos su performance

Modelo	Idioma	IHS F1	TR F1	AG F1	EMR	Macro F1
<i>BETO</i>	multi	0.741	0.765	0.688	0.685	0.731
	hier	0.735	0.758	0.674	<b>0.703</b>	0.722
	combi	<b>0.742</b>	0.763	0.668	0.698	0.721
<i>BERT</i>	multi	0.638	0.600	0.443	0.380	0.560
	hier	0.642	0.592	0.451	0.388	0.562
	combi	0.644	0.593	0.442	0.398	0.560
<i>RoBERTa</i>	multi	0.634	0.578	0.454	0.365	0.555
	hier	0.637	0.572	0.456	0.370	0.555
	combi	<b>0.636</b>	0.576	0.442	0.377	0.551
<i>BERTweet</i>	multi	0.658	0.629	<b>0.462</b>	0.426	<b>0.583</b>
	hier	0.656	0.617	0.450	0.423	0.574
	combi	<b>0.666</b>	<b>0.637</b>	0.444	<b>0.449</b>	0.582

Tab. 4.4: Resultados de la evaluación para para Tarea B en términos de las F1 de las clases HS (Hate Speech), TR (Targeted), AG (Aggressive), el Exact Match Ratio (EMR), las Macro F1 de las clases en cuestión, y la Macro F1 de la clase HS. Las 3 variaciones de los modelos son: *multi* es la salida de multiclasiificación estándar, *hier* es la salida de multiclasiificación con una jerarquía de clasificación, y *combi* es la salida de multiclasiificación con una combinación de clasificaciones. Los resultados están expresados como las medias de 10 corridas independientes.

Modelo	Idioma	Tarea	Precision	Recall	F1	Macro F1
<i>BERTweet</i>	en	A	$0.495 \pm 0.012$	$0.959 \pm 0.012$	$0.653 \pm 0.009$	$0.546 \pm 0.027$
		B	$0.505 \pm 0.011$	$0.948 \pm 0.018$	$0.658 \pm 0.005$	$0.567 \pm 0.022$
<i>BETO</i>	es	A	$0.674 \pm 0.021$	$0.839 \pm 0.026$	$0.717 \pm 0.007$	$0.764 \pm 0.011$
		B	$0.713 \pm 0.042$	$0.778 \pm 0.054$	$0.741 \pm 0.013$	$0.771 \pm 0.015$

Tab. 4.5: Comparación de la performance sobre la detección de discurso de odio para los clasificadores entrenados sobre Tarea A y Tarea B. Resultados expresados como la media de 10 corridas independientes del experimento junto a sus desviaciones estándar. Ambos clasificadores de la Tarea B están entrenados sobre el problema de multi-clasificación

el problema de clasificación múltiple y sin la función de costo jerárquica, pero de nuevo esta diferencia no es significativa ( $H(9) = 3,656, p = 0,16$ ).

Respecto al inglés, los mejores resultados pueden observarse en el modelo entrenado con *BERTweet* con la salida de 5 clases en el caso del EMR, y con la salida múltiple (sin pérdida jerárquica) para la Macro-F1. Este resultado, sin embargo, queda en términos de EMR por debajo del baseline, y cercano en términos de Macro F1 a los mejores resultados de la competencia. En Gertner et al. [49], se basaron en un ensemble de modelos entrenados con BERT y usando también un ajuste de dominio sobre tweets. Esta baja performance de nuestros modelos (y de los modelos en general sobre este dataset) puede deberse a problemas de anotación y que las particiones de train y test no son idénticamente distribuidas.<sup>9</sup>

Lejos de dañarse la performance de la detección de lenguaje discriminatorio (lo que analizamos en la Tarea A), predecir más de una variable parecía mantener

<sup>9</sup> En Gertner et al. [49] dan evidencia de esto en su trabajo algo que perjudica el desempeño de estos modelos

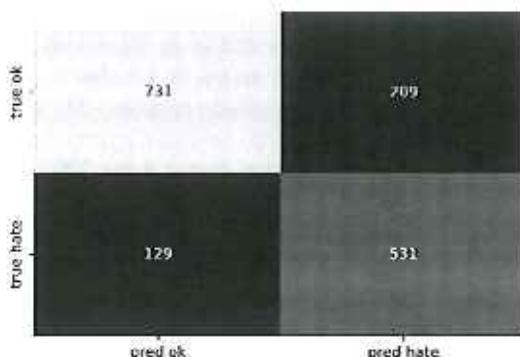


Fig. 4.3: Matriz de confusión para la detección de discurso de odio sobre un ensemble de voto mayoritario de 10 clasificadores entrenados para la Tarea B

el desempeño e incluso mejorarlo levemente. La tabla 4.5 muestra la comparativa para la detección de discurso de odio (HS) para aquellos clasificadores que obtuvieron mejores resultados para Tarea A y Tarea B (en ambos casos, *BETO* y *BERTweet*). En ambos casos, hay ligeras diferencias ~~para~~ a favor de uno u otro clasificador pero no parecen ser significativas.

#### 4.6.1. Análisis de Error

Para tener una mejor idea de lo ocurrido con nuestros clasificadores, realizamos un análisis de error sobre los datasets en español. Tomamos las salidas de 10 clasificadores BETO entrenados con una distinta semilla y analizamos el error sobre un clasificador de ensemble por voto mayoritario para disminuir los efectos de la varianza de los clasificadores. De esta manera, tratamos de buscar aquellos errores frecuentes, aquellos que la mayoría de los 10 clasificadores erran. Nos centraremos en observar los errores de la etiqueta HS, y particularmente veremos qué pasa con los falsos positivos. La figura 4.3 muestra la matriz de confusión de nuestros clasificadores sobre el dataset de test. En base a esto, haremos éje en analizar los falsos positivos.

*¿por qué  
introducen más  
los FP?  
↓  
son FN?*

En la tabla 4.6 podemos observar algunos de los errores que comete nuestros clasificadores. Por un lado, podemos observar que algunos errores se deben a cierto overfitting a ciertas palabras “clave” (como las nro. 1, 2 y 3) muchas de las cuales son producto del proceso de recolección que está fuertemente basada en keywords (inmigrante, sudaca, por ejemplo). Haciendo un poco de probing<sup>10</sup> en los clasificadores, podemos ver que ciertas palabras como “sudaca” o “migrante” automáticamente disparan una salida positiva en discurso de odio<sup>10</sup>. Otros de los casos que nuestros clasificadores no parecen detectar son los reportes de discriminación (como el caso 5, parcialmente también el 1) o casos de apropiación cultural (caso 3 o 5) o contenido irónico (casos 3, 4, 7 y 8). Estos casos dan cuenta de la dificultad de la detección de esta tarea.

<sup>10</sup> Puede probarse esto en el cuadro interactivo de <https://huggingface.co/pysentimiento/robertuito-hate-speech?text=migrante>. Si bien este modelo es el que presentaremos en el capítulo 7, el problema se mantiene.

Índice	Tweet
1	Una cosa que le pasó a una amiga mexicana en Barcelona. Una chica repartía flyers en castellano y catalán, le dio uno a mi amiga en catalán y al segundo se lo quitó y se lo dio en castellano. -No, déjalo que entiendo catalán -Ah, es que como te he visto cara de sudaca Racismo 0 URL
2	Los fascistas salen de la ratonera para atacar a los 100 inmigrantes que han logrado saltar la valla de Ceuta. Asco de prensa y partidos políticos que se pelean por lograr los votos de los fascistas. Ningún ser humano es ilegal.
3	@saludhernandezm Esas personas que extrañan dictaduras, cualquier dictadura, aún más si son las ibéricas que ponen como eje a Madrid; su cultura, política e idioma sobre la idiosincrasia de los demás. Esos mismos que añoran a Franco y viven en un país 'Sudaca' creyéndose dueños de la verdad.
4	@cacienna después sigo siendo un sudaca pero bueno
5	@ArturoMonteduro Pues tienes toda la razón del mundo. Yo lo peores que seguro que tenía papeles y DNI español, pero queda mejor decir "rgelino." jinete d'estos del top manta", "puto inmigrante", o "moro mierda ya pucs matás dos pájaros de un tiro.
6	Obvio me ofendo cuando algu Sudaca hace algu comentario presuponiendo que los mexicanos somos feos, o que el pas est de la verga. Entre mexicanos podemos hablar mierda de Mxico, pero que a ningn pinche extranjero se le ocurra, porque va a haber pedo!
7	TODOS LOS INMIGRANTES Y GITANOS FUERA!!! Menos: el colombiano que me vende coca, el negro que me consigui putas, el moro que me pasa costo y el gitano que me vende maría.
8	Ayer nos fuimos a tomar algo con los cumpas: Dos españoles, un ponja, un africano y un sudaca. Estamos para campaña de United Colours of Benneton.

Tab. 4.6: Falsos positivos del modelo de clasificación para la detección de discurso de odio (HS).

*no son falsos negativos ?? No entiendo, Estos fueron clasificados como HS=1 pero eran HS=0?*

Algo que observamos también es que hay una fuerte cantidad de errores de etiquetado. La tabla 4.7 muestra algunos de estos ejemplos, todos estos etiquetados incorrectamente como discurso de odio. Si bien es difícil establecer una causa específica para cada uno de estos errores, es posible que sean causados por una combinación de:

- Confusión entre lenguaje ofensivo y discurso de odio (ejemplos 1, 2, 3, 4 y quizás 5)
- Desconocimiento cultural (ejemplo 7, quizás 10 también)
- Opinión pero no necesariamente discurso de odio (ejemplo 9)
- Tweets truncados (el ejemplo 6 fue mal recolectado y está truncado) *No entiendo, ¿fue truncado?*

Dentro de estas asignaciones posiblemente erróneas, en el análisis de error separamos un subconjunto especial sobre el cual entendemos que falta contexto para asignar una etiqueta. Si bien esto a veces puede ser inferido por un humano (por ejemplo, infiriendo que un comentario habla de tal o cual suceso), un algoritmo de detección podría beneficiarse de esta información. Para el dataset en concreto, muchos de estos comentarios tienen un contexto implícito: hablan de incidentes en la frontera de Ceuta (España) entre inmigrantes y la policía, debido al proceso de

Índice	Tweet
1	Callate la puta boca pesada @Lauu tb
2	me cansada tía cállate ya la puta boca lo que te queda de vid
3	FLOR SALTANDO AHÍ LA RE PUTA MADRE TE LO MERECES PEDAZO DE FORRA TE LO MERECES, EN LA CARA DE TODOS HIJA ASÍ SE HACE
4	Callaté cerrá el orto un ratoool es mas, no te querés morir mejor? la puta que te parió
5	Esta mujer es lo más la puta madre URL
6	@csdb530 @carvasar Claro, porque la culpa la tienen las niñas embarazadas y las víctimas de acoso sexual, violación... URL
7	#Pendejos Don't call me gringo You fuckin beaner Stay on your side Of that goddamn river Don't call me gringo You beaner No me digas beaner Mr. puñetero Te sacaré un susto Por racista y culero No me llames frijolero Pinche gringo puñetero
8	Mónica que te calles la puta boca #OTGala7
9	@pablocasado_ @imbrodamelilla @TeoGarciaEgea pablo no quieren concertinas , un muro de 12 metros o mas de altura de dos plantas de pisos ,se acabo los problemas @guardiacivil @policia @EMADinde @MonarquiaEspaña nadie se quejaría de nada
10	#OTGala7 Noemí JAJAJAJAJA ESTA MUJER ES LA PUTA AMA

Tab. 4.7: Ejemplos mal etiquetados como discurso de odio. En etiqueta marcamos cómo están etiquetados (erróneamente). El índice es meramente para referencia.

→ por humanos?

recolección realizado. La tabla 4.8 muestra algunos de estos ejemplos. En algunos ejemplos podemos observar que falta contexto conversacional: aquellos donde tenemos un @uscriname al principio del tweet indican que son una respuesta. En otros casos, falta información de un suceso o noticia sobre la que se está comentando: los que terminan con URL suelen ser citas de otros tweets o links a noticias.

#### 4.7. Discusión → Muy buena

Respecto a la performance de los modelos presentados, los modelos basados en transformers son notoriamente superiores a los demás modelos, en ambas tareas c idiom as. Particularmente, en inglés podemos observar que aquellos pre-entrenados sobre tweets (*BERTweet*) tienen mejor performance que aquellos que son pre-entrenados sobre wikipedia como BERT o RoBERTa.

Sobre la tarea más difícil de detección múltiple de discurso de odio (Tarea B), propusimos varios enfoques: uno basado en predecir cada variable por separado y otro en predecir una variable que indique la combinación en cuestión. El modelo de predicción múltiple entrenado con la función de costo jerárquica 4.2 obtuvo la mejor performance en términos de EMR, y la de multi-clasificación obtuvo la mejor en términos de Macro F1. En el caso de inglés, el modelo entrenado sobre 5 clases obtuvo la mejor performance en EMR y de nuevo el de multi-clasificación sobre Macro F1; sin embargo, esta dista de la mejor performance de la competencia (obtenida por el equipo MITRE [49]) que usa una combinación de técnicas, algunas de las cuales veremos en el capítulo 7 <sup>11</sup>. De estos dos casos, el modelo de multi-clasificación corre con la ventaja de calcular cada variable de manera independiente y tener un hiperparámetro menos.

Cincos

<sup>11</sup> En ese trabajo hacen un ensemble de clasificadores BERT adaptados a dominio

Etiqueta	Tweet
No HS	<p>- @ArturoMonteduro Pues tienes toda la razón del mundo. Y lo peor es que seguro que tenía papeles y DNI español, pero queda mejor decir "argelino" o "uno d'estos del top manta", "puto inmigrante", o "moro mierda" y ya pues matas dos pájaros de un tiro.</p> <p>@UbedaJoseA @WalkingDead_AMC Perro cállate ala puta  @iancandado Es sudaca como.tu</p> <p>@Stopfalsasdemun Pero que tienes en la cabesaaaa?(con voz de sudaca de mierda)</p> <p>@_Lassana_ Me la has puesto como un tanque apuntando a un campanario negrata  @odiomistweets Es una analogía pendejo sudaca</p> <p>@ElQuemaMarias te parece que un sudaca tatuado excelentemente bien dotado y carismático no es buen ejemplo?</p> <p>@IIQGatica @AnglicaVillarN1 Este sueco con cara de bolita .xq eso es este aspiracional desclasado . x dignidad debería irse . no lo hará xq este sudaca no la tiene !!!</p> <p>@SleeplessXenn @envdiaN Este finde te recojo y vamos al barrio negrata de envdian  @elmundoes El q quería quitar concertinas y luego quitarse a los inmigrantes !!</p> <p>@AlexTor1975 @sanchezcastejon Disfráccese de moro o negro, y al momento tiene resuelto el problema de espera, de verdad, mano de santol. Nada como ser invasor, refugiado, o inmigrante en este paraíso terrenal. Ahora, si eres nativo del paraíso!!!!, Se te puede complicar la existencia!!!!.</p>
HS	<p>CALLATE YA Y HAZME LA PUTA CENA JODERRR URL</p> <p>Basta ya de buenismo, si te no respetas a la polica ni a la guardia civil en la frontera nada de concertinas, simplemente 4 tiros. Fin del problema</p> <p>@benemeritosGC La solución es fácil, muro de hormigón de 3 metros de alto, garitas cerradas para vigilancia y por delante la valla actual con concertinas. No es caro, es cuestión de quererlo hacer.</p> <p>Por desgracia, no queda otra, aportan poco y nos cuestan mucho. Incluido nuestra seguridad. #Inmigración #Immigration URL</p>

Tab. 4.8: Algunos ejemplos observados en el análisis de error que carecen de contexto conversacional. Etiqueta es la asignada en el dataset

Algo que merece cierta atención es que, lejos de empeorar el desempeño de nuestros modelos, agregar nuevas variables a predecir (además de la existencia de discurso de odio) pareciera mejorar levemente la performance de la detección de este fenómeno, a la vez que obteniendo salidas más ricas e interpretables. Más aún, observamos que modelos como el del equipo MITRE [49] en dicha competencia mejoraron la performance con una capa adaptadora que modela las dos variables latentes que el dataset no brinda: la misoginia y el racismo. Teniendo esto en cuenta, una pregunta a explorar es si contar con esta información (las características agredidas) puede mejorar la performance de los clasificadores o tener salidas más interpretables que sólo una etiqueta binaria.

Hacemos a continuación una disquisición no sólo sobre este trabajo y el dataset en el que se basa sino en líneas generales sobre los recursos y enfoques actuales en el área de detección de discurso de odio. Continuando con la idea del párrafo anterior, una limitación que puede verse es que la mayoría de los trabajos atacan una, dos, o a como mucho tres características protegidas. Por ejemplo, los datasets de [147] y [7] (el utilizado en esta tarea) sólo consideran racismo y sexismo, mientras que Davidson et al. [32] agrega homofobia a esta consideración. Sería deseable poder contar con un dataset que como mínimo cuente con estas tres características en conjunto a otras quizás menos utilizadas: odio de clase (a veces conocida como “aporofobia”), discriminación por aspecto físico, o por discapacidad. Esto, desde ya, con un framework unificado de anotación y no recolectando datasets anotados individualmente.

Un problema particular que se puede observar en este dataset (pero que atraviesa a muchos otros) es el proceso de recolección de los datos: los tweets son recolectados mayormente a través de keywords. Como está explicado en el overview de esta shared-task [7], se usó para su recolección una combinación de estrategias. Sin embargo (ver apéndice A), hay una altísima incidencia de algunas palabras (como *sudaca* o *inmigrante*) que sesgan fuertemente el dataset. Esto (entre otras cuestiones) puede ser un problema para los modelos que se entrena sobre estos datos, haciendo que aprendan correlaciones espurias generadas por estas distribuciones. De todas formas, esto es una limitación general para estos tipos de aprendizaje sobre datos crudos y etiquetas, donde es difícil establecer e interpretar cómo un clasificador termina encontrando patrones para detectar el fenómeno medido.

La anotación, la etapa subsiguiente a la recolección de datos, pareciera presentar en este dataset algunos problemas. Hemos visto en la anterior sección una lista no extensiva de numerosos errores de etiquetado, aún cuando este dataset fue realizado con un etiquetado de 2 + desempate. Si bien es difícil trazar las razones detrás de estos problemas, observando las instancias podría uno imaginarse que esto es producto de un no entendimiento de las expresiones en los distintos dialectos del español. Waseem [146] mostró que las anotaciones “amateurs” (producto del uso de crowdsourcing) tienden a tener mayores instancias de Hate Speech (algo que daría la impresión de ocurrir aquí) y que datasets anotados por expertos mejoran la performance de los modelos. Este problema podría profundizarse dado que no queda claro si los anotadores son hablantes nativos de español.

Un problema del dataset estudiado en este capítulo (pero que aplica a otros también) es la falta de contexto: los mensajes carecen de información adicional

sobre la noticia o el tema del que se está hablando. Cuando leemos un mensaje de un tweet, casi siempre lo leemos en el contexto de una noticia, o un trending topic. Muy rara vez leemos un mensaje en total aislamiento. De hecho, gran parte de los comentarios de este dataset tiene un contexto implícito: la noticia de conflicto migratorio en Ceuta. Otros comentarios, por otro lado, no se entienden bien ya que son respuestas a un tweet y que según el hilo de conversación pueden entenderse o no como discriminatorios.

Sobre esta falta de contexto, hay muchos mensajes aislados que pueden requerir información adicional para entender su significado. Por ejemplo, un comentario que dice "hay que matarlos" puede o no entenderse como discurso de odio. Si el objeto del mensaje se refiere a mosquitos, ese mensaje no es odioso; si, por otro lado, está hablando sobre chinos en el contexto del COVID-19, entonces ese mensaje es discriminatorio (y además llama a tomar una medida violenta). En ese sentido, podemos preguntarnos sobre este punto si el acceso a información contextual nos puede auxiliar en la detección de discurso de odio: siendo este un hilo de conversación, una noticia a la que se refiere, o alguna otra forma de información.

Otro problema que suele ocurrir relacionado al anterior es que no tenemos información granular de los datos anotados. Si bien algunos trabajos agregan información de la característica vulnerada, la mayoría simplemente agrega una etiqueta binaria sobre la existencia o no de discurso de odio (o bien algún nivel intermedio como si hay o no discurso ofensivo, como el caso de Davidson et al. [32]). Teniendo en cuenta lo observado en este capítulo, agregar información más detallada sobre cada caso puede ayudar a mejorar la detección del discurso de odio mediante una señal más rica a nuestros clasificadores sobre las diferentes fronteras de cada característica.

#### 4.8. Conclusiones

En este capítulo hemos hecho un primer acercamiento a la tarea de detección de lenguaje discriminatorio, haciendo un repaso de su definición desde un marco legal y desde el usado en la literatura de Procesamiento de Lenguaje Natural. Analizamos técnicas de clasificación del estado del arte sobre el dataset presentado en la shared task multilingual *hatEval*[7]. En base a este dataset, analizamos dos tareas: detección binaria de discurso de odio, y detección de múltiples variables (si es discurso de odio, si es dirigido, si es agresivo).

Para estas tareas, presentamos varios modelos de clasificación. Por un lado, clasificadores lineales que consumen distintos tipos de entrada como ser tweet embeddings y bolsas de caracteres; modelos recurrentes que consumen embeddings contextualizados; y finalmente, modelos del estado del arte basados en modelos de lenguaje pre-entrenados usando la arquitectura de transformers. Para ambas, los modelos de transformers

En el caso de la tarea de detección múltiple, propusimos dos formas de atacar el problema: como clasificación múltiple (predic平do simultáneamente las 3 variables), y convirtiendo a un problema de clasificación simple sobre 5 clases posibles. Observamos, a su vez, que lejos de dañar la performance de la detección de discurso de odio, predecir más de una variable mejora la performance de nuestros clasificadores.

Analizando este dataset y algunos otros de la bibliografía, marcamos algunas

puntos de mejora en la detección de discurso de odio: principalmente, la falta de información contextual. La mayoría de los datasets no tienen mayor información sobre los mensajes de los usuarios, algo que usualmente no pasa en las redes sociales. Por otro lado, y teniendo en cuenta la observación hecha en el párrafo anterior, nos preguntamos si tener información más granular acerca de las características. Otro punto no menor es que para la creación de datasets de un fenómeno tan complejo y social es indispensable tener muchos recaudos a la hora de la anotación, algo que ya ha sido observado en otros trabajos.

En los siguientes capítulos, exploraremos algunas de estas observaciones. Particularmente, nos centraremos en la incorporación de contexto en la detección de discurso discriminatorio, construyendo un dataset que incorpore esta información a los mensajes anotados, y explorando cómo mejorar los algoritmos del estado del arte que aprovechen esa información.



## 5. CONSTRUCCIÓN DE UN DATASET DE DISCURSO DE ODIO CONTEXTUALIZADO

Por lo marcado en anteriores secciones, consideramos interesante el problema de analizar el impacto del contexto en la detección de lenguaje discriminatorio. Antes de proseguir, podemos preguntarnos: ¿a qué nos referimos con el término “contexto”? La contextualización, según John Cook-Gumperz<sup>X</sup> es:

(el) uso que hacen hablantes y oyentes de señales verbales y no verbales para poder conectar lo que se dice en un momento con el conocimiento adquirido a través de la experiencia para poder mantener la participación en la conversación y entender lo que se pretende decir. (Gumperz [56])

En este sentido, cualquier señal que pueda ayudar a entender las intenciones del interlocutor en una red social es información que ayuda a situar los mensajes: desde el hilo de una conversación, la noticia a la que hace referencia, el historial de conversaciones previas entre los interactores, información sociocultural de los interlocutores, entre otras [135]. Para poner un ejemplo de por qué es necesario disponer de información adicional al comentario analizado, el mensaje “sos un hombre” en solitario puede parecer inofensivo; ahora, si ese mismo mensaje está dirigido hacia una mujer trans, su contenido es claramente discriminatorio. El comentario con claro tono agresivo “hay que tirar una bomba ahí” puede tener carácter discriminatorio si lo consideramos en el contexto de una nota que habla sobre China y el COVID-19; sin embargo, es distinto si estamos hablando de un partido de fútbol, donde un remitente de un club manifiesta su enemistad contra otro equipo.

Vimos en el anterior capítulo que muchos mensajes del dataset analizado no se entendían bien al carecer de información contextual, tanto conversacional o del tópico al que hace referencia. En líneas generales, la mayoría de los problemas de NLP sobre textos sociales suelen plantearse sobre comentarios sin ningún otro tipo de dato de quien lo emite, a quién se lo dirige, ni sobre qué tema está hablando. Para analizar esto desde el problema de la detección de discurso de odio, nos abocamos en primer lugar a la tarea de crear un dataset que no sólo contenga un mensaje/comentario, sino que provoca un contexto para éste. Un ámbito natural para esta tarea son las notas periodísticas, donde disponemos de una nota y comentarios realizados sobre ésta. En este caso, el comentario es el texto a analizar, mientras que el contexto está dado por la nota.

Muchos sitios de noticias disponen de sistemas embebidos de comentarios, pero vista la dificultad para la recolección a la vez que los limitados datos provistos por estos sitios nos llevaron a buscar otro medio: Twitter. Esta red social provee una sencilla API para descargar datos, a la vez que tiene términos y condiciones amigables para poder publicar estos datos. Así mismo, Twitter opera de una manera similar a un foro de comentarios de un sitio de noticias. Este tipo de datos (comentarios sobre artículos periodísticos) tiene una naturaleza particular, ya que las agresiones

<sup>X</sup> Lingüista que estudió el problema de la contextualización

(No apabo nota!)

discriminatorias son usualmente a personajes públicos o colectivos de personas, y se dan de manera indirecta (a través del comentario en la noticia) y no directa (es decir, como respuesta al usuario de Twitter ofendido).

El trabajo realizado en este capítulo tuvo lugar en el contexto de un Proyecto Interdisciplinario de la UBA<sup>2</sup> junto a sociólogos, abogados, lingüistas, y computólogos. Particularmente, el trabajo de la construcción del manual de etiquetado fue discutido en conjunto, contemplando varias perspectivas a la hora de armar una definición propia (algunas de estas ya fueron vertidas en la discusión en la sección 4.1). Teniendo en cuenta que un alto porcentaje de trabajos del área de detección de discurso de odio (y de manera más importante, en la construcción de sus recursos) mediante técnicas de NLP no abordan una mirada interdisciplinaria, es un aspecto a remarcar de lo realizado en la construcción de este recurso.

### 5.1. Trabajos previos

Pocos trabajos del área de detección de lenguaje abusivo o discurso de odio incorporan algún tipo de contexto a los comentarios del usuario para estas tareas. En esta sección haremos un repaso de los trabajos que han abordado esto de alguna manera. Gao and Huang [46] construyó un dataset de lenguaje discriminatorio sobre 1518 comentarios del sitio de Fox News. A los anotadores les fue presentado tanto el comentario como la noticia a la hora de realizar el etiquetado. Sobre este dataset, los autores efectuaron experimentos de clasificación usando modelos lineales (regresiones logísticas) y modelos neuronales. En estos experimentos, observaron que un clasificador (tanto lineal como neuronal) mejora su performance al consumir el título de la noticia, dando indicios de que se puede aprovechar el contexto para mejorar la detección de este fenómeno. Sin embargo, como marca Pavlopoulos et al. [109] este trabajo cuenta con algunos problemas: en primer lugar, el tamaño del dataset es pequeño, y está extraído de sólo 10 noticias, lo cual limita fuertemente los posibles contextos. A su vez, la anotación fue realizada mayormente por una única persona, lo cual hace poco confiables las etiquetas obtenidas. Luego, algunos detalles menores debieran ser analizados con mayor detalle, como por ejemplo la utilización de los nombres de usuarios como features predictivas.

Mubarak et al. [100] construyó un dataset en árabe sobre comentarios con contenido abusivo del portal Al Jazeera. Sin embargo, este dataset tiene un problema: los comentarios son presentados a los anotadores sobre noticias, ignorando todo el thread de la conversación. Esto hace que el contexto sea presentado de manera parcial.

Paralelamente a nuestro trabajo, Pavlopoulos et al. [109] analizó el impacto de agregar contexto a la tarea de detección de toxicidad. En particular, plantea dos preguntas:

- ¿Qué tanto afecta el contexto a la toxicidad percibida por humanos en conversaciones online?

<sup>2</sup> <https://cyt.rec.uba.ar/vinculacion-transferencia/piuba/>

- ¿Puede el contexto ayudar a mejorar la performance de clasificadores de toxicidad en comentarios?

Para responder estas preguntas, los autores construyeron dos datasets en base a Wikipedia Talk Pages [68], un dataset de discusiones del sitio de Wikipedia. En primer lugar, armaron un dataset de 250 comentarios anotados por dos grupos disjuntos de anotadores: uno de los grupos anotó los comentarios de manera contextualizada, viendo tanto el comentario en cuestión como el título de la discusión; el otro grupo sólo vio el comentario a anotar sin contexto alguno. En dicho experimento observaron que los anotadores contextualizados percibieron 6.4 % de comentarios tóxicos versus un 4.4 % de quienes anotaron sin contexto, una diferencia significativa aplicando un test Mann-Whitney. Desagregando estos resultados, observaron que 13 de los 250 comentarios (5.2 %) tuvieron diferencias de anotación entre los dos grupos, con 9 (3.6 %) comentarios donde aumentó la toxicidad percibida y 4 comentarios donde bajó la toxicidad al ser agregado el contexto.

Para responder la segunda pregunta, anotaron un dataset de 20k comentarios, 10k anotados por un grupo que etiquetó viendo el contexto y otros 10k que no lo vio. Entre todos los comentarios del dataset original de Wikipedia Talk Pages, eligieron aquellos con profundidad entre 2 (respuestas directas) a 5, y con entre 10 y 400 caracteres de largo. Luego, entrenaron varios clasificadores usando este dataset y allí pudieron observar que el contexto no parecía mejorar la performance. En el próximo capítulo nos extenderemos sobre las técnicas utilizadas por este trabajo.

Xenos et al. [150] continúa el trabajo de este trabajo desagregando el resultado de la segunda pregunta. Puntualmente, y observando que sólo un porcentaje pequeño de los comentarios parecen ser incididos por el contexto en el trabajo previo, construyen una nueva tarea: estimación de sensibilidad al contexto. Para ello, toman el dataset de Civil Comments [20], y reanotan un subconjunto de este dataset usando información de contexto a través de crowdsourcing. Las etiquetas de este dataset son de toxicidad en un estilo similar a una regresión ordinal, entendiendo las categorías no tóxico, incierto, tóxico, y muy tóxico. Ahora, teniendo las anotaciones originales del dataset (que fueron hechas sin contexto) y las nuevas anotaciones, pueden definir para cada comentario una sensibilidad al contexto, dada por

$$\delta(p) = s^{oc}(p) - s^{ic}(p) \quad (5.1)$$

donde  $s^{oc}$  es la fracción de anotadores sin contexto que marcaron toxicidad, y  $s^{ic}$  los que no tienen contexto.

Sheth et al. [135], en un trabajo muy reciente, señala algunas oportunidades y desafíos para incorporar fuentes de información más ricas a la tarea de detección de toxicidad. Por ejemplo, incorporar información como el background socio-cultural de los interactores puede ayudar a distinguir algunos tipos de reapropiación de términos potencialmente catalogados como tóxicos. Así mismo, el historial de interacción entre los usuarios puede ayudar a distinguir interacciones abusivas de charlas amistosas entre amigos que usan vocabulario potencialmente tóxico. Finalmente, se promueve el uso de contenido externo para acercarse lo más posible al conocimiento humano a través de conocimiento del contenido, el individuo (atacado) y la comunidad. Para



Fig. 5.1: Muestra de la recolección de datos

ello, se promueve el uso de bases de conocimiento y knowledge-infusion learning [48] para combinar el cómputo neuronal y simbólico.

Wiegand et al. [148] menciona formas implícitas de abuso, mucho más complejas que las basadas solamente en palabras ofensivas. Por ejemplo, deshumanizaciones (“los judíos son una plaga que merece ser eliminada”), llamadas a la acción (“hay que tirar una bomba en ese país”), acusaciones (“los chinos inventaron el coronavirus”), entre otros tipos sutiles de comportamiento tóxico. Así mismo, menciona que la mayoría de los datasets no consiguen capturar estos fenómenos debido a la forma de recolección usualmente basada en keywords.

## 5.2. Esquema del dataset

Para construir un dataset contextualizado barajamos varias opciones. Como vimos en otros datasets, se puede entender el contexto de varias maneras: un contexto “temático”, donde sabemos que cierto comentario habla sobre un tema en particular; y un contexto conversacional, donde tenemos una secuencia de comentarios (un hilo o thread) y podemos extraer un comentario padre para cada uno salvo el raíz. La primera opción es la explorada por Gao and Huang [46], Mubarak et al. [100], donde construyen un dataset de comentarios de Fox News y Al-Jazeera respectivamente. El contexto conversacional, como hemos relatado anteriormente, es explorado en Pavlopoulos et al. [109], Xenos et al. [150]; sin embargo, como es marcado en el primer trabajo, la recolección de datos es no trivial, aún en un caso más amplio como el lenguaje abusivo, ya que la incidencia es relativamente baja. Puede esperarse que en el contexto de lenguaje discriminatorio se dificulte aún más esto.

Para analizar el contexto, decidimos entonces usar la primera opción: comentar-

rios sobre notas periodísticas. No vamos a considerar un hilo de respuestas, sino simplemente aquellos comentarios que sean directos sobre la nota. En ese punto, el dataset que queremos construir sería similar al de [46]. Una diferencia respecto a este dataset es la de incorporar dos modos de contexto: uno corto, donde sólo tengamos el título de la noticia; y uno largo, donde tengamos el texto completo de la noticia.

Algo no menor a la hora de considerar la construcción del dataset es la posibilidad de publicar los datos. Por citar un ejemplo, el dataset de Gao and Huang [46], si bien tiene sus datos de acceso público<sup>3</sup>, no queda claro que los términos y condiciones de la fuente de donde se trajeron permita esto. Más aún, si hubiésemos querido extraerlo de múltiples fuentes (por ejemplo, varios diarios), deberíamos chequear y/o acceder a permisos para cada sitio, a la vez que tendríamos el problema de tener fuentes diversas de los datos (diferentes longitudes, metadatos distintos, entre otras).

Para evitar muchos de estos problemas, y reutilizar muchas cuestiones con las que venimos trabajando en esta tesis, decidimos trabajar sobre comentarios hechos por usuarios en Twitter. Concretamente, sobre respuestas de comentarios de usuarios a posteos hechos por cuentas de medios. De alguna manera, esto emula~~a~~ un foro de comentarios de medios, tenemos un formato único para comentarios mientras tenemos diferentes "audiencias". La Figura 5.1 ilustra esta idea. A su vez, los términos y condiciones de Twitter nos permiten publicar los datos. Las notas periodísticas las descargaremos pero debido a~~problems~~ de copyright no serán publicados.

Finalmente, la elección del idioma. El dataset construido será sobre comentarios realizados en idioma español, más precisamente en la variedad dialectal del Río de la Plata. Considerando que el discurso de odio es un fenómeno cultural, es importante que quienes creamos este recurso seamos conscientes del trasfondo donde éste ocurre. Por otro lado, siempre es importante generar recursos en idiomas menos favorecidos.

### 5.3. Proceso de construcción

Dividiremos la construcción del dataset en tres etapas:

1. Recolección: Proceso de recolección de datos de Twitter y de los artículos periodísticos
2. Selección: Dado el conjunto de artículos y comentarios recolectados, tomar una muestra de artículos y comentarios a etiquetar
3. Anotación: Proceso de etiquetado de los artículos seleccionados

Si bien en muchos casos las dos primeras etapas suelen ser la misma o bien la selección se limita a una muestra aleatoria de la recolección, este procedimiento sería muy incívico en el caso de discurso de odio. Esto se debe a que en nuestro dominio de comentarios periodísticos y discurso de odio, encontramos este tipo de discurso distribuido de manera muy poco uniforme, usualmente concentrada alrededor de ciertos tópicos. Para construir un dataset con una proporción no marginal del

<sup>3</sup> <https://github.com/sjtuprog/fox-news-comments>

Nombre	username	#Followers
La Nación	@LANACION	3.6M
Clarín	@clarincom	3.2M
Infobae	@infobae	3.0M
Perfil	@perfilcom	0.81M
Crónica	@cronica	0.80M

Tab. 5.1: Cuentas de medios utilizadas para la recolección de datos.

fenómeno estudiado, estudiamos algunas posibilidades para seleccionar los artículos y sus respectivos comentarios.

En algunos trabajos previos (como por ejemplo Basile et al. [7], Waseem and Hovy [147]) la recolección y selección constan conjuntamente de usar ciertos keywords y, o bien recolectar tweets que usen esas palabras, o bien sirven para preseleccionar usuarios de los cuales luego extraer tweets para ser etiquetados. En nuestro caso, la selección de artículos y comentarios presenta cierta novedad y complejidad, con lo cual separamos este procedimiento para explicarlo detalladamente en las siguientes secciones.

#### 5.4. Recolección de datos

En esta sección describiremos el proceso de recolección de datos. La salida de esta etapa será un conjunto de artículos y sus comentarios extraídos de Twitter. Describiremos a continuación las decisiones realizadas respecto a las fuentes y a decisiones técnicas realizadas.

Limitamos nuestra recolección de datos a cuentas de medios de la República Argentina y, puntualmente, nos centramos en diarios con comunidad mayormente rioplatense ya que (como comentaremos más adelante) los anotadores son nativos de esa variedad dialectal. Esto es teniendo en cuenta que esta tarea depende fuertemente de la jerga y de las variaciones dialectales de cada país, decidimos realizar sólo anotación de estos diarios. Esto además se debe a que, habiendo buscando en otros medios de Argentina (como por ejemplo “La voz del Interior”, diario dirigido mayormente a un público fuera de la Metrópolis de Buenos Aires) observamos que la interacción en Twitter de estos medios es muy baja: pocos usuarios comentan sus notas. Centrándonos en diarios que generen interacción, los cinco diarios sobre los que anotamos datos están listados en la tabla 5.1.

Si bien recolectamos notas de otros medios, no los consideraremos a partir de ahora, y los dejamos para análisis posteriores. Los medios elegidos son medios formales y tradicionales, 4 de ellos con soporte escrito, siendo Infobae el único medio enteramente digital. Consideramos la posibilidad de elegir medios no tradicionales y más orientados a grupos de la “derecha alternativa”. Estos medios son altamente generadores de contenido de odio. Sin embargo, finalmente tomamos la decisión de descartarlos de la etapa de anotación.

Creo que los comillas deberían abarcar también a ‘derecha’, porque es cualquier cosa realmente...

### 5.4.1. Método de recolección

La API de Twitter, en su versión gratuita, nos brinda dos modos de recolectar tweets de su plataforma<sup>4</sup>:

1. Search API: permite buscar tweets en base a términos, de hasta 15 días atrás sobre una pequeña muestra, recreando lo que vemos en la UI de Twitter
2. Stream API: permite buscar tweets en tiempo real sobre una muestra de cerca del 1% de todos los tweets de la red social

La API Stream (también conocida como Spritzer), mientras por un lado limita temporalmente la recolección de datos, por el otro nos brinda la posibilidad de recolectar una mayor cantidad de información en tiempo real. Más aún, dada la naturaleza de nuestros datos (discurso de odio), se corre el riesgo de que con el tiempo sean moderados e inaccesibles para cualquier búsqueda con la API Search.

Por lo explicado, usamos la API de Twitter Stream mencionando cualquiera de estas cuentas. Si estamos entonces recolectando tweets sobre @medio, el proceso de recolección nos da:

1. Tweets de @medio
2. Respuestas a los tweets de @medio
3. Tweets de terceros que mencionan a @medio
4. Retweets (RT) de tweets de @medio
5. Citas de tweets de @medio

de las cuentas de los demás?

simplemente

Los RTs y tweets que arrojen a @medio carecen de interés para nuestro estudio, con lo cual los descartamos. Por otro lado, también descartamos las citas, aunque podrían entenderse como "respuestas" a los tweets originales. Nos quedamos con tweets de @medio y las respuestas a estos. Si bien la API nos da estos tweets desestructuradamente, reconstruimos el árbol de la discusión mediante el campo `in_reply_to_status_id`<sup>5</sup>.

Algo importante a remarcar es que para el propósito de este trabajo, solo estamos interesados en el primer nivel de respuestas al tweet original, y no incorporaremos hilos de respuestas. Trabajo futuro debería explorar este nivel adicional de complejidad incorporando contexto conversacional adicional.

Accidentalmente, la recolección de datos se dio al mismo tiempo del estallido de la pandemia del COVID-19. Por ese motivo, y dadas las implicancias de la pandemia sobre el discurso discriminatorio en las redes sociales, se volcó el foco hacia artículos relacionados con el coronavirus. Para ello seleccionamos artículos buscando una cantidad de palabras en su cuerpo, por lo que seleccionamos específicamente

<sup>4</sup> Usamos la versión 1.1 de la API. La versión 2.0 parece facilitar la recopilación de conversaciones. Recomendamos investigar mejor esta versión actualizada para esquivar muchas de las dificultades técnicas que incursionamos para lo descripto en esta sección

<sup>5</sup> Ver la documentación y la referencia al campo en <https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/tweet>

una "cantidad"?

buscando la ocurrencia de al menos una palabra de un conjunto que seleccionamos específicamente relacionados al COVID-19: coronavirus, encierro, ...

Medio	# Artículos	# Comentarios
@infobae	45,652	822,462
@clarincom	29,050	672,101
@perfilcom	8,761	61,203
@LANACION	16,040	506,091
@cronica	17,250	70,872
Total	116,756	2,133,029

Tab. 5.2: Artículos recolectados por medio

artículos relacionados con COVID-19. Utilizamos las siguientes palabras: coronavirus, encierro, síntomas, covid, fase, fiebre, cuarentena, infectados, distanciamiento, normalidad, Wuhan, aislamiento.

Por último, nos quedamos con aquellos tweets de los medios periodísticos que tuvieran un link a un artículo. Para ello, utilizamos la librería *newspaper3k*<sup>6</sup>, que nos permite acceder a la información relacionada a los artículos en cuestión, en particular siendo lo que más nos interesa el cuerpo del artículo. Esto vamos a utilizarlo posteriormente como el contexto “largo” para los comentarios. Aquellos tweets de medios periodísticos que no contengan un link a un artículo fueron descartados de las siguientes etapas.

#### 5.4.2. Datos recolectados

La tabla 5.2 contiene los números de los artículos recolectados por cada medio, luego de aplicado el filtro de palabras mencionado en la anterior sección. Si bien recolectamos más artículos de otros medios, no son enumerados. Infobae es el medio que más producción de artículos genera, y también será finalmente sobre el que más comentarios etiquetemos.

En el apéndice C.1 mostramos la distribución temporal de los datos. Si bien tenemos un pequeño gap en los datos por un problema técnico en la recolección, tenemos datos desde Marzo de 2020 hasta Febrero de 2021.

En siguientes secciones realizaremos un filtrado de la mayoría de estos artículos previamente a la anotación, pero este conjunto de datos no filtrado será utilizado para efectuar ajustes de dominio, y es liberado como se recomienda en Gururangan et al. [58]. Hablaremos más sobre esto en los capítulos 6 y 7.

#### 5.5. Selección de datos a anotar

Un problema que se nos presenta antes de comenzar el etiquetado es el de seleccionar los artículos que vamos a etiquetar, considerando la gran cantidad de datos recolectados y los recursos disponibles. Una primera posibilidad para hacer esto es realizar una selección aleatoria de artículos y comentarios; sin embargo, los comentarios discriminatorios no se distribuyen de manera uniforme entre los artículos sino que se concentran sobre algunos temas. Es mucho más probable encontrar comentarios de índole discriminatoria en notas que tengan temas cercanos a alguna de

<sup>6</sup> <https://newspaper.readthedocs.io/en/latest/>

China	piqueteros	mamá	domésticas
Cuba	villas	de género	la modelo
cubano	la villa	aborto	la periodista
bolivia	movimientos sociales	actriz	la cantante
paraguayo	organizaciones sociales	actrices	travesti
judío	tomas de tierras	feminista	trans
camionero	toma de tierras	femicidio	gay
ladrón	sindicatos	enfermera	homosexual
represión	Guernica	madre	de la V
criminal	mapuches	Ofelia	

Tab. 5.3: Palabras utilizadas para la selección de artículos. Cada palabra se busca sobre el cuerpo del artículo para seleccionarlo como candidato a ser etiquetado

ver Tab. 5.4.

las características protegidas: por ejemplo, es esperable que encontremos contenido discriminatorio en notas sobre China y el Coronavirus o sobre una chica transgénero antes que en un artículo de fútbol o economía. Si bien una selección aleatoria preservaría una tasa de incidencia mucho más cercana a la observada en el universo de comentarios, es más importante poder obtener una mayor cantidad de observaciones que reflejen el fenómeno estudiado.

Teniendo esto en cuenta, evaluamos varias alternativas para realizar la selección de artículos. La primera fue intentar seleccionar aquellos artículos que consideremos que puedan tener contenido potencialmente discriminatorio. Una posibilidad para esto sería usar algunas palabras “semilla” para seleccionar artículos interesantes.

Otra posibilidad que evaluamos fue la de buscar directamente comentarios que nos marquen que ese artículo suscita contenido discriminatorio. Para ello, podemos listar algunos insultos comunes o expresiones peyorativas hacia los grupos protegidos considerados. Es necesario remarcar que esto lo hacemos para seleccionar **artículos** y no los comentarios que contengan esos insultos; hacer esto último nos genera una muestra muy distorsionada y tendiente a encontrar el fenómeno más explícito de la discriminación (el insulto racista, homofóbico, etc.).

Describimos a continuación las alternativas analizadas para seleccionar los artículos y sus respectivos comentarios.

### 5.5.1. Selección en base a artículos

En primer lugar, consideramos la posibilidad de hacer una selección en base al contenido de los artículos. Luego de realizar algunos experimentos usando LDA [16] para buscar tópicos posibles de las notas, decidimos realizar una selección un poco más controlada y determinística en base a la utilización de palabras clave. Es decir, seleccionar artículos en base a la aparición o no de ciertas keywords.

Para ello, indexamos todos nuestros artículos en MongoDB<sup>7</sup>, una base de datos no relacional y desestructurada. MongoDB permite la utilización de índices en base a texto, y realizar búsquedas en base a textos, palabras, e inflexiones. Cada artículo fue indexado en base al contenido de su cuerpo.

<sup>7</sup> <https://www.mongodb.com/>

bija	urraca	vicio puto	trolo	peruano	matarlos	negra
prostituta	tucán	trabuco	sodomita	peruca	una bomba	negro de
feministas	putita	travesti	chinos de	judío	vayan a laburar	negros
feminazis	reventada	trava	bolita	sionista	vayan a trabajar	bala
aborteras	marica	degenerado	paraguayo	villeros	gorda	uno menos

Tab. 5.4: Palabras utilizadas para recolectar comentarios

Hay que decir algo sobre estos listados  
que construye esto

La tabla 5.3 muestra el conjunto utilizado para recolectar artículos. Como vemos, hay diversas palabras que recogen distintas temáticas de posibles tópicos “calientes”, algunos muy locales respecto a eventos concretos durante la pandemia. Si algún artículo contiene una de las frases mencionadas, se selecciona el artículo para ser etiquetado.

### 5.5.2. Selección en base a comentarios

Otra posibilidad evaluada fue la de observar los comentarios de los artículos en lugar del contenido del artículo, y seleccionarlos en base a esto. En este punto, la idea es únicamente seleccionar los artículos y no los comentarios; estos últimos son sólo usados como “pistas” para encontrar comentarios con posible contenido discriminatorio, y como tal identificar a ese artículo como un posible generador de este tipo de contenido.

La idea es similar a la de la selección con artículos, sólo que aplicada a comentarios: buscamos comentarios que contengan alguna de las palabras semilla listadas en la Tabla 5.4. Estas palabras fueron recolectadas a base de experimentación y observación de los datos, y tratan de contener diversas expresiones de contenido mayormente discriminatorio. El procedimiento de selección consta de –dado un artículo– marcar sus comentarios que contengan una o más de las expresiones listadas. Si el artículo tiene tres o más comentarios marcados, entonces seleccionamos el artículo; caso contrario, es descartado.

Remarcamos nuevamente que este proceso de selección es para los *artículos*, no para los comentarios. Hacer esto para los comentarios implicaría necesariamente que tengan alguna de las expresiones, algo que sesgaría nuestro dataset.

Luego de algunos análisis experimentales y observacionales de las dos posibles metodologías, decidimos utilizar el muestreo de artículos en base a comentarios. En base a un análisis subjetivo, los artículos seleccionados parecían tener mayor incidencia de mensajes discriminatorios y eso nos decantó hacia esa opción.

Una posibilidad que tuvimos en cuenta fue la de usar un clasificador pre-entrenado que nos señale posibles comentarios discriminatorios y luego usar eso para seleccionar artículos candidatos a etiquetar. Para ello, aplicamos un clasificador basado en BETO (Canete et al. [26]) (ver sección 4) sobre los comentarios de los artículos. Una evaluación subjetiva de esto nos dio pobres resultados, tanto porque no captaba algunas agresiones discriminatorias (de características no incluidas en el dataset de Basile et al. [7]) como muchos falsos positivos o errores debido al cambio de dominio (temático y también dialectal). Si bien descartamos este método, puede ser de relevancia usar algún método que no esté basado en palabras sencillas o utilizar algún método semi-automático para encontrar candidatos a etiquetar.

Esto!  
Agregó lo  
mismo para b  
Tabla 5-3  
entonces.

### 5.5.3. Muestreo de comentarios

Una vez que seleccionamos los artículos, resta decidir qué comentarios vamos a anotar. No podemos seleccionar todos ya que muchos artículos cuentan con una cantidad importante de comentarios (en el orden de los cientos) y es deseable mantener un balance entre los comentarios anotados por artículo. Tampoco es deseable (en pos de maximizar el producto de la anotación) seleccionar comentarios de artículos escasamente discutidos.

Teniendo esto en mente, realizamos lo siguiente: primero, nos quedamos sólo con los comentarios de artículos que tengan al menos 20 comentarios. Luego, para cada artículo, seleccionamos aleatoriamente hasta 50 comentarios que no contengan URLs u otro contenido no textual. En este punto, consideramos el muestreo aleatorio como la forma menos sesgada para seleccionar nuestros comentarios. Observemos que en estos pasos no tuvimos en cuenta en ningún momento ?

Mencionamos así mismo algunas de las posibilidades que evaluamos para la selección de comentarios. Una fue la de considerar todo el universo de comentarios y seleccionar la muestra de allí. Sin embargo, esto sobrerepresentaría a aquellos temas muy comentados (muchos de ellos, de temas políticos que se filtran en nuestra selección). Otra consideración posible es la de utilizar información de usuarios y sus conexiones, información que Twitter nos brinda a través de los followers de cada usuario. Muchos usuarios que generan contenido discriminatorio en redes sociales se agrupan en comunidades, y usar algún tipo de información sobre esto (posiblemente, sectorizando en comunidades con algún algoritmo como el de Louvain [17]) podría auxiliar al balance de comentarios posiblemente discriminatorios. Para la tarea de stance detection, Lai et al. [80] y Furman et al. [44] usan este tipo de algoritmos para detectar de una manera semi-automática las posturas de los usuarios respecto a distintos temas.

## 5.6. Anotación

Hasta este momento describimos la recolección de los datos, al cual le siguió la selección de los artículos y comentarios a anotar. Pasamos ahora a detallar el proceso de anotación, describiendo el modelo utilizado, dando información sobre los etiquetadores, y las herramientas utilizadas para este fin. Comenzamos detallando qué es lo que queremos anotar.

### 5.6.1. Modelo de etiquetado

Un modelo de anotación es una representación práctica del objetivo de anotación [121]. En base a la discusión del capítulo anterior, queremos marcar comentarios discriminatorios y ~~marcas~~, granularmente, a qué grupos y/o características se está ofendiendo. También, queremos identificar llamados a tomar alguna acción contra los objetos de esos discursos.

En Zampieri et al. [154] se introdujo un modelo jerárquico de anotación para la tarea de lenguaje ofensivo, utilizado en las competiciones OffensEval [155] y hatEval [7]. La idea de la anotación jerárquica es realizar anotaciones adicionales sólo para

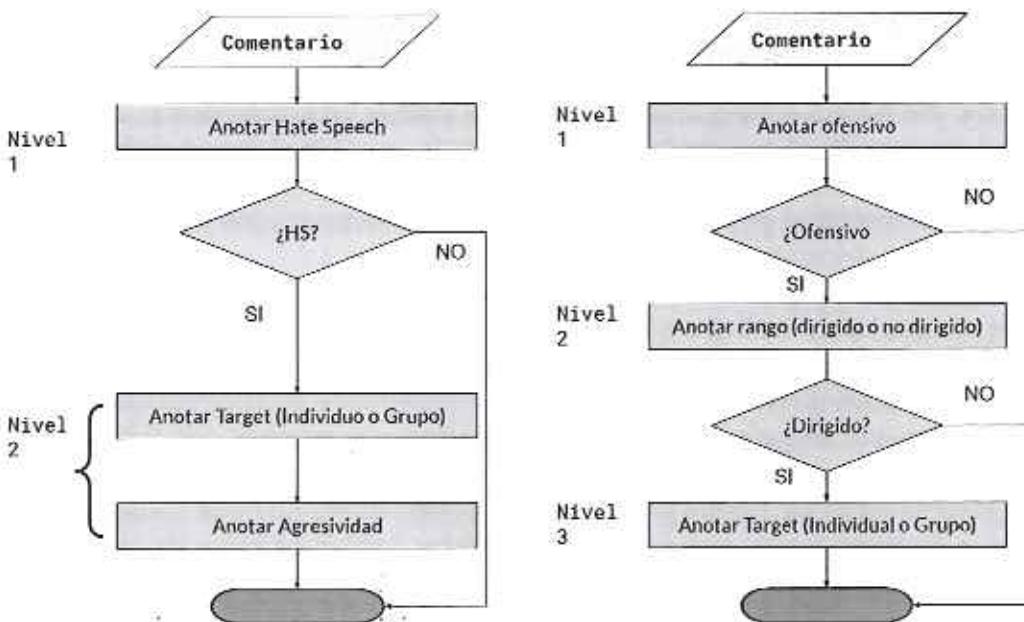


Fig. 5.2: Modelos jerárquicos de anotación. A la izquierda, tenemos el modelo jerárquico propuesto para *HatEval* [7], a la derecha el modelo propuesto para *OffensEval* [155]

algunos casos de anotaciones del nivel anterior. En el caso de *HatEval*, tenemos un primer nivel que consta de anotar si un tweet contiene o no discurso de odio (nivel 1). Si el tweet tiene discurso de odio, entonces anotamos si está dirigido a un individuo o a un grupo, y también anotamos si es agresivo o no (ambos nivel 2). En el caso de *OffensEval*, primero anotamos si es ofensivo (nivel 1), luego si está dirigido o es un insulto no dirigido (nivel 2) y finalmente, si es dirigido y ofensivo, marcamos su objetivo (nivel 3). En la figura 5.2 ilustramos ambos modelos.

Basándonos en esto planteamos nuestro modelo, ilustrado en la figura 5.3. Para cada comentario y su respectivo contexto (el artículo), requerimos una anotación para decidir si el comentario es odioso o no. Si no es odioso, no se necesita más información. Si es así, el par artículo-comentario debe contener, además, una anotación por si llama o no a la acción, y al menos una categoría protegida marcada como ofendida.

### 5.6.2. Definición de discurso de odio y manual de etiquetado

Teniendo en cuenta la discusión realizada en la sección 4.1 realizamos nuestra propia definición de discurso de odio. Entendemos que hay discurso de odio en un texto social si éste contiene declaraciones de carácter intenso e irracional de rechazo, enemistad y aborrecimiento contra un individuo o contra un grupo, siendo estos objetivos de estas expresiones por poseer (o aparentar poseer) una característica protegida. Esta expresión puede manifestarse de manera explícita como insultos directos, celebraciones de crímenes, incitaciones a tomar medidas contra el individuo o grupo, o también expresiones más veladas. Siempre, considerando, que no es nece-

Este libro  
de pensar

revisar juntas

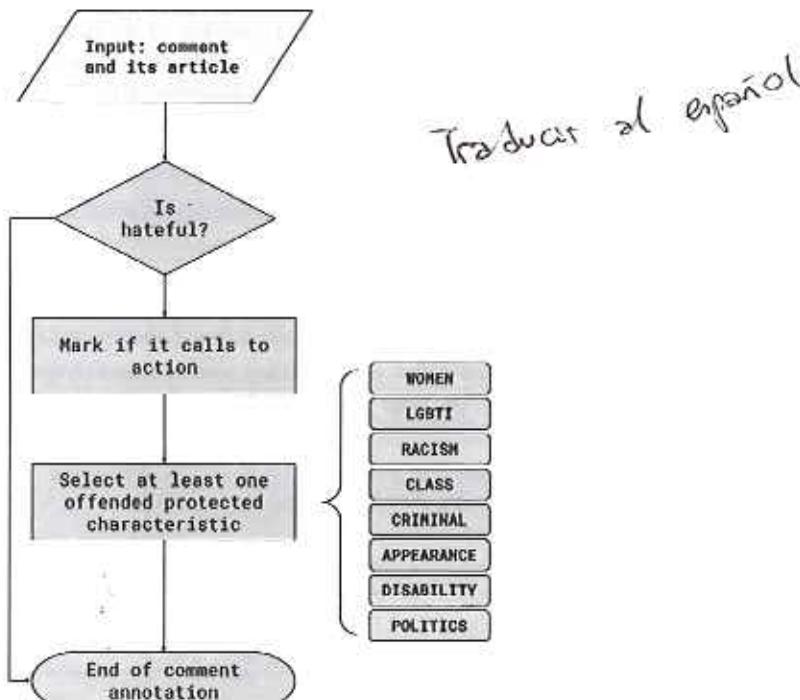


Fig. 5.3: Modelo de anotación

sario solamente un insulto o una agresión: es necesario hacer una apelación explícita o implícita a al menos una característica protegida.

A diferencia de otros trabajos, nuestra definición comprende varias características, incluso algunas que están en la frontera de ser "protegidas". Mientras en otros trabajos se centran mayormente en racismo y misoginia, aquí agregaremos homofobia y transfobia, odio de clase (a veces conocido como aporofobia), por su aspecto físico, y otras. En particular, hay dos características no convencionales que tuvimos en cuenta. En primer lugar, el discurso de odio "político", que de acuerdo a CIDII [29] (ver sección 4.1) es difícil considerar como protegida ya que puede dar lugar a censura y restricciones de la libertad de expresión.

Por otro lado, consideramos el discurso de odio contra criminales, presos, y otras personas en situación de conflicto con la ley. Si bien este punto ni siquiera es considerado como una característica protegida en ninguno de los trabajos mencionados en la sección 4.1, agregamos esta característica debido a la enorme cantidad de contenido alentando la violencia contra criminales en las noticias de policiales, teniendo en cuenta que nos interesa detectar la incitación a la violencia.

Tenemos entonces 8 características que agrupan tipos de discurso de odio: contra las mujeres; racismo y xenofobia; contra la comunidad LGBTI; odio de clase; gordofobia, gerontofobia y demás odio por aspecto; por su ideología política; y finalmente contra discapacitados y adictos. Las características en cuestión son listadas en la tabla 5.5 junto a nombres de referencia que usaremos en éste y el próximo capítulo.

Con esta definición confeccionamos un manual de etiquetado de referencia para

Nombre	Descripción
MUJER	Misoginia, agresiones basadas en ser mujer
LGBTI	Homonofobia, transfobia, y ofensas a la comunidad LGBTI
RACISMO	Racismo, Xenofobia, Judeofobia, etc.
POBREZA	Basado en su condición de clase
POLÍTICA	En base a la filiación política del agredido
ASPECTO	Gordofobia, gerontofobia
CRIMINAL	Criminales, presos, y personas en conflicto con la ley
DISCAPACIDAD	Discapacidades y problemas de adicciones

Tab. 5.5: Características protegidas consideradas en este trabajo. Consideramos una agrupación de ciertas características bajo una misma denominación: por ejemplo, LGBTI contempla homofobia, transfobia, entre otras; análogamente racismo puede contemplar xenofobia, y otras variantes de este fenómeno.

Género	Edad	Estudios	Área	Identificación	¿Activista?	Experiencia
F	27	Doctorado*	Psicología	Mujer	No	Sí
NB	33	Grado*	Artes	LGBTTIQ	No	No
F	30	Grado*	Antropología	Mujer, LGBTTIQ	Feminista	Sí
M	38	Grado	Sociología	No	No	No
F	36	Doctorado	Psicología	Mujer	No	No
F	34	Grado	Comunicación	No	Migrantes	No

Tab. 5.6: Información sobre los anotadores. En el caso de estudios, \* indica en curso. Identificación se refiere a si se autopercebe como perteneciente de una característica protegida considerada en este trabajo. Experiencia se refiere a haber etiquetado previamente otros datasets.

*'participado de la anotación de otros datasets'*

*¡No basta o no es necesario?*

los anotadores. Esta definición y el manual de etiquetado fueron desarrollados iterativamente, primero realizando algunas pruebas de etiquetado entre miembros del equipo y rondas de discusión posterior analizando los ejemplos problemáticos y casos borde. De estas iteraciones logramos ir mejorando el manual, agregándole ejemplos y definiciones, hasta llegar a una versión definitiva. Para cada característica agregamos algunas consideraciones adicionales: por ejemplo, para MUJER no basta con que se insulte a la mujer sino que se apele a algo distintivo de la mujer ("algo que no le diría a un hombre") y para la característica LGBTI mencionamos particularmente las expresiones de asco, incluyendo en esto a los emojis. En el apéndice C.3 puede encontrarse el manual de etiquetado completo entregado a los etiquetadores.

### 5.6.3. Etiquetadores *están*

A diferencia de otros trabajos (como hatEval [7]) decidimos garantizar que nuestros anotadores están más cerca culturalmente al problema en cuestión y también tener mayor control del perfil de estos. El discurso de odio tiene un fuerte componente cultural, muchas veces expresado a través de jerga o expresiones dialectales muy particulares, y relacionado con noticias muy propias de esta región.

Reclutamos etiquetadores hablantes nativos, estudiantes o graduados/as de carreras de ciencias sociales, humanidades o afines – como ser Psicología, Sociología, Comunicación, Antropología. Algo que particularmente nos interesó fue que no

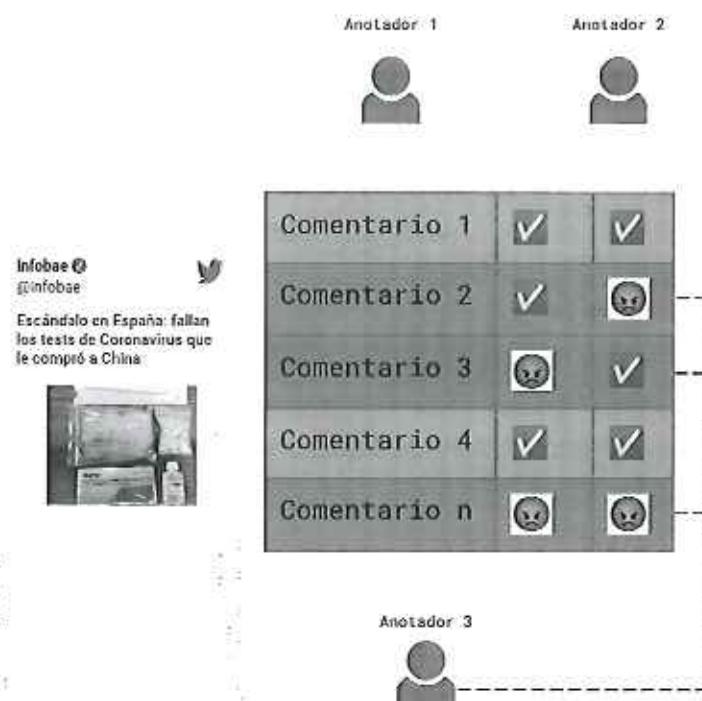


Fig. 5.4: Esquema de anotación. Caso en que ambos anotadores etiqueten los comentarios del artículo

*fuerza*  
 tengan conocimientos de inteligencia artificial, ciencia de datos ni relacionados de manera de no sesgar su tarea. También, que *sean* usuarios asiduos de redes sociales.  
*fuerza*  
*fuerza*  
 El proceso de reclutamiento constó en una breve entrevista donde corroboramos que *sean* hablantes nativos, les describimos la tarea mientras le mostrábamos la herramienta de etiquetado. Finalmente, se les solicitó hacer una prueba paga de leer el manual de etiquetado y anotar 10 artículos. Esto lo hicimos para corroborar la calidad de los etiquetadores. No rechazamos ningún etiquetador en este proceso. La tabla 5.6 brinda información desagregada sobre los 6 etiquetadores. Los etiquetadores reclutados tienen un perfil altamente escolarizado, con 2 etiquetadoras con experiencia previa, y siendo 2 activistas.

*creo que dentro eepba a que te refieres con esto, no?*

Luego de la entrevista, se les dio un feedback de su anotación, y se les reasignaron 5 de los artículos seleccionados junto a diez más (15 en total) para su anotación a modo de entrenamiento. Este es el único conjunto de artículos que fue anotado por la totalidad de los seis anotadores. Al finalizar esta etapa, se les brindó un nuevo feedback para ajustar el criterio de anotación, y se procedió a la etapa de anotación del dataset.

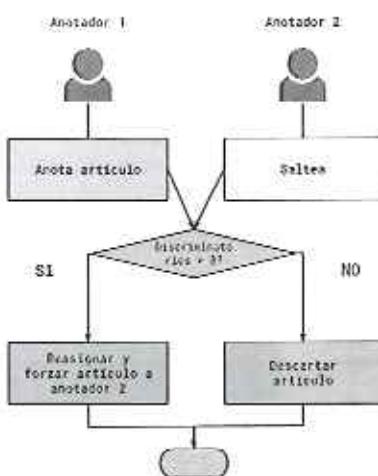


Fig. 5.5: Esquema de anotación. Caso en que un anotador saltee

#### 5.6.4. Esquema de anotación

La unidad de anotación del proceso de etiquetado es el artículo. Cada etiquetador, al serle presentado un artículo, tiene dos opciones: etiquetarlo o saltárselo. La idea de saltárselo es doble: evitar contenido poco "interesante" en términos de comentarios discriminatorios, o evitar contenido sensible para el anotador (algo que afortunadamente no ocurrió). En caso de etiquetarlo, tiene que asignarle las etiquetas correspondientes a cada comentario.

Una posibilidad que barajamos en un principio fue asignar el artículo completo a 3 anotadores. Sin embargo, esta modalidad sería altamente ineficiente dada la baja cantidad de contenido discriminatorio. Decidimos entonces ir por un esquema de desempate: dos anotadores anotan un artículo, y luego un tercero anota sólo aquellos donde al menos uno marcó que es discriminatorio. Esto da la posibilidad de que haya una tercera anotación incluso cuando dos previas marcaron que el comentario es discriminatorio, y lo hacemos para recolectar más información. Con este esquema de anotación, y teniendo en cuenta los números finales obtenidos del dataset, dedicamos 2.16 etiquetados por comentario versus 3 etiquetados por comentario de anotar tres veces todo. La figura 5.4 ilustra este flujo de anotación.

Cada artículo es, en primer lugar, asignado a 2 anotadores. Luego de esto, se solicita una tercera anotación pero sólo sobre los comentarios que tengan alguna de las dos etiquetas marcando contenido discriminatorio y no dando la posibilidad de saltar a este tercer etiquetador. Ahora ¿qué pasa si alguno de los dos anotadores saltea el artículo?. Tenemos dos casos. Si los dos saltan el artículo, entonces descartamos ese artículo. Ahora, puede ocurrir el caso de que uno lo salte y el otro lo anote: en ese caso, y en pos de maximizar el contenido discriminatorio encontrado o uno lo hace y el otro anota menos de 4 comentarios odiosos, entonces no pasa a 3ra anotación y lo descartamos del dataset. Si uno salteó y el otro anotador anotó 4 o más comentarios odiosos, entonces forzamos al primer anotador a anotar el artículo,

No entiendo el sentido de esto  
No es "infrente"  
también?  
¿Cuál info adicional  
querés pedir?  
por qué?

en promedio de  
que corresponden a  
poco claro

*¿Por qué pasa si tenía contenido sensible para el etiquetado?*

sin dar esta vez opción de saltar. La figura 5.5 ilustra el flujo para este caso.

Cada tweet es presentado con un preprocesado básico, consistente en reemplazar handles de usuarios por un token especial @usuario para evitar cualquier sesgo. Por ejemplo, si un usuario conocido como “odiador” (llamemos @hater) retuitea la noticia y otro responde a ese RT, aparece ese nombre de usuario lo cual podría condicionar al etiquetador.

Como resultado de este esquema, cada comentario de nuestro dataset puede tener dos o tres anotaciones, siendo los casos posibles los siguientes:

1. Dos anotaciones *(que es negativo?)* *{no discriminador?}* *(creo que es > 12 vez que usas esta palabra)*
2. Tres anotaciones, siendo al menos una que marque el comentario como discriminatorio

#### 5.6.5. Herramienta de etiquetado

Al no utilizar ningún servicio de etiquetado, optamos por desarrollar nuestra propia aplicación para el etiquetado de tweets. En ella, a cada etiquetador les fueron asignados progresivamente los artículos a anotar, agrupados en “lotes” para facilitar la tarea administrativa de la asignación.

La figura 5.6 muestra la interfaz presentada a los etiquetadores. Cada artículo es presentado al etiquetador junto a los comentarios asignados. Ante esto, el etiquetador puede elegir saltar el artículo o etiquetarlo. Si decide etiquetarlo, el etiquetador debe para cada comentario marcar usando un control de tipo switch:

1. Si el comentario contiene discurso discriminatorio
2. En caso de ser discriminatorio, marcar si llama a la acción
3. En caso de ser discriminatorio, marcar al menos una característica ofendida

Para el desarrollo de la aplicación usamos Django<sup>8</sup>, un framework de Python para desarrollo web, y Javascript plano. Como base de datos utilizamos SQLite ya que tenía una baja tasa de concurrencia (sólo 6 usuarios).

#### 5.6.6. Asignación

Llamamos asignación al procedimiento de asignar gold labels al dataset anotado [121]. Nuestro modelo consta de una etiqueta binaria marcando si el contenido es discriminatorio o no (notamos HS) en el primer nivel, y luego 9 etiquetas binarias: una para las llamadas a la acción (CALLS) y otras 8 para las características ofendidas. Recordemos que una anotación negativa sólo consta de HS negativo, mientras que una positiva consta de un HS positivo, una etiqueta para CALLS y al menos una etiqueta positiva de las características restantes.

Para este dataset, tomamos las siguientes decisiones:

1. Para la etiqueta de HS, realizamos la votación mayoritaria (2 o más votos para HS positivo, caso contrario HS negativo)

<sup>8</sup> <https://www.djangoproject.com/>



[MIRAR ARTÍCULO](#)

## Comentarios

Comentario	Etiquetas	Tipos													
Usuario Y la culpa es de españa ? O los pulos chinos ?	<p>¿Discriminación?</p> <table border="1"> <tr> <td>No</td> <td>Si</td> </tr> </table> <p>¿Llama a acción?</p> <table border="1"> <tr> <td>No</td> <td>Si</td> </tr> </table>	No	Si	No	Si	<table border="1"> <tr> <td>MUJER</td> <td>POBREZA</td> <td>ASPECTO</td> </tr> <tr> <td>LGBTI</td> <td>DISCAPAC</td> <td>CRIMINAL</td> </tr> <tr> <td>RACISMO</td> <td>POLITICA</td> <td>OTROS</td> </tr> </table>	MUJER	POBREZA	ASPECTO	LGBTI	DISCAPAC	CRIMINAL	RACISMO	POLITICA	OTROS
No	Si														
No	Si														
MUJER	POBREZA	ASPECTO													
LGBTI	DISCAPAC	CRIMINAL													
RACISMO	POLITICA	OTROS													
Usuario Chinos del orto	<p>¿Discriminación?</p> <table border="1"> <tr> <td>No</td> <td>Si</td> </tr> </table> <p>¿Llama a acción?</p> <table border="1"> <tr> <td>No</td> <td>Si</td> </tr> </table>	No	Si	No	Si	<table border="1"> <tr> <td>MUJER</td> <td>POBREZA</td> <td>ASPECTO</td> </tr> <tr> <td>LGBTI</td> <td>DISCAPAC</td> <td>CRIMINAL</td> </tr> <tr> <td>RACISMO</td> <td>POLITICA</td> <td>OTROS</td> </tr> </table>	MUJER	POBREZA	ASPECTO	LGBTI	DISCAPAC	CRIMINAL	RACISMO	POLITICA	OTROS
No	Si														
No	Si														
MUJER	POBREZA	ASPECTO													
LGBTI	DISCAPAC	CRIMINAL													
RACISMO	POLITICA	OTROS													

Fig. 5.6: Pantalla del etiquetador

Característica	Número	Llamadas a acción
RACISMO	2469	674
APARIENCIA	1803	34
CRIMINAL	1642	722
POLITICA	1428	136
MUJER	1332	18
CLASE	823	135
LGBTI	818	11
DISCAPACIDAD	580	4
TOTAL	8715	1451

Tab. 5.7: Datos desagregados por característica de los comentarios discriminatorios del dataset resultante. Se listan además la cantidad de llamados a la acción dentro de cada una. Notar que el total no corresponde con la suma de las columnas ya que un mismo comentario puede estar asignado a más de una característica.

- UNIFICAR
- 2. Si hay IIS, CALLS es positivo si es votación mayoritaria (ídem anterior)
  - 3. Si hay IIS, marco como positivas todas aquellas características marcadas por los anotadores

La primer decisión es la más obvia y razonable, pero las otras dos decisiones merecen alguna discusión. Para que sea un comentario considerado como HS, tiene que ocurrir que al menos dos etiquetadores lo marquen como tal. En ese caso, para que haya votación mayoritaria de CALLS, tiene que haber dos o más votos marcados como tal; en caso de empate, es decir, que un anotador marca que hay llamado a la acción y otro que no, marcamos que no hay llamado a la acción.

En el caso de las características, marcamos todas las que hayan marcado aquellos anotadores que hayan etiquetado HS. Esta decisión podría haberse tomado de otra manera; por ejemplo, sólo tomando aquellos casos donde haya cierto grado de coincidencia entre los comentarios. Sin embargo, al considerar que los límites entre las características son difusos (por ejemplo, apariencia y mujer tienen un grado de coincidencia, y a veces clasismo y racismo también) preferimos optar por este esquema.

Este me  
parece muy  
razonable!

## 5.7. Resultados

El dataset resultante consta de 1238 artículos etiquetados, y 56869 comentarios respectivamente, de los cuales 8715 contienen contenido discriminatorio según los criterios de asignación antes referidos. Podemos observar que aproximadamente 1 de cada 6 comentarios es discriminatorio; esto no es representativo del universo de notas periodísticas ya que recordemos que la selección de los datos no fue aleatoria. La tabla 5.7 contiene estos datos estadísticos.

De todos los tweets discriminatorios, tenemos en particular los llamados a la acción. La mayoría de estos está dirigido hacia la categoría CRIMINAL, muchos en la forma de llamados a matar a criminales y otros delincuentes. La categoría

¿Por qué?  
Es polémica  
entre densión.  
No alcanza  
con que  
alguno considere  
que es lindo?  
Va a caer?

Categoría	$\alpha$ de Krippendorff
Discurso de odio	0.579
Llamados a la acción	0.641
MUJER	0.783
LGBTI	0.920
RACISMO	0.929
CLASE	0.706
POLITICA	0.808
DISCAPACIDAD	0.849
APARIENCIA	0.871
CRIMINAL	0.931

Tab. 5.8: Tabla de acuerdo para la etiqueta de discurso de odio y diferentes características medido por  $\alpha$  de Krippendorff. El acuerdo sobre *discurso de odio* es reportado sobre el acuerdo total de la etiqueta respectiva. Para el resto de las características, el acuerdo es calculado sólo sobre aquellas anotaciones que marcaron discurso de odio.

*'personas de origen chino'?*

*RACISMO contiene acapara* también muchos llamados a la acción, mayormente *contrachinos*.

La tabla 5.8 reporta el acuerdo entre anotadores usando la métrica alpha de Krippendorff [76] mediante la librería `krippendorff`<sup>9</sup>. Reportamos el acuerdo para HS sobre todas las etiquetas, y luego todas las etiquetas del segundo nivel del modelo jerárquico (características y llamado a la acción) sólo sobre aquellas que hayan marcado que el comentario contiene HS. Esto es equivalente a calcular el acuerdo con una etiqueta faltante en el segundo nivel para las características y el llamado a la acción. Si bien el acuerdo sobre cada característica tiende a ser alto, debe leerse como el acuerdo sobre la razón detrás del hate speech. La mayor penalización queda reservada a la etiqueta de discurso de odio que tiene  $\alpha = 0,58$ , algo que podría marcarse como un acuerdo razonable teniendo en cuenta algunos valores observados en la literatura [119].

### 5.7.1. Co-ocurrencia de características ofendidas

De los 8715 comentarios odiosos, el 77 % de ellos (6777) tiene una sola característica ofendida marcada. Del resto, cerca del 20 % ~~de ellos~~ tiene 2 características ofendidas, y 220 comentarios tienen 3 o más. En la figura 5.7 podemos observar la matriz de co-ocurrencia entre las distintas características para aquellos comentarios que tengan más de una marcada. En ella podemos ver que la máxima co-ocurrencia se da entre la ~~S~~ característica MUJER y APARIENCIA, seguidos por RACISMO y CLASE, POLITICA y CLASE, y RACISMO y POLITICA.

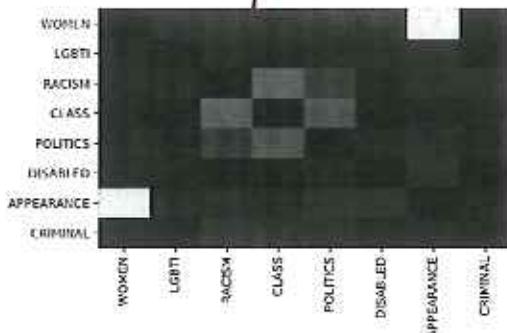
Otra forma de analizar la co-ocurrencia de comentarios es agrupando por artículos para observar cómo un mismo contexto puede suscitar distintos tipos de comentarios discriminatorios. La figura 5.7b ilustra las interacciones entre las distintas características por artículo. Podemos observar que tenemos en este mapa de

<sup>9</sup> <https://github.com/pln-fing-udelar/fast-krippendorff>

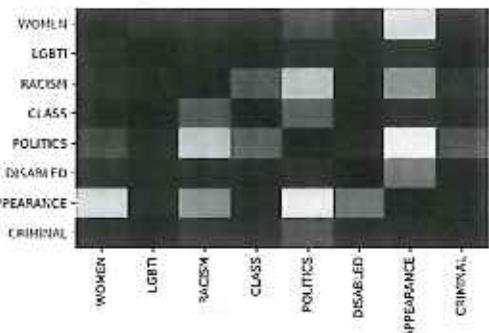
¿Por qué no son simétricas?

### 5.7. Resultados

79



(a) Co-ocurrencia de las características ofendidas en un comentario



(b) Co-ocurrencia de las características ofendidas en un artículo

Fig. 5.7: Matrices de co-ocurrencias de características ofendidas. La figura 5.7a muestra la co-ocurrencia dentro de un mismo comentario, y la figura 5.7b muestra la co-ocurrencia dentro de los comentarios de un mismo artículo. Más luminoso indica más co-ocurrencia

calor que tenemos mayor dispersión en las co-ocurrencias que reduciendo al análisis a sólo observar comentarios. Por mencionar algunas que no aparecen en la figura agrupada únicamente por comentarios, puede verse una mayor interacción entre discurso de odio RACISMO y POLITICA, y, quizás inesperadamente entre APARIENCIA y POLITICA. Las interacciones de la característica LGTBI se mantienen muy bajas, indicando que este tema suele estar concentrado en este tipo de ataques.

La tabla ?? muestra algunos ejemplos de comentarios con más de una característica ofendida marcada. Podemos ver que algunos están muy en el borde, justo en la frontera de las características (por ejemplo, APARIENCIA y MUJER en los ejemplos 1 y 2, o CLASE y RACISMO en el ejemplo 3). Otras instancias son conjunciones de expresiones discriminatorias: en el caso de la 5, tenemos una conjunción de RACISMO, CLASE, y LGTBI; en el caso de la 6, de APARIENCIA y LGTBI.

#### 5.7.2. Análisis por característica

En las tablas 5.10, 5.11 y 5.12 podemos observar ejemplos seleccionados de comentarios discriminatorios. Haremos a continuación un análisis cualitativo y observaciones generales sobre cada característica. Algo a ver es que los comentarios misóginos tienen en algunos casos grandes complejidades, como las acusaciones de "mentirosa" a una mujer que sufrió una violación,<sup>10</sup> apreciaciones a su cuerpo, entre otros comentarios de difícil apreciación.

Una categoría desafiante parecería ser los comentarios discriminatorios contra la comunidad LGTBI. Más allá de algunos insultos explícitamente ofensivos (mediante insultos del estilo trolo, trabuco, maricón, etc), hay muchos que tienen un contenido difícil de descifrar; en particular, aquellos comentarios contra personas trans. Muchos de estos mensajes hacen alusiones a su genitalidad o a su cuerpo en general, de manera metafórica o irónica, lo cual hace compleja su detección. A su vez, es

<sup>10</sup> <https://www.lavanguardia.com/gente/20181212/453520382646/denuncia-actor-juan-darthes-violar-thelma-fardis.html>

Índice	Contexto	Comentario
1	Ofelia Fernández apoyó al Gobierno en la polémica por los presos y apuntó a la Justicia que "odia a las mujeres"	Hijadept, ojala pronto recibas la visita de alguno de esos gusanos. Te van a quedar. Ganas de apoyar al. Gobierno? Larva ras-trera gorda. Decerebrada
2	"Es hora de ponerle límites al odio"   Por Victoria Donda	Justo ésta zurda mugrienta, ignorante y altanera...
3	Coronavirus en la Argentina: un video pone en evidencia la violación de la cuarentena en la Villa 1-11-14	Cierren esa nido de negros y napalm.. Has-ta reducís el crimen y el gasto público.
4	Fabiola Yáñez denunció a un periodista por publicaciones agraviantes	Claro si ofendel a la que se cuelga en el caño xq ahora crec ser primera dama? hay que ser peruna para dar asco y ser basuras bigote enseguida ordena como se metió en Facebook y en todo que culpa te.emos que saque la mujer del cabarute?
5	Los infectados en villas porteñas crecieron un 80 % en cuatro días	Ojalá que el virus penetre más en las vi-las y maten a todos esos delincuentes que viven ahí, hay paraguayos narcos, bolivia-nos que traen la droga de bolivia, y gente de mala vida. También hay travas que van a trabajar de noche a palermo.
6	El enojo de Moria Casán contra Rocío Oliva: "Mucha agua oxigenada, le quedó media neurona para jugar a la pelota"	Y la vieja Moria, mucha cirugía y estira-miento. de cara que parece un travesti
7	Ricky Martin: "Soy un hombre latino y homosexual viviendo en los Estados Unidos, soy una amenaza"	Ridículo perdiste tú rumbo das náuseas famosos eternos (víctimas) ándate a Puerto Rico entonces ahí no serás una amenaza

Tab. 5.9: Ejemplos con más de una característica ofendida marcada

claro que en muchos de estos comentarios es sumamente necesaria la información contextual para poder comprender el carácter abusivo de estos comentarios, ~~pero~~ en algunos casos ni siquiera queda explícito.

En el caso de la categoría CRIMINAL, se puede observar por un lado comentarios muy violentos ("bala", "mátenlos", "plomo") que necesitan el contexto para entenderse como ofensivos contra esa característica (si un artículo fuese sobre una plaga de osos o langostas no deberíamos considerarlos como tal). Por otro lado, algunos comentarios son más difíciles de descifrar y dependientes del contexto, como las celebraciones ante el abatimiento de un preso o criminal ("bravo", "felicitacio-nes!") que parecen inofensivas hasta que se lee el contexto de la noticia. De hecho, a diferencia de otros comentarios, estos mensajes tienen una polaridad positiva y contenido altamente irónico, indescifrable sólo observando el texto del tweet.

En el caso de racismo (la categoría más marcada del dataset) hay una fuerte cantidad de comentarios discriminatorios contra la comunidad china. Esto es espe-rable por el brote racista debido a la pandemia del COVID-19 que obviamente tuvo su replica en las redes sociales [61]. El racismo de las categorías que más llamados a la acción tiene, muchos del estilo de tirar bombas, aniquilar, etc a China o a la

En este caso falta el predicho.

Tipo	Contexto	Texto
MUJER	Por el mundo: Florencia Peña mostró su lujosa nueva casa con bar, mueble y pileta	@usuario Cuando succionas las correctas
	El exabrupto de Rodolfo Barilli con una modelo que se metió en el baúl de un auto para violar la cuarentena	@usuario Barilli, ojo es mujer, en cualquier momento te caen las hordas de feminazis, pero estoy al 100% CON VOS!!
	Vidic: tres mujeres forzaron la puerta de un edificio en Palermo y desvalijaron dos departamentos	@usuario 3 feministas , claramente es el perfil de las feministas
	Británica llegó a Perú por 10 días y se quedó por amor: vive sin agua potable y entre insectos	@usuario Pero empomada todo el dia...
	"Me esterilicé, pero no odio a los niños": mi vida dentro del movimiento "libré de hijos"	@usuario Es que sos incogible..quien te va a hacer un hijo....alguii ciegooooo
	Mia Khalifa: actuó en videos porno durante pocos meses, cobró fama mundial y ahora lucha por borrar su pasado	@usuario JAJAJAJAJAJA SEGUILA CHUPANDO....
	Thelma Fardin revictimizada: las víctimas de violación no pueden quedar presas del estigma mientras los violadores siguen libres	Forra anda a laburar a PornHub que la caras te salen bien. Inmunda sos una mentira
	Narda Lepes: "Mil veces me tocaron el culo en las cocinas de los restaurantes"	@usuario Cada dia dudo mas si Dantes la violo a la estupida 😂
	Un policía se defendió de un robo en Pompeya, mató a un ladrón y estuvo dos días preso	@usuario Tenes lindo culo?
	"Cristian, mirá lo que hago por vos": quién es la modelo que se metió en el baúl de un auto para violar la cuarentena?	@usuario Que jueza hija de mil puta ojalá t afansen a vos y t maten un ser querido a ver qué haces LIÉGUA de mierda puta
LGBTI	La aberrante arenga machista que dio el preparador físico de Güemes de Santiago del Estero	@usuario Jajaja modelo Hot jaja Telosull Mami diría la Moria !! Jaja ridícula tafaldegaver
	Por qué Flor de la V no continuó en Mujeres de eltrece, tras la salida de Claudia Fontán	@usuario Que aliado te volviste Clarín, vos te culcas la que tiene olor a pingo (feminazi)
	Histórico: María Gómez fue habilitada y será la primera jugadora trans en el fútbol argentino	@usuario y..porque no es mujer, más claro echale agua
	La historia de la modelo colombiana trans que besa la panza de su esposo embarazado de ocho meses	@usuario Unos huevos bárbaros tiene esta mina!!!!
	Luis Novaresio le dedicó un romántico mensaje a Braulio Bauab por su cumpleaños	@usuario Un macho besando a otro macho
	Eugenio Zaffaroni le contestó a Sergio Berni tras la polémica por las domiciliarias: "Es el populacherismo vindicativo que llenó las cárceles"	@usuario Guacale
	La impactante historia de la tenista trans que hoy es la N° 3 de Argentina en la categoría senior femenina	@usuario cuando se muere este viejo trolo enfermo
	Joe Biden nominó a Rachel Levine, una mujer transgénero, para que sea su subsecretaria de Salud	@usuario Vergonzoso que las mujeres toleren esto.\nEse tenista debería jugar con hombres o a lo sumo, en un torneo de sujetos como él.
	Así luce el actor Elliot Page tras declararse trans	@usuario Este presidente es la dejeneración total del mundo
	El abuelo que a los 90 años confesó: "Soy gay, soy libre y estoy afuera"	@usuario Tiene Bija? No. Tiene Concha? Si. Es mujer entonces

Tab. 5.10: Ejemplos discriminatorios del dataset contra mujeres y la comunidad LGBTI.

Tipo	Contexto	Texto
RACISMO	Coronavirus: las terribles imágenes del increado donde se originó la pandemia	@usuario Hay que matarlos hijos de puta
	Malestar en Washington con el Gobierno argentino porque no dejó atracar al buque más moderno de la guardia costera de Estados Unidos	@usuario Amo ver los sudacas que se creen yankis enojados por esto.
	Milagro Sala: "Seguimos presos, los que nos gobiernan tienen que cambiar los cabezas"	@usuario Negra, seguis presa por chorra.
	Al menos 7 muertos en China a causa de un virus transmitido por garrapatas	@usuario Que no venga ningún chino más a la Argentina! Por favor! Ya Basta!
	En China comenzó el tradicional Festival de Carne de Perro a pesar de la pandemia de coronavirus y una ONG intenta salvarlos	@usuario No soy racista, pero hay que matar a todos los chinos
	Científicos identificaron en China otro virus respiratorio "con potencial para convertirse en pandemia" Coronavirus. Yanzhong Huang: "Es bastante probable que ya se esté gestando un Covid-21"	@usuario Nos infecstan a Todos!!!
CRIMINAL	Denunciaron la nueva maniobra de China para ocultar las verdaderas cifras del coronavirus	@usuario Bombas urgente a esta maldita raza
	Villa Mascardi: impresionante operativo con tanquetas blindadas para que una fiscal ingresara a una zona controlada por mapuches	@usuario Mundialmente mantenemos china xq todo viene de ahí y hoy estamos fundidos y en emergencia... #ChinaVirus no quiero ver un #chino x mucho tiempo!
	Aberrante: un político de Misiones admitió haber esclavizado y violado a sus tres hijastros	@usuario Solo se requiere una bala.
CLASE	Rosario: un grupo de vecinos linchó y mató a golpes a un joven acusado de robar autos	@usuario esta perfecto, ejemplo a los demás
	El panadero que mató a un ladrón en La Matanza: "No soy un asesino, estoy arrepentido"	@usuario Que dice señor ! No sé arrepienta, que hizo una obra de bien.Era su vida o la del delincuente.
	Video: salió de la cárcel por el coronavirus y murió de un tiro el mismo día al festejar su libertad	@usuario Justicia divina!!
	Le sacó el arma al ladrón que lo asaltó, lo corrió y lo mató de un disparo: quedó detenido	@usuario Buenísimo vamos por el exterminio total de estos primates.
CONTRACRIMEN	La Justicia ordenó el desalojo de la masiva toma de terrenos en Guernica	@usuario Lanzallamas y a otra cosa
	Hubo tensión en la Quinta de Olivos entre un grupo que apoyaba a Alberto Fernández y manifestantes del banderazo contra el Gobierno	@usuario PLANEROS Y BARRABRAVAS
	Organizaciones sociales cortaron la avenida 9 de Julio: reclamaron un salario mínimo de \$ 45.000	@usuario Vayan a lo laburar hdp.
	La historia de una familia de cartoneros en la toma de Guernica: "Por primera vez sentimos que tenemos un hogar"	@usuario Bala.
	El Gobierno autorizó la apertura de las escuelas porteñas para las elecciones de Bolivia	@usuario No sería mejor deportar a los bolivianos indocumentados?.además nos suman pobreza e indigencia
	Coronavirus en Argentina: un dirigente radical deseó que la pandemia "haga una limpieza étnica" con "negros de La Matanza"	@usuario Es el deseo de todo argentino de bien
	El Polo Obrero realiza un corte en la Panamericana en contra de la flexibilización de la cuarentena y en reclamo de aumentos a los planes sociales	@usuario Clarísimo que no quieren laburar y quieren vivir dc nosotros!
	Coronavirus en la Argentina: movimientos sociales reclaman asistencia alimentaria en el Obelisco	@usuario Anda a laburar lpqtp

Tab. 5.11: Ejemplos discriminatorios del dataset por motivos de clase, racismo, o contra criminales.

Tipo	Contexto	Texto
POLITICA	Confirman una mutación en el coronavirus que puede hacerlo 10 veces más contagioso que la cepa original de Wuhan  Murió un nieto recuperado por Abuelas de Plaza de Mayo: los mensajes de Alberto Fernández y Cristina Kirchner  Última encuesta: ¿Qué mujer superó a Alberto Fernández en imagen positiva? Cómo es la cerveza "peronista" que el Chacho Coudet le regaló a Alberto Fernández El descargo de Nicolás Wiñazki después de que Vero Lozano se burlara de él: "Quizás le afecta la cuarentena"	@usuario ME ALEGRO MUCHÍSIMO.\nÓ-JALÁ LLEGUE PRONTO A ARGENTINA Y ARRASE CON TODO.\nPODRÍAMOS VER AL FIN ALGO MÁS DAÑINO QUE EL CÁNCER PERONISTA Y SU METÁSTASIS KIRCHNERISTA. @usuario Un planero menos.
APARIENCIA	Axel Kicillof recomendó una "cuarentena previa" de 14 días para "llegar sanos a Navidad y Año Nuevo" Video indignante: piba violó la cuarentena y viajó en el baúl de un taxi para ver a un chico  El senador José Mayans defendió a Gildo Insfrán: "En pandemia no hay derechos" El video sexy de More Rial en corpiño El sensual paseo en moto de Florencia Peña: "Próxima parada: tu casa";	@usuario Chuoame la verga enano moishe @usuario Habría que buscar también y meter en cana al cirujano que le hizo la nariz!! Parece Michael Jackson la loca!!! @usuario Volve al gancho , docer
DISCAPACIDAD	Patricia Bullrich pidió ser drásticos con los docentes: "El que no va, tendrá que ser reemplazado" El abuelo que a los 90 años confesó: "Soy gay, soy libre y estoy afuera" Elisa Carrió dijo que "ninguna pandemia es excusa para suspender la República" y advirtió que "vienen por los campos" Florencia Kirchner y su posteo a favor de la amistad: "Nunca entendí la desesperación por la pareja"	@usuario Estragos del tinto @usuario Alzheimer o demencia senil!!!!
INCITACIÓN	Harán un listado de los presos en situación de riesgo por el coronavirus para evaluar si deben salir de prisión  La advertencia de Juan Grabois: "Van a haber 1, 5, 20 Guernicas" Otro caso de peste bubónica enciende las alarmas en China Coronavirus: afirman que volvió la venta de carne de murciélagos en China Villa Mascardi: impresionante operativo con tanquetas blindadas para que una fiscal ingresara a una zona controlada por mapuches Coronavirus en China: la ciudad de Shenzhen prohíbe comer perros y gatos	@usuario Si sacan a la paciente psiquiatrica es porque están hasta las manos.\n#CarcelACambiamos @usuario La enfermita está mejor que yo,y no se calienta por la hija @usuario @usuario Todo al revés! Si hay alguno con coronavirus PONGANLO EN EL MEDIO! @usuario @usuario Habrá 100 paredones @usuario Una atómica a China... @usuario Boicot a todo producto chino!!! @usuario El Diálogo se Inicia con BALAS y Finaliza con LA ÚLTIMA @usuario habrá alguna manera de erradicar a estos tipos del mundo ?

Tab. 5.12: Ejemplos discriminatorios del dataset. INCITACIÓN refiere a los llamados a realizar algún tipo de medida contra el grupo o la persona atacada.

comunidad de dicho país, o llamados a tomar medidas “blandas”, como “no ir a comprarles a los supermercados”.

Algunas de las categorías como las de la tabla 5.12 poseen características más elementales y menos interesantes, basadas en agravios directos y explícitos. A priori, uno podría pensar que son las características que menos necesidad de contexto revisen, ya que –mayormente– su carga de odio es notoria y centrada en insultos. Algo a notar es que en muchos de estos insultos se usan técnicas de camuflaje (“tafaldegaver”, falta de verga, “docer”, cerdo), que dificultan su detección por las técnicas actuales.

### 5.8. Discusión

En este capítulo, describimos la construcción de un dataset contextualizado de lenguaje discriminatorio. Separamos la construcción de este dataset en tres etapas: recolección, selección, y anotación. Con respecto a la recolección, esta se basó en recolectar respuestas a noticias periodísticas posteadas en Twitter por los principales medios de noticias de Argentina.

*¿cuén o éste?*

La etapa de selección presentó cierta complejidad, teniendo en cuenta que el discurso de odio no está distribuido uniformemente entre los distintos artículos periodísticos. Exploramos distintas alternativas para esta tarea, tanto observando los tópicos como los comentarios a éste. Decidimos elegir los artículos en base a sus comentarios potencialmente discriminatorios usando un lexicon de expresiones. Las evaluaciones subjetivas de los dos métodos –seleccionar en base al artículo vs seleccionar en base a los comentarios– resultaron en una mejor calidad de artículos seleccionados en base a los comentarios. A diferencia de otros trabajos, estos lexícones no marcaron los comentarios a etiquetar, sino los artículos: los comentarios se seleccionaron –ahora sí– de manera aleatoria entre los artículos ya seleccionados.

*wando*

Para realizar la tarea de etiquetado, definimos un modelo de anotación jerárquico y granular para la tarea teniendo en cuenta lo discutido en la sección 4.7. El hecho de anotar las características –y no sólo la etiqueta binaria de presencia de discurso de odio– es algo que no suele realizarse en muchos trabajos. Seis etiquetadores nativos de la variedad dialectal rioplatense realizaron la tarea de anotación bajo un esquema de 2 anotaciones + desempate. Como producto, obtuvimos un dataset de cerca de 57,000 comentarios repartidos en 1,238 artículos, una cantidad de tamaño considerable en términos de comentarios aunque no tengamos parámetro de comparación ya que no existen muchos datasets similares. De los comentarios, alrededor de 8,000 comentarios tienen contenido discriminatorio (una tasa de 1 cada 6).

Un análisis exploratorio de los comentarios discriminatorios muestra ejemplos complejos y ricos, algunos de ellos altamente dependientes del contexto. Finalmente, un análisis de la co-ocurrencia de las características ofendidas da muestra de que el conjunto de datos anotado posee diversidad en sus instancias, con comentarios conteniendo múltiples tipos de discriminación y artículos que poseen comentarios odiosos de diversa naturaleza. Podemos especular que tanto el texto (el comentario en sí) como el contexto (el tweet del medio periodístico y su artículo periodístico) contienen información valiosa para poder distinguir entre las distintas categorías discriminatorias.

### 5.9. Conclusión

En este capítulo hemos desarrollado el proceso de construcción de un dataset contextualizado de discurso de odio en redes sociales. Para ello, recolectamos respuestas de usuarios a noticias periodísticas posteadas en Twitter por los principales medios de noticias de Argentina. Describimos detalladamente el proceso de su construcción –tanto en la recolección, selección y anotación de los datos– haciendo éje en las distintas dificultades que fuimos encontrando y posibilidades de mejora.

Como resultado, obtuvimos un dataset de más de 8,000 comentarios discriminatorios, anotados de manera granular de acuerdo a las diferentes características ofendidas. Mediante evaluaciones subjetivas y análisis de las co-ocurrencias de las características, podemos afirmar que este dataset posee comentarios con cierta complejidad, discurso de odio explícito e implícito, y artículos que suscitan distintos tipos de reacciones discriminatorias, lo cual aporta a la riqueza de los datos.

yo diría  
notable

Con este dataset como insumo, pasaremos ahora a analizar algo que discutimos en la sección 4.7: la contextualización de los mensajes. Este tema ha sido poco abordado en la literatura, es por ello que pasaremos ahora a analizar el impacto de poseer información contextual para la tarea de detección de discurso de odio.

### 5.10. Notas

En el apéndice C pueden encontrarse el manual de etiquetado, como así información adicional sobre la construcción del dataset. La herramienta de etiquetado puede encontrarse en <https://github.com/finiteautomata/news-labelling>.



## 6. EXPERIMENTOS DE DETECCIÓN CONTEXTUALIZADA

En base a las discusiones previas, en este capítulo analizaremos el impacto del contexto en la tarea de detección de discurso de odio en redes sociales, algo que ha recibido poca atención en trabajos previos. Utilizaremos el dataset construido en el capítulo 5 que está basado en comentarios de artículos periodísticos en Twitter, y que nos brinda información adicional a cada comentario tanto por el tweet del medio periodístico como el contenido del artículo. Para evaluar si la adición de contexto resulta en una mejora en la detección de discurso de odio realizaremos experimentos de clasificación con modelos que sólo consuman el comentario en cuestión, y otros que a su vez consuman algún tipo de contexto asociado. Por la naturaleza de nuestro dataset, tenemos dos tipos de contexto naturales: uno “corto”, que contiene el tweet del medio periodístico, y otro “largo” que incorpora además el cuerpo del artículo asociado.

La anotación del dataset, con información detallada de las características ofendidas, nos permite salir de analizar exclusivamente la existencia de discurso de odio sino que podemos pedir más detalle sobre la ofensa cometida. Proponemos entonces dos tareas: una tarea de detección binaria, donde sólo predecimos si hay o no discurso de odio; y una tarea de detección granular, donde además predecimos todas las características ofendidas (potencialmente más de una). Para estas tareas, propondremos algoritmos de clasificación sobre modelos pre-entrenados de lenguaje, concretamente sobre *BETO* (la versión en español de *BERT*). Estos modelos tienen incorporados naturalmente la posibilidad de consumir dos entradas, con lo cual son ideales para nuestros experimentos.

Evaluaremos los resultados tanto en términos de la performance de las distintas configuraciones de nuestros clasificadores, como así también realizando análisis de error comparativos entre los modelos contextualizados y los no contextualizados. También analizaremos las dificultades en general que presenta la detección de este fenómeno sobre comentarios de notas periodísticas.

### 6.1. Trabajos previos

Como mencionamos en la sección 5.1, no se ha dado demasiada atención en la literatura a la utilización de contexto en la detección de discurso de odio y otros fenómenos similares (como la detección de toxicidad, por ejemplo). En dicho capítulo se puede encontrar una descripción de algunos de los datasets que sí contienen algún tipo de contexto. Pasamos ahora a describir los algoritmos de detección que utilizaron en los trabajos correspondientes.

Gao and Huang [46] propone dos tipos de modelos: regresiones logísticas y redes neuronales recurrentes. Para los modelos de regresiones logísticas, usan como inputs bolsas de palabras, bolsas de caracteres, vectores semánticos producidos con Linguistic Inquiry and Word Count (LIWC) [111] y features de un lexicon de emociones [99]. Por otro lado, utiliza LSTM bidireccionales con mecanismo de atención de Bahdanau [6] usando embeddings *word2Vec* de dimensión 100.

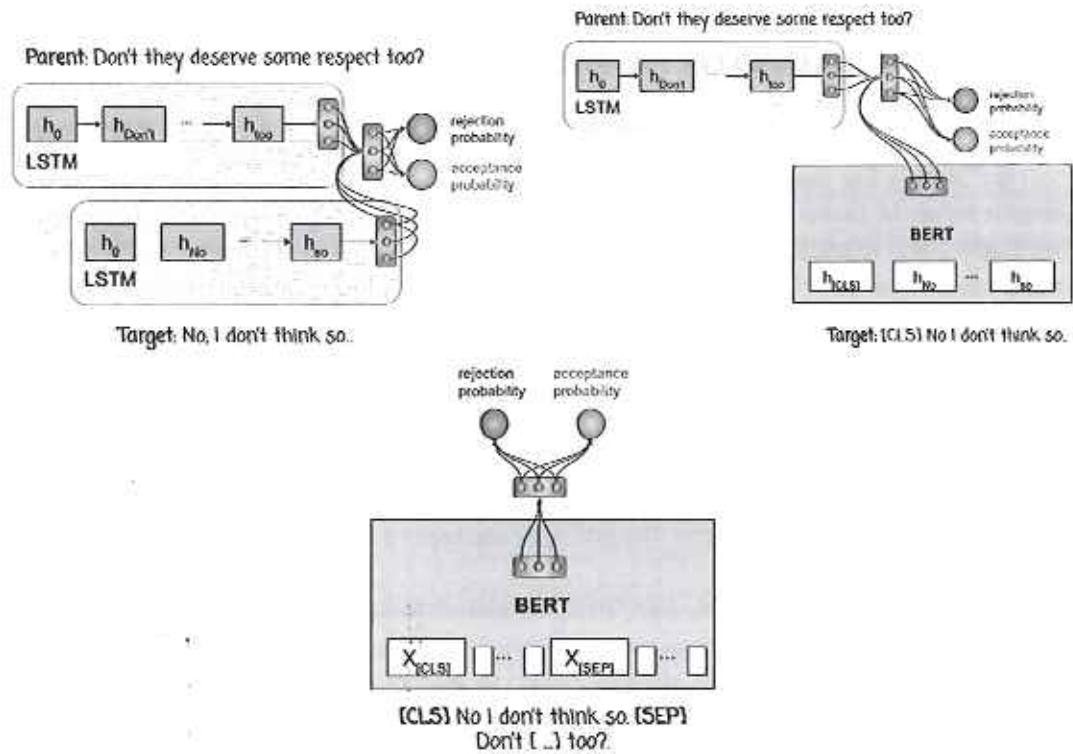


Fig. 6.1: Clasificadores que consumen contexto propuestos por Pavlopoulos et al. [109]. Los dos primeros clasificadores proponen una arquitectura de dos encoders, uno para el texto y otro para el contexto usando bi-LSTMs y BERT como posibilidades. El tercer clasificador propuesto es un BERT usando su estructura natural para codificar dos oraciones separadas por el token *SEP*.

Un punto criticable de este trabajo es que utiliza el nombre de usuario como feature; algo que a priori no suele hacerse ya que permitiría “prejuzgar” a un usuario antes que por el contenido de sus tweets. Si bien es cierto que la información de usuarios y sus conexiones es valiosa, introducir esta información a nuestros modelos da lugar a posibles correlaciones espurias que es preferible evitar.

En la sección 5.1 hemos descripto el dataset construido por Pavlopoulos et al. [109]. Nos detendremos un momento para analizar sus experimentos de clasificación ya que guardan importantes similitudes con lo que haremos en este capítulo. En ese trabajo se obtuvieron dos datasets de entrenamiento: uno en el cual los etiquetadores tenían información del contexto y otro en el que no, mientras que el dataset de test fue etiquetado viendo el contexto, considerando que el etiquetado es de mejor calidad usando más información. Los autores plantearon dos preguntas sobre esta base: ¿mejora la performance de los clasificadores que son entrenados con el dataset etiquetado con contexto? ¿mejora la performance de los clasificadores consumiendo información contextual? Con estas preguntas, tenemos que tomar dos decisiones: dataset de entrenamiento con o sin contexto, y clasificador con o sin contexto. Tenemos 4 posibles combinaciones de experimentos, sin aún considerar posibles técnicas de clasificación.

Nosotros? o los autores de [109]?

*¿quién?*

Para cada una de estas combinaciones, se consideraron técnicas del estado del arte de clasificación. Para aquellos clasificadores que no consumen contexto, las opciones son las mismas que hemos visto en capítulos anteriores: bi-LSTM o BERT. Para aquellos que sí consumen contexto, se evalúan dos estrategias: una, concatenar con algún carácter, y otra usando dos encoders distintos para el contexto y el texto. A su vez también utilizan la API Perspective de Google, que finalmente obtiene los mejores resultados en términos de performance. En todas las combinaciones posibles, si bien hay una mejora en la performance medida con ROC-AUC al usar contexto en ambas formas, esta no es estadísticamente significativa.

Algo a mencionar (que retomaremos en este y en el siguiente capítulo) es que usan dos versiones de *BERT*, una usando los pesos del modelo de *BERT*, y otro haciendo un ajuste de dominio corriendo la tarea de MLM sobre un dataset grande y no etiquetado. En el caso de el trabajo mencionado, sólo hacen un fine-tuning sobre comentarios sueltos del dataset de Civil Comments. Esto podría tener algún efecto deteriorando la performance al usar contexto; sin embargo, en el *BERT* a secas (sin hacer ajustes) tampoco se observa mejora significativa en la performance.

Algunas limitaciones marcadas por los autores son:

- Contexto muy pequeño: sólo el título más el comentario previo
- Se ignora el hilo completo de comentarios
- Los datos fueron sampleados aleatoriamente

*Expertos!  
No entiendo esto*

En Xenos et al. [150], continuación de este trabajo, reetiquetaron el dataset de Civil Comments usando contexto y –como mencionamos en la sección 5.1– presentaron una nueva tarea de detección de sensibilidad al contexto. Usando la API Perspective (y la estrategia de concatenación “básica” con algún carácter), notaron que la performance del clasificador que consume el contexto mejora con respecto al que no lo hace a medida que restringimos nuestra atención a comentarios más “sensibles al contexto” (de acuerdo a la métrica definida por los autores).

## 6.2. Tareas de clasificación propuestas

*cosas*

Para analizar el impacto del contexto en la detección de discurso de odio, y teniendo en cuenta que contamos de un dataset con anotaciones granulares sobre las características ofendidas, proponemos dos tareas de clasificación:

1. **Detección binaria:** Dado un tweet y su contexto, predecir si contiene contenido discriminatorio.
2. **Detección granular<sup>1</sup>:** Dado un tweet y su contexto, predecir las características ofendidas (si hay alguna) y si contiene un llamado a la acción.

Puede pensarse la tarea de detección binaria (la que usualmente se aborda en la literatura sobre el tema) como una relajación de la tarea detallada: mientras la

<sup>1</sup> En inglés usamos la denominación *fine-grained*, aunque no hay una traducción precisa para este término en español.

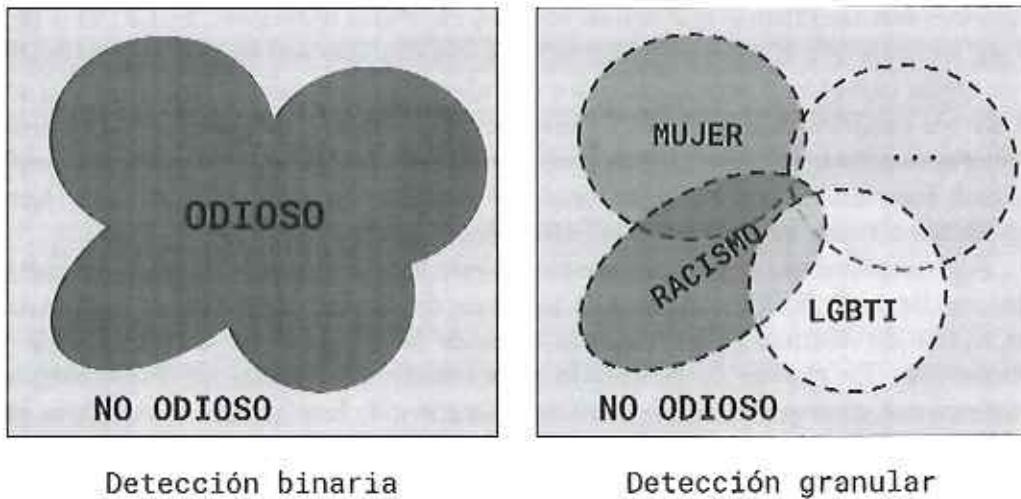


Fig. 6.2: Tareas propuestas de detección de discurso de odio. La tarea de detección binaria consta de predecir si un tweet contiene contenido discriminatorio, discriminando la frontera conjunta. En la tarea granular, predecimos por separado cada una de las características ofendidas, pudiendo haber más de una o ninguna.

primera sólo nos permite detectar si hay o no contenido discriminatorio, la segunda ~~nos~~ requiere información más precisa acerca de las ~~características~~ ofendidas. Esta segunda tarea es posible dado que el dataset ~~que~~ construimos ~~contiene~~ esta información, algo usualmente faltante en otros trabajos. La tarea granular nos permite a su vez tener mayor entendimiento de la salida e interpretar mejor sus errores. La figura 6.2 ilustra las dos tareas propuestas. Mientras en la tarea binaria sólo debemos decidir la frontera sobre si el contenido es discriminatorio o no, en la tarea granular necesitamos decir en cuál de todas las intersecciones ~~está~~ el comentario y su contexto.

vs sin tilde.

Plantéandolos como problemas de clasificación, la detección binaria consta de predecir una sola etiqueta binaria, mientras que la tarea granular consta de  $n$  etiquetas binarias; es decir,  $n$  problemas distintos de clasificación. Vale mencionar que, entendiendo una tarea como una relajación de la otra, si tenemos un clasificador entrenado para la tarea granular podemos construir un clasificador para la tarea binaria tomando la disyunción lógica de sus salidas; o dicho más coloquialmente, poniendo una compuerta OR sobre la salida del clasificador granular. Retomaremos esta idea más adelante al hablar de cómo evaluamos nuestras técnicas de clasificación para cada tarea.

### 6.3. Modelos de clasificación

Para las dos tareas, entrenamos varios clasificadores basados en *BETO* [26]. Respecto a la información contextual, tenemos tres tipos de entrada por instancia:

del medio periodístico

el comentario sin ningún tipo de contexto, con contexto simple (el tweet al que responde), y con contexto largo (el tweet al que responde + el texto del artículo periodístico). Usamos el token especial BERT [SEP] para separar el contexto y el texto analizado. [Este token es el que [SEP] se usa para la tarea de predicción de la siguiente oración (tarea NSP) en el preentrenamiento al estilo BERT (ver la sección 2.3.2 para más información).

Para la tarea de clasificación binaria, la salida de los clasificadores es la salida estándar de clasificadores del tipo *BERT*: una capa softmax para las dos posibles clases (ver la sección 3.6 o 4.5.2 para más información). En cuanto a la tarea de detección granular, lo planteamos como la predicción de 9 variables distintas: llamado a la acción (CALLS) y las 8 características ofendidas: MUJER, RACISMO, CLASE, LGTBI, CRIMINAL, ASPECTO, DISCAPACIDAD, POLITICA. En lugar de entrenar un clasificador diferente para cada característica, entrenamos un BERT de múltiples salidas, compartiendo todos sus pesos con la excepción de 9 capas lineales diferentes para cada salida. La función de costo utilizada es:

$$J = \sum_{c \in CHAR'} J_c$$

donde  $CHAR'$  es el conjunto de todas las características protegidas y además la variable de llamada a la acción, y cada  $J_c$  es la función de entropía cruzada. Compartir los pesos entre todas las salidas tiene dos objetivos: primero, poder generar un modelo más compacto (de otra forma serían 9 BERT distintos) y segundo, para compartir información común entre las distintas características atacadas, ya que consideramos que guardan similaridades y muchas de ellas tienen cierta intersección, como hemos visto en el capítulo anterior. Para tener costos computacionales más amigables, limitamos nuestras secuencias a 128, 256 y 512 tokens para el modelo sin contexto, el modelo que consume el tweet, y el modelo de tweet + cuerpo respectivamente. La figura 6.3 muestra el modelo de clasificación para la tarea granular, junto a los 3 tipos de entrada descriptos.

Una práctica cada vez más extendida en trabajos del área de clasificación de documentos es realizar una adaptación de dominio sobre textos relacionados a nuestro. Esto se realiza corriendo la tarea de masked language model (ver sección 2.3.2) sobre un dataset grande y no supervisado relacionado a nuestro dominio, o directamente sobre el dataset de la tarea si esto no está disponible [58]. En la sección 7.1 del siguiente capítulo haremos una reseña más extensa de esta técnica, pero por lo pronto podemos entender que ajusta el modelo de lenguaje a los textos disponibles; en este caso, *BETO* fue entrenado en Wikipedia y textos formales, con esta técnica lo ajustaremos a nuestro dominio particular de comentarios en Twitter a notas periodísticas.

Para lo que nos concierne en esta sección, Pavlopoulos et al. [109] realizó una adaptación de dominio sobre los comentarios del corpus de *Civil Comments* (a lo que denomina BERT-CCTK). Esta adaptación la realizó únicamente con los comentarios, sin utilizar ningún tipo de contexto. Proponemos a diferencia de este trabajo, 3 tipos de adaptaciones: una adaptación sin contexto, una adaptación con el contexto del tweet, y una adaptación con el contexto del tweet y el cuerpo de la noticia.

Realizamos el ajuste utilizando el sobrante de la recolección de datos del anterior

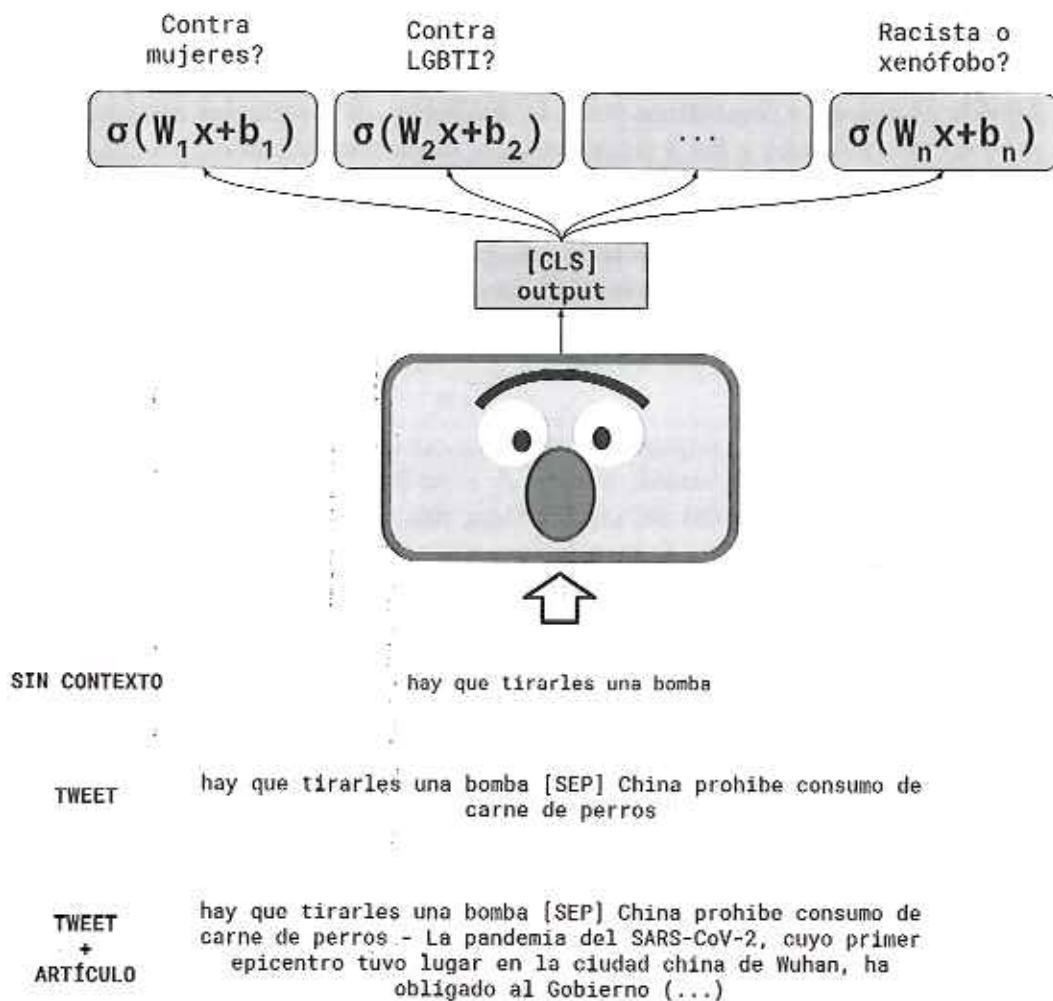


Fig. 6.3: Modelo de multiclasiificación para la tarea granular. Los modelos son entrenados de 3 maneras distintas: sin contexto (sólo el comentario), con el contexto del tweet, y con el contexto del tweet y el texto del artículo.

Hiperparámetro	Valor
Steps	10K
Batch size	2048
max seq length	128, 256 y 512
$\beta_1$	0,9
$\beta_2$	0,98
c	$10^{-6}$
decay	0,01
Peak LR	$4 * 10^{-4}$
warmup ratio	0,1

Tab. 6.1: Hiperparámetros para la adaptación de dominio de BERT

	Entre anotadores		Gold	
	F1 mean	F1 median	F1 Mean	F1 Median
ODIO	0.653	0.675	0.829	0.851
CALLS	0.434	0.495	0.701	0.842
MUJER	0.490	0.468	0.741	0.759
LGBTI	0.596	0.577	0.846	0.915
RACISMO	0.653	0.644	0.871	0.879
CLASE	0.443	0.444	0.722	0.732
POLITICA	0.461	0.436	0.795	0.815
DISCAPACIDAD	0.550	0.600	0.813	0.842
APARIENCIA	0.649	0.743	0.831	0.915
CRIMINAL	0.527	0.580	0.841	0.929
Macro F1	0.534	0.554	0.796	0.848

Tab. 6.2: Estadísticos de las cotas de performance entre anotadores. Cada característica es tomada como una etiqueta binaria independientemente del cálculo de la métrica para discurso de odio. La Macro F1 es el promedio de los F1 todas las características y la F1 de la llamada a la acción (CALLS). Las dos primeras columnas marcan las métricas medidas entre anotadores, y las dos últimas la de los anotadores contra la etiqueta gold

Agrego un apartado memoria: "que no fueron etiquetados ni incluidos en el dataset de entrenamiento", o algo así.

capítulo: alrededor de 288k artículos y 5.1M comentarios<sup>2</sup>. La tabla 6.1 contiene los hiperparámetros utilizados. Estos ajustes los realizamos sobre una TPU v2-8 y una máquina de Google Colab Pro por alrededor de 10 hs (en su largo de cadena máximo).

Preprocesamos ambos tweets – contexto y texto – utilizando las técnicas descritas en la sección 3.5: conversión de usernames a un token especial (usuario), tratamiento de hashtags (separación e inserción de un hashtag especial), y conversión de emojis a su representación textual.

Métrica	Sin Contexto		Tweet		Tweet + Cuerpo	
	¬FT	FT	¬FT	FT	¬FT	FT
Accuracy	0.889	0.899	0.902	0.910	0.904	0.905
Precision	0.678	0.718	0.731	0.748	0.739	0.728
Recall	0.568	0.602	0.601	0.653	0.611	0.641
F1	0.618	0.655	0.660	0.697	0.669	0.681
Macro F1	0.776	0.798	0.801	0.822	0.806	0.813

Tab. 6.3: Resultados de los experimentos de clasificación para la tarea *binaria* de detección de discurso de odio. Cada modelo es un BERT con 3 posibles entradas: sólo el comentario (*Sin contexto*), el tweet de la noticia a la cual responde el comentario (*Tweet*), y el tweet más el cuerpo de la noticia (*Tweet + Cuerpo*). Para cada una de estas posibilidades usamos dos versiones: una sobre BETO ( $\neg$ FT) y otra sobre BETO ajustado al dominio (FT)

### 6.3.1. Performance humana de la tarea

si existe

Como observamos en la anterior sección, la tarea de detección de lenguaje discriminatorio contiene una alta cantidad de ruido, y el acuerdo entre humanos es moderado. En este contexto, cabe preguntarse una cota a la performance que puede lograr un clasificador para esta tarea, que claramente por la misma naturaleza del problema, va a distar mucho de la perfección. Para obtener algunas medidas de esto, calculamos en primer lugar las F1 usando todos los posibles pares de anotadores.

Como la F1 es simétrica no necesitamos hacer ninguna asunción sobre sus roles.

¿qué?  
cuáles roles?

Algo a tener en cuenta es que la métrica final será contra la etiqueta resultante de la votación mayoritaria (nuestro *gold standard*). Una cota que seguro está por arriba de nuestra performance es el acuerdo que haya entre los anotadores y este *gold standard*; hay que también observar que cada etiqueta asignada (ver sección ??) codifica información de cada anotador, con lo cual este número es una cota superior pero puede que sobreestimada.

La tabla 6.2 contiene estadísticos para estas cotas, tanto entre anotadores como contra el *gold-standard*. Como podemos observar, la mediana entre anotadores de la F1 (usada para obviar outliers) es relativamente baja para la detección de odio ( $\sim 0.67$ ), mientras que contra el *gold standard* es de 0.85. De esto entendemos que la performance una cota superior a la performance está entre esos dos números.

## 6.4. Resultados

La tabla 6.3 contiene los resultados de la tarea de clasificación binaria, medidos por accuracy, precision, recall, F1 de la clase positiva y Macro F1 entre las dos clases, expresados como las medias de 10 corridas independientes de los experimentos. Las seis columnas corresponden a la combinación de los 3 posibles modelos dependiendo del contexto utilizado y de acuerdo a si ajustamos al dominio o no. Podemos observar que, en todos los casos, la adaptación de dominio (las columnas marcadas con FT) obtiene mejor performance que los modelos que no están adaptados ( $\neg$ FT), resultando en los casos sin contexto y con contexto de tweet en una mejora de alrededor

<sup>2</sup> Utilizamos algunos datos extra recolectados a posteriori de lo mencionado en el capítulo anterior

prácticamente

Me costó mucho entender estos párrafos y no me convence las explicaciones. Seguro podés mejorarlos mucho.

\Large?

Métrica	Sin Contexto		Tweect		Tweect + Cuerpo	
	¬FT	FT	¬FT	FT	¬FT	FT
Calls F1	0.646	0.651	0.638	<b>0.685</b>	0.653	0.680
Women F1	0.373	0.389	0.411	<b>0.421</b>	0.381	<b>0.421</b>
Lgbti F1	0.351	0.366	0.451	<b>0.482</b>	0.427	0.445
Racism F1	0.635	0.653	0.688	<b>0.720</b>	0.691	0.711
Class F1	0.401	0.433	0.491	<b>0.511</b>	0.451	0.476
Politics F1	0.555	0.611	0.579	0.625	0.591	<b>0.648</b>
Disabled F1	0.551	0.582	0.585	<b>0.609</b>	0.557	0.578
Appearance F1	0.726	0.742	0.741	<b>0.766</b>	0.755	0.758
Criminal F1	0.513	0.529	0.650	<b>0.699</b>	0.654	0.668
Macro F1	0.528	0.551	0.582	<b>0.613</b>	0.573	0.598
Macro Precision	0.558	0.630	0.642	<b>0.702</b>	0.677	0.678
Macro Recall	0.506	0.499	0.540	<b>0.551</b>	0.504	0.541

Tab. 6.4: Performance de los modelos para la tarea de detección granular de discurso de odio. Cada modelo es un BERT con 3 posibles entradas: sólo el comentario (*Sin contexto*), el tweet de la noticia a la cual responde el comentario (*Tweet*), y el tweet más el cuerpo de la noticia (*Tweet + Cuerpo*). Para cada una de estas posibilidades usamos dos versiones: una sobre BERT(¬FT) y otra sobre BERT ajustado al dominio (FT) de acuerdo a lo descripto en la sección 6.3.

de 4 puntos de F1. Entre los modelos sin ajustar a dominio, el modelo que consume el contexto completo (tweet + cuerpo de la noticia) obtiene el mejor desempeño; sin embargo, el contexto simple mejora esta performance cuando es adaptado. Viendo sólo las columnas adaptadas a dominio (FT), la mejora contra el modelo que no consume contexto es de 4.2 puntos de F1. El modelo con el contexto completo, si bien mejora la performance general contra no tener contexto, pierde precisión al ser adaptado al dominio.

La tabla 6.4 muestra los resultados de los experimentos de clasificación para la tarea de detección fina, medida en F1 para cada una de las características, y las medidas agrupadas de forma macro precision, recall, y F1. Se muestra en cada caso el resultado de la media de 10 corridas del experimento. Como era esperable, la ganancia de tener contexto disponible es más evidente en esta tarea, con una diferencia de Macro F1 aproximadamente 6 puntos de F1 entre la mejor versión sin contexto y la mejor versión con contexto (0.55 Macro F1 de la versión FT sin contexto vs 0.61 F1 de la versión FT con el contexto del tweet).

La figura 6.4 muestra los resultados de las F1 por característica, esta vez ordenados de mayor a menor según el gap entre la performance contextualizada vs la no contextualizada, junto a sus intervalos de confianza 95 %, usando como modelos las versiones ajustadas al dominio. Todas las características obtienen una mejora estadísticamente significativa al correr un test Mann-Whitney U ( $p \leq 0,005$ , p valores

la brecha

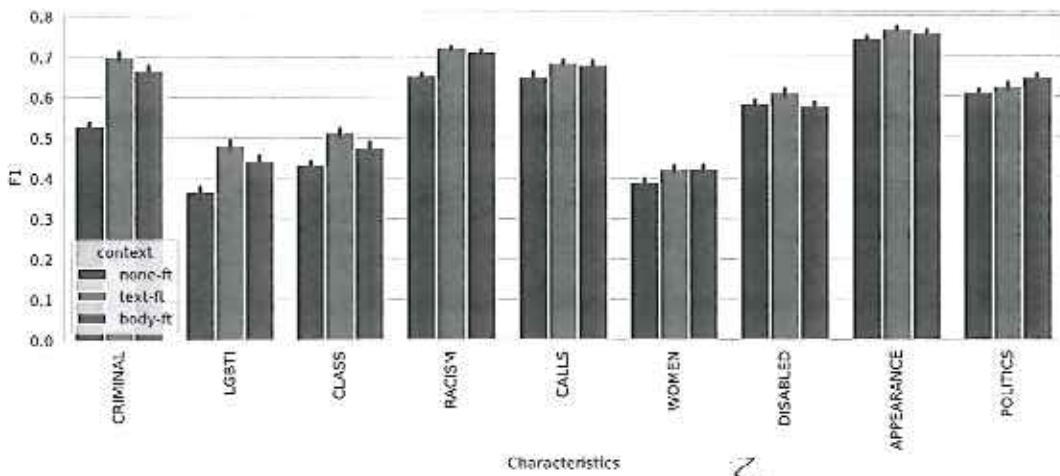


Fig. 6.4: Métrica F1 para cada característica de la tarea Task B. Las características están ordenadas de mayor a menor de acuerdo a la diferencia de performance entre el modelo sin contexto y el modelo contextualizado.

ajustados por Benjamini-Hochberg [14]). Las diferencias más sustanciales se dan en el caso de CRIMINAL (+0,17 F1 de diferencia), LGTBI (+0,12), CLASE (+0,08), y RACISMO (casi +0,07). Del otro lado, las características con menos mejora son APARIENCIA y POLITICA, algo esperable dado que el fenómeno tiene características poco dependientes del contexto (ver Apéndice C.3 para ejemplos). Finalmente, y como resumen de estas tablas, se observa que los modelos con contexto simple (que consumen sólo el tweet) son los que mejor performance tienen, en general y para cada característica, con la excepción de POLITICA.

La tabla 6.5 muestra la precisión y recall/sensibilidad de los clasificadores de la tarea granular. La sensibilidad es analizada de dos maneras: exacta, donde consideramos recuperado un tweet sólo si el clasificador acierta a la característica analizada (es decir, si la característica es MUJER, tiene que decir MUJER); y total, donde consideramos un tweet recuperado si alguna característica se enciende, independientemente si es la adecuada. Podemos ver que la categoría MUJER pasa de ser la de menor sensibilidad a dejar a la categoría LGTBI como la que tiene menor cantidad de recuperados. Análogamente la categoría CLASE obtiene una mejora sustancial en su sensibilidad, otra que es esperable que tenga cierta mezcla con otras características como RACISMO y posiblemente CRIMINAL. También puede observarse que en líneas generales, el contexto favorece una mejora de la precisión para cada característica, y también un mayor recall exacto. Si relajamos las condiciones de la detección, la sensibilidad de los clasificadores se empareja entre las versiones contextualizadas y no contextualizadas por cada categoría, con la excepción más notoria de LGTBI y CRIMINAL.

Como se mencionó en la sección 6.2, un clasificador sobre la tarea granular puede convertirse fácilmente en un clasificador para la tarea binaria tomando la disyunción lógica de sus salidas: si al menos una salida es 1, entonces el tweet es discurso de odio. Analizamos así el desempeño en la tarea binaria de aquellos clasificadores

Métrica	Sin Contexto		Tweet		Tweet + Cuerpo	
	Plano	Granular	Plano	Granular	Plano	Granular
precision	0,718	0,711	0,748	0,759	0,728	0,740
recall	0,602	0,636	0,653	0,667	0,641	0,660
F1	0,655	0,671	0,697	0,710	0,681	0,697
Macro F1	0,798	0,806	0,822	0,830	0,813	0,822

Tab. 6.5: Performance de los modelos para la tarea de detección de discurso de odio. Los modelos son BERT ajustados a dominio, consumiendo los 3 tipos de contexto: sin contexto, con contexto de tweet, y con contexto completo (tweet+cuerpo). Comparamos la performance entre los clasificadores que fueron entrenados sobre la tarea binaria y la tarea granular.

entrenados granularmente. La tabla 6.5 muestra esta comparativa. Podemos observar que en todos los casos, entregar el modelo sobre la tarea granular produce mejoras pequeñas pero significativas en la performance de los clasificadores entrenados sobre la tarea binaria.

## 6.5. Análisis de error

Analizaremos el error de los clasificadores propuestos de tres formas distintas: comparando las predicciones del mejor clasificador en términos de performance contra las etiquetas de nuestro dataset; comparando las predicciones del mejor clasificador contextualizado contra el mejor clasificador no contextualizado; y finalmente, comparando las predicciones entre los clasificadores entrenados sobre la tarea granular y los entrenados sobre la tarea fina.

Para analizar los errores de nuestros clasificadores, elegimos el que mejor performance tuvo en la tarea B: el clasificador que toma el tweet más el tweet de la noticia, ajustado a dominio. De una manera similar a la que realizamos en la sección 4.6.1, entrenamos 10 clasificadores y analizamos el error sobre el ensamble de voto mayoritario para centrarnos en los errores más frecuentes. Para ver los casos más problemáticos, analizamos aquellas clases donde peor desempeño muestran los clasificadores: MUJER, LGTBI, y GLASE. Observando las matrices de confusión, vemos que los resultados de los clasificadores son bajos en términos de recall. Observaremos entonces los casos de falsos negativos, que revisten el mayor problema.

La tabla 6.6 muestra una selección de comentarios falsos negativos para la característica LGTBI junto a sus contextos. De estas instancias, y observando también aquellos casos donde sí puede detectar el discurso discriminatorio contra LGTBI, podemos esbozar algunas posibles razones. En primer lugar, ciertos mensajes altamente ofensivos son complejos de entender o con una alta variabilidad léxica: por ejemplo, los que tratan de “enfermos” o mencionan cuestiones de la genitalidad (ejemplos 5 y 6). Esto puede hacerse de numerosas e “ingeniosas” maneras, y haciendo alusiones que difícilmente pueda un modelo de lenguaje entender (pensar todas las maneras que se puede hacer referencia a la genitalidad de una mujer trans, como las de los ejemplos 10 y 12).

Otros contextos –como los ejemplos 3, 7, 9, 10 y 11– omiten información acerca de

Mandar  
esto a un  
apéndice

A mí me  
gusta  
cómo  
queda así.

	Contexto	Comentario
1	Contó que era lesbiana, su papá le confesó que era gay y ahora su madre se enamoró de una mujer, lo que inspiró su segundo film	WTF. Mucho ESI los degeneró...
2		@usuario Esta familia tiene los genes alterados
3	Oscar González Oro ya está instalado en el Uruguay: Recuperé mi libertad"	Ahora quedate allá, y hablá mal d Macri d nuevo pa tener rating. Y opiná q los taxi boy uruguayos son mas educados q los escorts argentinos! Ano abierto!
4	¿Por qué un beso entre dos hombres los vuelve tan violentos?: la vida después de haber sido víctima de ataques homofóbicos	Será xq va contra la naturaleza de la raza...
5	"¿Por qué no vemos médicos trans?": el reclamo de un prestigioso cardiólogo para que América sea más inclusiva	Es difícil ser médico con la cabeza quemada
6		porque un enfermo no cura a otro enfermo
7	"Te amo". La emotiva dedicatoria de Luis Novaresio a su pareja en su cumpleaños	😂😂😂
8	Elizabeth Gómez Alcorta: "Por la pandemia, vamos a tener una suba de los femicidios y travesticidios"	Travesticidios... Osea asesinatos de tipos con peluca y tetas
9	Mariana Genesio Peña pasa su cuarentena total con guantes, barbijo y desnuda: "Mi cuerpo es el planeta Tierra"	Coronavirus nivel pelotudo en bolas
10		Con 3 piernas cualquiera es feliz!!!
11		pasa la cuarentena rascándose las bolas
12	Tras una ráfaga de más de 20 disparos asesinaron a una mujer trans en Rosario	Cómo no saco su escopeta y aplicó la defensa propia?! @usuario Salió de caño... cuac!
13		

Tab. 6.6: Falsos negativos para la característica LGBTI. Ninguno de los 10 clasificadores que consumen contexto y texto indentificaron como discriminatorios a estos comentarios

la sexualidad o género sobre quienes versa la nota. Esta información faltante no permite a los clasificadores (y tampoco a un humano que carezca de esta información) entender completamente el carácter discriminatorio de los mensajes.

En el caso de la clase MUJER, podemos observar que de nuevo el principal problema es un bajo recall. Sin embargo, observando que hay muchos casos que pueden mezclarse con otras características (como APARIENCIA) buscamos aquellos ejemplos que no son detectados por otras características. La tabla 6.7 muestra una selección de estos falsos negativos. Algunos ejemplos están muy en el borde de ser simplemente ofensivos, o algunos mensajes irónicos complejos de descifrar (que las feministas celebren por la muerte de una mujer trans, o hablar de chocar mujeres en un curso de manejo). Esto es esperable dado el acuerdo relativamente bajo para la categoría MUJER en el etiquetado de este dataset reportado en la tabla 5.8.

En la tabla 6.8 podemos observar una selección de ejemplos donde el clasificador contextualizado acierta y el no contextualizado se equivoca en su predicción, separados por falsos negativos (el modelo no contextualizado no logra detectarlos) y falsos positivos (el modelo no contextualizado predice equivocadamente que son discriminatorios). En el caso de LGBTI, la información contextual permite desambi-

siendo  
podemos  
observar esto,  
y por qué  
de nuevo?

Contexto	Comentario
1 Martha Rosenberg: "En situación de pandemia, legalizar el aborto es más urgente que nunca"	Quien es esta vieja?. No debería estar tejiendo?
2 Mara Gómez: la historia de la primera futbolista trans en el torneo argentino	Feminismo pierde de nuevo... ya le metieron un tipo... jaja punto para el patriarcado...
3 Tras una ráfaga de más de 20 disparos asesinaron a una mujer trans en Rosario	Las feministas en modo error 404 al no saber si celebrar o ofenderse
4 El desesperado pedido de Actrices Argentinas ante la violencia de género en cuarentena: "Es urgente"	Que risa me dan las feministas!!! Ignorantes.
5 Leche de cucaracha, la nueva bebida nutritiva: ¿quién se anima a probarla?	No me digas q la hija de CFK está embarazada y ya sale leche por esos senos
6 Los fans de Florencia Kirchner le piden casamiento por Instagram	Zoofilia
7	Hdp tienen que tener estomago para querer casarse con terrible adhesio
8 Rosario: para sacar una licencia de conducir habrá que hacer un curso de perspectiva de género	Te quieren adoctrinar desde cualquier ámbito, y se están metiendo en todo para que empieces a hablar como el orto, como a ellas les gusta.
9	El que choca más feministas le dan más años de licencia
10 Por qué los países liderados por mujeres parecen haber respondido mejor a la crisis del coronavirus	Son mujeres inteligentes que se dejan asesorar de sus esposos
11 Joe Biden presentó su nuevo equipo de comunicación compuesto enteramente por mujeres	Será equipo conche seque?

Tab. 6.7: Falsos negativos para la característica MUJER. Ninguno de los 10 clasificadores que consumen contexto y texto (ajustados a dominio) lograron identificar como discriminatorios a estos comentarios

guar casos como los 7 y 8, muy similares pero con un contexto sumamente distinto. Un problema que se puede apreciar es que el clasificador no contextualizado –al ser entrenado con datos contextualizados– aprende correlaciones espurias como que las celebraciones son discriminatorias (ejemplo 16).

La comparativa entre los clasificadores entrenados sobre la tarea binaria y la tarea granular es más difícil de interpretar. Si bien se observa que entre los falsos negativos del clasificador binario se encuentra una proporción más alta de tweets racistas, es difícil elucidar una posible razón detrás de esto. En la tabla D.1 en el apéndice D se encuentra una tabla con una muestra de ejemplos en los cuales el clasificador granular acertó y el binario falló.

## 6.6. Discusión

Para analizar el impacto del contexto en la detección de discurso de odio, planteamos dos tareas de clasificación sobre el dataset construido en el capítulo 5; la tarea de detección binaria, donde respondemos si un comentario contiene discurso

✓ sin filde

CRIMINAL			
	Tipo	Contexto	Comentario
1	FN	Una policía baleó y mató a un joven de 17 años que la atacó con una tijera en Moreno url	Excelente!
2		El polémico cortejo del ladrón asesinado por el jubilado en Quilmes	Le hubieran puesto una bomba al cortejo
3	FP	Ivana Nadal se cansó de la criticaron y sorprendió con su respuesta: "Gracias a Dios a fin de año me voy del país"	Esa si es una buena noticia.
4		¿Se va del país? Juana Viale estaría tramitando la ciudadanía uruguaya	Q bueno? Una mierda menos. Q se quede alla
LGBTI			
6	FN	"Te amo". La emotiva dedicatoria de Luis Novaresio a su pareja en su cumpleaños	Definitivamente no acepto esta degeneración repugnante de la humanidad.
7		Mara Gómez: la historia de la primera futbolista trans en el torneo argentino	Sigue siendo HOMBRE, que por GENÉTICA, no por una ideología u orientación sexual, GE-NÉ-TI-CA, es más fuerte que la mujer.
8	FP	URGENTE: Un hombre se incrustó con su auto en la puerta de la Embajada de China y aseguró que tenía explosivos	No es hombre. Es un boludo
9		Detuvieron al hombre que admitió violar a su hija en audios de WhatsApp y fue tendencia en redes: un tío lo entregó	Degenerado
CLASE o RACISMO			
10	FN	Las organizaciones sociales salieron al cruce de la acusación de Sergio Berni por la toma de tierras: "La falta de vivienda no se resuelve con balas"	Si, los podemos echar matando a todos y listo
11		La temporada de verano en la Costa Atlántica comenzó con un corte total en la Ruta 2: organizaciones sociales piden canastas navideñas	Hay que desparasitar urgente el país.
12		La pregunta billionaria: ¿China debería pagar el costo de la pandemia?	Si obviamente y desaparecer de la faz de la tierra. Mira el quilombo que armó. Se nos están muriendo todos...
13	FP	Javier Milei confirmó que va "a militar en política" junto a José Luis Espert para que "en 35 años la Argentina sea primera potencia mundial"	Otro parásito
14		Martha Rosenberg: "En situación de pandemia, legalizar el aborto es más urgente que nunca"	La verdad que sí... así se dejan de reproducir!!!
15		Confusión por la foto de Alberto Fernández con activistas veganos en medio de las negociaciones con China por el acuerdo porcino	Quilombo en puerta con China, son un desastre
16	?	La aberrante arenga machista que dio el preparador físico de Güemes de Santiago del Estero	Un capo.

Tab. 6.8: Ejemplos donde el clasificador contextualizado acierta y el no contextualizado falla. FN marca que el clasificador no contextualizado no detecta el comentario como discriminatorio (ni para la característica marcada ni para otras) mientras que el contextualizado sí lo hace; FP es al revés, que el clasificador no contextualizado marca erróneamente el comentario como discriminatorio

de odio; y una tarea de detección granular, donde se debe mencionar la o las características protegidas ofendidas (si agrede a las mujeres, al colectivo LGBTI, si es racista, etc). En ambas tareas planteadas propusimos clasificadores que consumen

3 tipos de entrada: el comentario sin contexto, el comentario con el contexto del tweet al que responde<sup>1</sup>, y el comentario más el tweet al que responde y también el texto del artículo. Podemos observar que el contexto parece dar una mejora moderada pero significativamente estadística en la tarea de detección del discurso de odio (alrededor de 3 puntos F1), y una mejora considerable en la tarea característica ofendida (alrededor de 6 puntos F1 medias).

Esto, en algún punto, indicaría que el contexto puede ser aprovechado para mejorar los algoritmos de detección de discurso de odio. Si bien este resultado podría estar en aparente contradicción con trabajos recientes que no encontraron ninguna mejora en el uso del contexto en la detección de toxicidad [109], se puede señalar que la detección del discurso de odio es una de las formas más complejas de comportamiento "tóxico" y, como tal, podría permitir que los clasificadores tengan más información para predecir si el texto dado es odioso o no. Otra razón detrás de este resultado es el dominio de nuestro conjunto de datos: mientras que Pavlopoulos et al. [109] usa el contexto conversacional, nosotros usamos el título y el cuerpo del artículo como contexto para los comentarios de los usuarios. Más recientemente, y como marcamos en la sección de trabajo previo, en Xenos et al. [150] han observado que, restringiendo el análisis a un subconjunto de comentarios sensibles al contexto (ver 6.1 y 5.1 para más información), la performance de los algoritmos de detección de toxicidad mejoran de manera significativa.

La utilización de un contexto más largo como el artículo de la noticia no mejora la performance de los clasificadores en nuestros experimentos. Hay varias interpretaciones posibles de esto: en primer lugar, los humanos suelen contestar sin leer el artículo, con lo cual este resultado parecería ser coherente con esta observación. Por otro lado, los humanos solemos tener acceso a un contexto mucho más rico, muchas veces equivalente a haber leído la nota, algo que nuestros clasificadores carecerían.

Podría también esto ser atribuido al modelo pre-entrenado que usamos para codificar esto (BETO, la variante en español de BERT) suelte estar pre-entrenada para textos más cortos. Teniendo esto en cuenta, realizamos el ajuste de dominio usando el contexto largo, pero aún así la performance del clasificador que consume este contexto largo se mantuvo por debajo del que usaba el contexto simple.

El análisis del error realizado demostró que, si bien el contexto parecería mejorar la detección de discurso de odio, para muchas características protegidas se mantiene aún como una tarea difícil. Un caso ejemplificador de esto es la discriminación contra el colectivo LGBTI. En las instancias del dataset –y en muchos de los ejemplos en los que el algoritmo de detección falla– puede verse que las agresiones contra este colectivo y sus miembros son sumamente sofisticadas, lejos de las agresiones meramente basadas en insultos u otras palabras ofensivas. Nuestros clasificadores, aún en sus mejores versiones (usando ajuste de dominio y contexto) obtuvieron una baja performance en la detección de este fenómeno (alrededor de 0,5 F1 score) dando cuenta que es un problema no trivial y merece ser analizado más detenidamente debido a la complejidad de estos mensajes, que suelen reunir ironía, metáforas, y otros artilugios que hacen difícil su detección.

En el caso de la categoría mujer, inesperadamente, también obtuvimos una performance muy baja en la detección de agresiones misóginas. En el análisis de error, podemos observar que tenemos casos complejos de descifrar que fueron marcados por

las Secciones

Revisar  
formatos

nuestros anotadores: por ejemplo, ataques velados a mujeres víctimas de violación (llamarlas mentirosas).

Algo que debe ser tenido en cuenta para matizar estos resultados es que utilizamos un amplio espectro de características protegidas. Incluso, la que más se beneficia del contexto es una que introdujimos ad-hoc para este experimento (discurso de odio contra criminales). En contrapunto, otras características "no convencionales" son poco beneficiadas por el contexto (como discurso de odio en base a la apariencia, opinión política y discapacidad).

Un resultado que también observamos es que pareciera ser que nuestros clasificadores mejoran leve pero significativamente su performance en un contexto de aprender cada característica por separado en vez de sólo aprender a distinguir la etiqueta binaria de discurso de odio. Si bien la mejora es marginal (cerca de un punto de F1) y no es observable de manera subjetiva mediante un análisis de error, una posible razón detrás de esto es que la señal más precisa acerca de la categoría ofendida puede ayudar a distinguir mejor las fronteras de este fenómeno. Aún cuando esta mejora no fuese tal, poder tener una salida más interpretable y granular es mejor que simplemente obtener una predicción binaria.

Una limitación importante de este estudio es que el entrenamiento lo realizamos sobre datos ~~entrenados~~ únicamente observando el contexto. Entrenar a los clasificadores no contextualizados sobre estas etiquetas induce a los clasificadores a tomar decisiones erróneas, como asumir que celebraciones son discurso de odio debido a instancias que –con contexto– tienen esa naturaleza. Un estudio más completo del impacto del contexto en esta tarea debería incluir los datos entrenados sin contexto.

En el terreno de la aplicación, un problema práctico de este resultado es que no siempre tenemos un contexto disponible para un texto dado. Incluso si podemos encontrarlo, a veces este contexto puede no ser en forma de artículo de noticias, sino como un hilo de conversación o incluso de alguna otra representación. Teniendo en cuenta alguna de las consideraciones hechas en esta discusión, una posible línea de investigación sería la de incorporar distintos tipos de contexto, desde más mensajes en el hilo de la conversación, conocimiento estructurado (por ejemplo, la propuesta en *ERNIE* [157] o *KI-BERT* [39]) o bien una combinación de diversas fuentes, incluso multimodales.

## 6.7. Conclusiones

En este capítulo hemos realizado experimentos de clasificación sobre el dataset construido en el capítulo anterior, focalizándonos en analizar el impacto de la utilización del contexto en la performance de los modelos. Planteamos dos tareas: una tarea binaria –detectar si existe discurso de odio o no– y una tarea granular –definir qué categorías son atacadas en un tweet, si es que las hay . Para ambas tareas, los modelos contextualizados obtienen mejoras significativas en la performance, dando indicios de que información adicional al comentario analizado puede ayudar a detectar el discurso de odio. Si bien en nuestros experimentos el contexto más pequeño (el tweet del artículo de la noticia) fue el que mejor resultados obtuvo, una línea de trabajo futuro podría explorar otras formas de incorporar el contexto más largo –en este caso, el artículo de la noticia.

Así mismo, observamos una pequeña pero estadísticamente significativa mejora en la detección de discurso de odio al entrenar un clasificador granular al ser evaluado de manera binaria. En este caso, obtenemos una ventaja al obtener una salida más interpretable de las características ofendidas, y que además que no sólo no empeora la performance de nuestro clasificador sino que hasta incluso mejora levemente.

Del análisis de error, se observa que algunas categorías del discurso de odio se muestran elusivas para los algoritmos de detección. Uno de estos casos son los mensajes abusivos contra la comunidad LGB'TI+, conteniendo mensajes semánticamente complejos, implícitos y con metáforas que son esquivas para los modelos propuestos. A pesar de estas limitaciones, esta característica fue una de las más beneficiadas por la adición de contexto, aunque su desempeño sigue siendo bajo, teniendo una puntuación de F1 de alrededor de 0.5.

Podemos concluir que los datasets de discurso de odio deberían –en la medida de lo posible– contener **información contextual** sobre los comentarios analizados. Esta información puede darse en forma de artículos de noticias, como un hilo de conversación, o incluso como otras formas –por ejemplo, como una base de conocimiento. Sobre esto, trabajo futuro debería explorar el impacto de utilizar esta información adicional para integrarla en algoritmos de detección de discurso de odio. La evidencia de estos experimentos –por ahora preliminares, y con las limitaciones marcadas en la discusión– indica que los modelos del estado del arte pueden utilizar esta información para mejorar la detección de discurso de odio en redes sociales. En segundo lugar, los datasets de discurso de odio deberían incluir **información granular** acerca de las características atacadas –y no sólo una etiqueta binaria– ya que por un lado esto mejora la interpretabilidad de los algoritmos de detección, y resultados preliminares de este estudio indicarían que mejoran marginalmente la performance en la detección en general.

Finalmente, un aspecto que introdujimos en este capítulo fue el de adaptar un modelo de lenguaje pre-entrenado a su dominio, siendo en nuestro caso los comentarios sobre notas periodísticas en Twitter. Las mejoras que reportó la utilización de estas técnicas fue significativa, en consonancia con otros trabajos recientes. Pasaremos a continuación a estudiar estas técnicas en el marco más general de la clasificación de textos sociales.

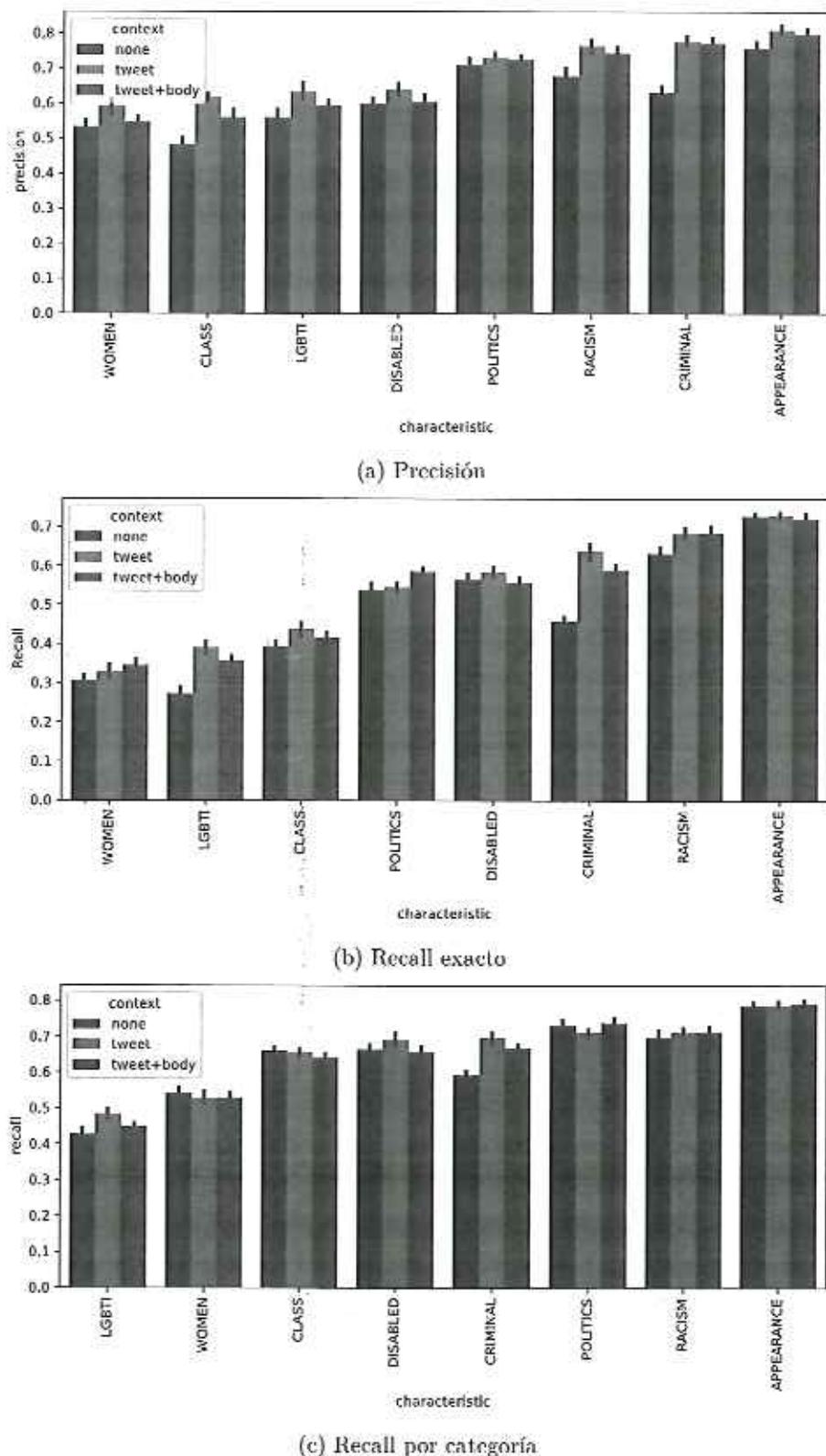


Fig. 6.5: Precision y Recall para cada característica de la tarea granular. Las diferentes barras marcan el tipo de contexto que recibe el clasificador. Recall exacto (Figura 6.5b) cuenta el recall sobre la salida exacta de cada categoría. Recall total cuenta como recuperado un tweet si al menos alguna característica del clasificador lo marca como discurso de odio.

Este figura  
que es muy lego  
de la tabla 6.5  
y ~~quiero~~ solo  
mencionarla.  
24

## 7. ADAPTACIÓN DE DOMINIO

En este capítulo exploraremos cómo mejorar la detección de discurso de odio desde una perspectiva más general, mediante técnicas de adaptación de dominio y construyendo modelos de lenguaje pre-entrenados en textos sociales. Hemos visto en capítulos anteriores que las técnicas de representación utilizadas en los últimos años (desde los word-embeddings hasta los modelos de lenguajes pre-entrenados) generan, por un lado, representaciones ricas al ser entrenadas en dominios sociales. A su vez, trabajos recientes muestran que modelos pre-entrenados mejoran su performance continuando el aprendizaje semi-supervisado sobre un conjunto de datos similar al de la tarea objetivo, en nuestro caso textos generados por usuarios.

Los modelos pre-entrenados de lenguaje suelen ser entrenados sobre cierto conjunto de datos que se suponen suficientemente generales, como Wikipedia que comprende textos de carácter enciclopédico, o como Common Crawl, que es una recopilación de datos de distintos sitios web<sup>1</sup>. El uso del lenguaje en estas fuentes suele tener cierta discordancia con el de muchas tareas; por ejemplo, tareas en textos médicos, textos científicos, o –lo que es de interés en esta tesis– las tareas sobre textos sociales. El texto generado por usuarios tiene un uso del lenguaje muy particular, con mucha jerga, expresiones particulares, y otros usos que se diferencian de los textos fuente en los que están entrenados BERT, RoBERTa, *BETO* y otros. A cada uno de estos grupos de textos con cierta relación se los denomina de manera poco concisa dominios, y al conjunto de técnicas utilizadas para adaptar distintos modelos a estos dominios se lo denomina adaptación de dominio [36].

Continuaremos aquí alguna de las ideas utilizadas en el capítulo 6 sobre adaptación de dominio. En primer lugar, describiremos el proceso de entrenamiento desde cero de un modelo de lenguaje pre-entrenado basado en transformers (RoBERTa) [84] sobre tweets, al cual llamamos *RoBERTuito*. En segundo lugar, probaremos tomar un modelo *BETO* y continuar el pre-entrenamiento sobre un gran conjunto de tweets. Trabajo previo ha demostrado que ambas estrategias mejoran la performance de los modelos de clasificación del estado del arte cuando trabajamos en dominios especializados; sin embargo, ninguna comparación de la que tengamos conocimiento se ha realizado entre estas dos formas de abordar el problema<sup>2</sup>. Este punto es de interés dado el gran costo que tiene entrenar modelos basados en transformers. Para ello, utilizamos como benchmark de este análisis las tareas que hemos tratado en esta tesis, introducidas en los capítulos 3, 4 y 6.

Comenzamos en la sección 7.1 haciendo un recuento de las técnicas de adaptación de dominio y modelos pre-entrenados sobre distintos dominios. En la sección 7.2 describimos la construcción y el entrenamiento de *RoBERTuito* [114], y en la sección 7.3 describimos la adaptación de dominio de *BETO* al dominio social y comparamos la performance de estos modelos contra *RoBERTuito*.

<sup>1</sup> [<https://commoncrawl.org/>]

<sup>2</sup> O al menos seguro no se hizo en español

, al menos para el idioma español

### 7.1. Trabajo previo

La definición de qué es un dominio suele ser relativamente amplia, pero podemos considerarlo como un conjunto de textos que guardan cierta similaridad respecto al tópico o género: por ejemplo, artículos de noticias, novelas, discursos políticos, posts de redes sociales, entre otros [58].

Goodfellow et al. [53] define la adaptación de dominio como una situación similar a la de Transfer Learning: dado un modelo que fue entrenado sobre una distribución de datos o dominio  $P_1$ , queremos utilizarlo sobre una distribución  $P_2$  relativamente similar. Concretamente, nos referimos a la aplicación de alguna técnica que ajuste la distribución de la entrada (de  $P_1$ ) a la distribución de nuestro nuevo dominio (la distribución  $P_2$ ), bajo la asunción de que ambas distribuciones son relativamente similares. Glorot et al. [51] es uno de los primeros trabajos que aplica esta técnica sobre modelos basados en Deep Learning en NLP<sup>3</sup>, usando stacked denoising auto-encoders para aprender características no supervisadas de los textos y usar técnicas de adaptación para los distintos dominios analizados. Eisenstein [36] trata puntualmente la adaptación de dominio para textos sociales, que se podría resumir en “adaptar las herramientas al texto” (social). Contrapuesto parcialmente está el concepto de *normalización* de la entrada, que sería intentar “adaptar el texto a las herramientas”.

Para lo que nos concierne en las técnicas actuales de modelos pre-entrenados (descriptos en 2.5) solemos querer ajustar un modelo de lenguaje a un dominio distinto al que fue utilizado en su pre-entrenamiento. Por ejemplo, un modelo *BERT* pre-entrenado en textos formales (como Wikipedia o noticias) queremos ajustarlo a la distribución de textos sociales, que si bien ambas mantienen el idioma (inglés o español) suelen tener distribuciones notoriamente distintas. Se ha observado que los modelos de lenguaje basados en transformers son mucho más robustos frente a los cambios de dominio [63], pero todavía siguen sufriendo cuando los datos analizados difieren fuertemente de los utilizados en el entrenamiento. Ruder [127] hace un repaso extenso de los últimos avances en las técnicas de adaptación de dominio utilizando modelos de lenguaje del estado del arte.

Dentro de la última ola que sacudió NLP de modelos pre-entrenados, la técnica de ULMFit [67] contempla una etapa de adaptación a la tarea utilizando de manera no-supervisada el texto del dataset de la tarea atacada corriendo la tarea de modelado de lenguaje. Los modelos basados en Transformers como BERT [35] y subsiguientes no siguieron esta costumbre, sólo haciendo la adaptación de los pesos sobre las etiquetas supervisadas, pero Gururangan et al. [58] han mostrado recientemente que esta adaptación no supervisada es beneficiosa para los modelos.

Concretamente, dentro de las tareas de extracción de opiniones en redes sociales –lo que nos concierne en esta tesis– la adaptación de dominio resulta importante ya que expresiones en distintos ámbitos pueden tener sentidos distintos: citando un ejemplo de Pang and Lee [106], decir “léé el libro” en una reseña de un libro Amazon puede ser algo positivo, mientras que en el comentario de una película puede ser considerado negativo.

Gururangan et al. [58] analizan el impacto de los ajustes de dominio para los

pero no sólo eso, no? también  
el medio, el estilo de escritura, y  
los autores a los que se dirige.

<sup>3</sup> 2011, incluso antes de word2vec

Nombre	Idioma	Dominio	Familia
SciBERT[9]	inglés	Papers	BERT
ClinicalBERT [69]	inglés	noticias médicas	BERT
BERT-CT [101]	inglés	Tweets en inglés sobre COVID	BERT
BERTweet [102]	inglés	Tweets, algunos COVID-related	RoBERTa
ALBERTo [120]	italiano	Tweets	RoBERTa
TwiBERT	español	Tweets	~ BERT

Tab. 7.1: Modelos pre-entrenados sobre dominios no canónicos. En familia nos referimos a qué tipo de pre-entrenamiento es realizado en el modelo de lenguaje: BERT es MLM + NSP, RoBERTa sólo MLM. En el caso de TwiBERT, se usa un símil BERT ya que no usan exactamente NSP sino una tarea muy parecida.)

modelos de lenguaje basados en transformers. Para ello, consideran varios dominios como ser biomédico, reviews de películas, papers de cs. de la computación (CS), y noticias. Plantearán dos configuraciones de adaptación de dominio:

- Domain Adaptation: ajustar el modelo de lenguaje sobre un extenso conjunto de datos no etiquetado, usualmente el “sobrante” del proceso de recolección que no es anotado
- Task Adaptation: ajustar el modelo de lenguaje sobre el dataset, de la misma manera que se hace en ULM-Fit

En todos los casos, usando modelos del estado del arte como RoBERTa muestran que aplicar conjuntamente lo que ellos consideran Domain Adaptation y Task Adaptation mejora la performance significativamente. La adaptación, dado que usan modelos como RoBERTa, consiste tan sólo en correr la tarea de MLM sobre los textos del dominio a adaptar.

Luego del estallido de los modelos de lenguaje basados en transformers, algunos trabajos se han basado en directamente entrenar estos modelos ya no en textos formales como Wikipedia o noticias sino directamente en el dominio en cuestión. Por ejemplo, SciBERT [9] es un modelo BERT entrenado directamente en textos científicos, BERTweet [102] entrena un modelo RoBERTa [84] sobre cerca de 850M tweets en inglés, una parte de ellos relacionados a la pandemia del COVID-19. La tabla 7.1 lista algunos de estos modelos. En español tenemos el modelo TwiBERT [52]; sin embargo, tiene algunas limitaciones: en primer lugar, no queda claro cuánto tiempo de entrenamiento recibió ni si los datos fueron suficientes; en segundo, usaron un modo de entrenamiento basado en una variante de la tarea NSP (ver subsección 2.5) cuando numerosos trabajos muestran que el tipo de entrenamiento basado en RoBERTa (sólo tarea MLM) mejora el desempeño en las tareas finales. Finalmente, su modelo no es accesible mediante el model hub de huggingface, limitando seriamente su acceso.

Algunas oportunidades de mejora de lo estudiado en Gururangan et al. [58] son, en primer lugar y siguiendo la regla de Bender [11], realizar el estudio en un idioma distinto del inglés. Por otro lado, un dominio que no está estudiado en dicho trabajo es el dominio de tareas en textos de redes sociales. Finalmente, es de interés realizar

en segundo lugar

una comparación de la performance de modelos adaptados al dominio contra aquellos que son entrenados desde cero en ese dominio, dado el enorme costo computacional, energético y ambiental que implica el entrenamiento de estos modelos desde cero [138].

## 7.2. Modelo pre-entrenado sobre tweets

*Constituye  
un énfasis*

Pasamos a continuación a describir a *RoBERTuito*, un modelo de lenguaje pre-entrenado sobre tweets en español. Entrenamos tres versiones de *RoBERTuito*: una versión *cased* que conserva las mayúsculas, una versión *uncased* que convierte todo a minúsculas y una versión *deacc*, que usa minúsculas y elimina los acentos en los tweets. El español normativo prescribe el uso de tildes en ciertos casos para señalar la acentuación de una palabra, algo que por lo general suele pasarse por alto en los textos sociales. Queremos analizar entonces si eliminarlos mejora el rendimiento de los modelos, algo que puede considerarse como una normalización levemente más fuerte que remover mayúsculas únicamente.

prescribe?

### 7.2.1. Recolección de tweets

A continuación describimos el proceso de recolección de tweets que utilizamos para entrenar *RoBERTuito*. El stream de API de acceso gratuito (también conocida como *Spritzer*) es una muestra de alrededor del 1% de los tweets, supuestamente aleatoria, aunque algunos estudios han mostrado algunas preocupaciones acerca de la posible manipulación de esta muestra [117]. Muestras no representativas y sesgadas pueden afectar al modelo en tareas finales, algunos de estos errores siendo potencialmente dañinos como sesgos raciales o de género. Por ello, publicamos el conjunto de datos para ser inspeccionado, y queda como trabajo futuro un análisis más detallado de sus instancias.

En primer lugar, descargamos una colección de Spritzer subida a Archive.org que data de Mayo de 2019<sup>4</sup>. Filtramos aquellos tweets cuya metadata indique que su idioma no sea español. Sobre los tweets en español, usamos la API de Twitter para descargar los tweets de los usuarios en cuestión. De este proceso recolectamos alrededor de 622 millones de tweets de cerca 432 mil usuarios.

Finalmente, considerando que no queremos entrenar , nos quedamos sólo con aquellos tweets que tengan 6 o más tokens, usando para esto el tokenizador entrenado en BERT [26], sin contar repeticiones de emojis y haciendo el preprocessado descripto en capítulos anteriores: reemplazamos los caracteres hasta un máximo de tres, convertimos los nombres de usuarios a un token especial `@usuario`, convertimos los emoji a una representación textual, y partimos los hashtags en lo posible (ver sección XXX). Esto lo realizamos a priori para bajar la carga de trabajo a la hora del pre-entrenamiento.

De este proceso quedan 500M tweets, los cuales ordenados en 1000 archivos para facilitar la lectura en procesos posteriores. El repositorio de la recolección de tweets puede encontrarse en <https://github.com/finiteautomata/spritzer-tweets>.

<sup>4</sup> <https://archive.org/details/archiveteam-twitter-stream-2019-05>

Hiperparámetro	Valor
#Heads	12
#Layers	12
Hidden Size	768
Intermediate Size	3072
Hidden activation	Gelu
Vocab. size	30,000
MLM probability	0.15
Max Seq length	128
Batch Size	4,096
Learning Rate	$3.5 * 10^{-4}$
Decay	0,1
$\beta_1$	0.9
$\beta_2$	0.98
$\epsilon$	$10^{-6}$
Warmup steps	36,000 (6 %)

Tab. 7.2: Hiperparámetros utilizados en el entrenamiento de *RoBERTuito*. Los valores de  $\beta$  y  $\epsilon$  refieren a los hiperparámetros de Adam

Entrenamos a los tokenizadores usando el algoritmo *SentencePiece* [78] en los tweets recopilados para cada una de las tres configuraciones. Guardamos 30K tokens para cada uno. Usamos *tokenizers* library<sup>5</sup> que proporciona implementaciones rápidas en el lenguaje de programación Rust para muchos algoritmos de tokenización.

### 7.2.2. Arquitectura y entrenamiento

Se utilizó una arquitectura RoBERTa base para *RoBERTuito*, con 12 capas de auto atención, 12 cabezas de atención y tamaño intermedio 768, de la misma manera que *BERTweet*. Entrenamos sobre la tarea de MLM, en la misma línea de RoBERTa y BERTweet, sin tener en cuenta la tarea de predicción de la siguiente oración usada en BERT u otras tareas de orden de tweets como la usada en Gonzalez et al. [52].

Teniendo en cuenta los hiperparámetros de RoBERTa y BERTweet, decidimos utilizar un tamaño de batch size grande para nuestro entrenamiento. Si bien se recomienda un tamaño de 8k en RoBERTa, debido a las limitaciones de recursos, decidimos equilibrar el número de actualizaciones utilizando un tamaño de lote de 4k.

Para comprobar la convergencia, primero entrenamos un modelo uncased para 200k pasos de optimización. Al comprobar que convergió (y obtuvo buenos resultados en las tareas del benchmark), procedimos al entrenamiento completo de los tres modelos. Entrenamos nuestros modelos durante aproximadamente tres semanas en una TPU v3-8 y una máquina pre-emptible *e2-standard-16* en GCP. Estas nos fueron provistas por el programa Google TPU Research Cloud. Nuestro código está basado en la biblioteca *huggingface's transformers*[? ] y su implementación de *RoBERTa*.

<sup>5</sup> <https://github.com/huggingface/tokenizers>

Model	Train loss	Eval loss	Eval ppl
cased	1.864	1.753	5.772
uncased	1.940	1.834	6.259
deacc	1.951	1.826	6.209

Tab. 7.3: Resultados del pre-entrenamiento para las tres versiones de *RoBERTuito*. La función de costo utilizada es la entropía cruzada de la tarea de MLM.

Cada tweet está tokenizado y enmascarado dinámicamente con una probabilidad igual a 0,15. La tabla 7.3 muestra los resultados del entrenamiento en términos de pérdida de entropía cruzada y perplejidad.

### 7.2.3. Evaluación

Para analizar la performance de este modelo, usamos un conjunto de tareas como benchmark. Las tareas elegidas son todas las que analizamos en este trabajo hasta el momento:

1. Análisis de sentimientos (capítulo 3)
2. Análisis de emociones (capítulo 3)
3. Detección de Ironía (capítulo 3)
4. Detección de discurso de odio (capítulo 4)
5. Detección contextualizada de discurso de odio (capítulo 6)

Para más detalles sobre los datasets y cuestiones puntuales de cada tarea, referimos a los capítulos correspondientes. Para cada una de estas tareas, comparamos la performance de *RoBERTuito* contra los siguientes modelos de lenguaje pre-entrenados:

- BERT [26], tanto en versión cased como uncased.
- *RoBERTa-BNE* [59], un modelo RoBERTa entrenado sobre una base de datos de 500GB de todos los sitios .cs
- *BERTin*<sup>6</sup>, otro modelo RoBERTa entrenado en el contexto de un evento de la comunidad Flax/Jax<sup>7</sup>, en el cual los autores exploraron diferentes estrategias de sampleo para entrenar este modelo en relativamente poco tiempo sobre la sección en español del corpus mc4, creado para entrenar T5 [123].

Para cada modelo y tarea corremos el experimento 10 veces y reportamos medias y desviaciones estándar. Utilizamos el mismo procedimiento de entrenamiento referidos en cada capítulo para hacer el fine-tuning discriminativo de cada modelo, sin efectuar ninguna optimización de los hiperparámetros.

<sup>6</sup> <https://huggingface.co/bertin-project/bertin-roberta-base-spanish>

<sup>7</sup> <https://discuss.huggingface.co/t/open-to-the-community-community-week-using-jax-flax-for-nlp-cv/7104>

Modelo	CONTEXT	HATE	SENTIMENT	EMOTION	IRONY	score
beto <sub>U</sub>	0,591 ± 0,006	0,757 ± 0,012	0,649 ± 0,005	0,521 ± 0,006	0,702 ± 0,008	0,644
bertin	0,557 ± 0,008	0,767 ± 0,005	0,665 ± 0,003	0,518 ± 0,012	0,716 ± 0,008	0,645
betoc	0,582 ± 0,007	0,768 ± 0,012	0,665 ± 0,004	0,521 ± 0,012	0,706 ± 0,007	0,649
roberta-bne	0,577 ± 0,004	0,766 ± 0,015	0,669 ± 0,006	0,533 ± 0,011	0,723 ± 0,017	0,654
robertuito <sub>C</sub>	0,590 ± 0,005	0,790 ± 0,012	0,701 ± 0,012	0,519 ± 0,032	0,719 ± 0,023	0,661
robertuito <sub>D</sub>	0,593 ± 0,006	0,798 ± 0,008	0,702 ± 0,004	0,543 ± 0,015	0,740 ± 0,006	0,675
robertuito <sub>G</sub>	0,593 ± 0,004	0,801 ± 0,010	0,707 ± 0,004	0,551 ± 0,011	0,736 ± 0,008	0,678

Tab. 7.4: Resultados de la evaluación de modelos pre-entrenados y modelos ajustados en dominio para el benchmark de tareas sociales: CONTEXT es contextualized hate speech, HATE es hate speech detection sobre el dataset de HatEval, SENTIMENT, EMOTION e IRONY son análisis de sentimiento, emociones e ironía sobre los corpus de TASS. El score es el promedio de todas las tareas.

#### 7.2.4. Resultados

La tabla 7.4 muestra los resultados de la evaluación de los modelos seleccionados para las cinco tareas de clasificación propuestas. Los resultados se muestran como la media de quince ejecuciones de los experimentos, junto con la desviación estándar, junto a un score promediando las puntuaciones de cada tarea de forma similar a lo hecho en GLUE [143]. Podemos observar que en la mayoría de los casos, todas las configuraciones de *RoBERTuito* funcionan significativamente por encima de otros modelos, en particular para las tareas de Discurso de odio y Sentimiento. El único caso donde esto no ocurre es en la tarea de detección de discurso de odio contextualizado, donde si bien hay una mejora, es marginal y no significativa.

Respecto a la comparación entre los tres modelos, analizamos las diferencias entre estos mediante una prueba de la diferencia de medias. Para ello, realizamos un test de Kruskal-Wallis [77] para las performances de cada tarea. Existen diferencias significativas entre el desempeño de los tres modelos de *RoBERTuito* ( $H(3) = 6,88, p < 0,05$  para Discurso de odio,  $H(3) = 9,90, p < 0,01$  para Análisis de sentimiento,  $H(3) = 11,85, p < 0,01$  para Análisis de emociones,  $H(3) = 11,54, p < 0,01$  para Detección de ironía), no así para la detección de odio contextualizado ( $H(3) = 3,59, p > 0,15$ ).

Para verificar las diferencias significativas entre las performances de los tres modelos para las 4 tareas mencionadas, realizamos un análisis post-hoc con un test de Dunn (con corrección de Benjamini-Hochberg). Exceptuando la tarea de análisis de sentimientos, la versión *cased* muestra siempre diferencias significativas contra las versiones *uncased* o *deacc*. Sin embargo, no se encuentran diferencias significativas entre las versiones *uncased* y *deacc*.

Mandar tablas a un apéndice

Este resultado puede leerse de dos maneras: primero, que una normalización más fuerte (remover las tildes) del texto de entrada en español no produce una mejora significativa en el rendimiento de los modelos; también, que mantener las tildes en el texto de entrada no es beneficioso ni perjudicial para el rendimiento del modelo.

Podrán ser más efectivas.

La figura 7.1 muestra la distribución del número de tokens en el texto de entrada agrupados por tarea. Podemos observar que los modelos de *RoBERTuito* tienen representaciones más compactas que *BETO* y *RoBERTa-BNE*; sin embargo, *berlin* parece tener representaciones iguales o más compactas que él, a pesar de su

Por qué es importante ver esto?

¿Qué es él?

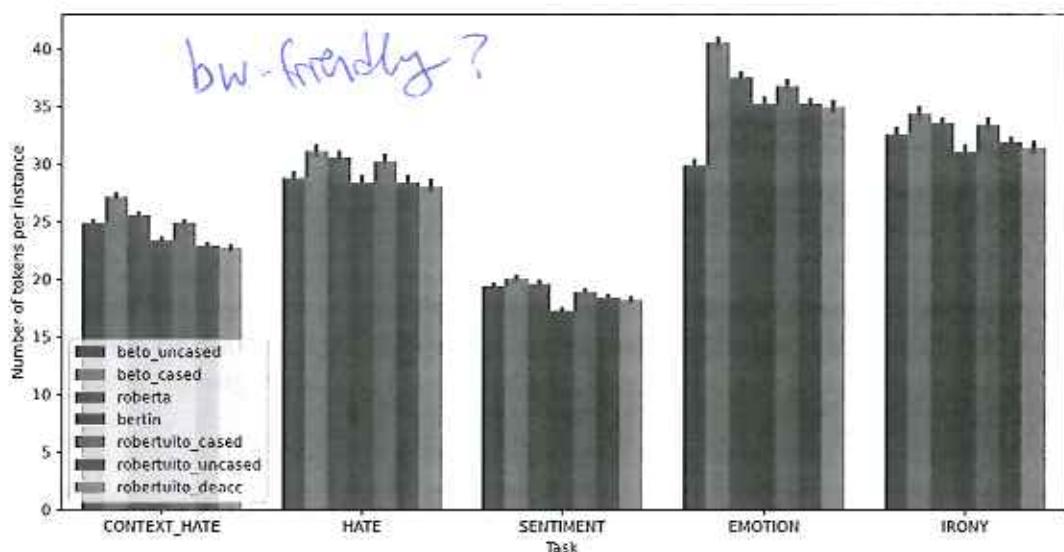


Fig. 7.1: Distribución de la cantidad de tokens por instancia para los tokenizadores de cada modelo. Las barras están agrupadas por tarea y muestran la media de la distribución junto a su intervalo de confianza a 95 %.

menor rendimiento en general. Además, entre los modelos de *RoBERTuito*, podemos observar que la versión `deacc` tiene una longitud media ligeramente menor en comparación con la versión `uncased`.

### 7.3. Adaptación de modelos pre-entrenados

Acabamos de observar que *RoBERTuito* obtiene una mejor performance que otros modelos pre-entrenados. Ahora, ¿puede ser esta mejora replicada adaptando otro modelo de lenguaje? Esta pregunta tiene –más allá del interés teórico de si un modelo de lenguaje entrenado en un dominio distinto puede adaptarse con éxito a un dominio diferente– dos consideraciones prácticas: en lenguajes de recursos relativamente bajos, entrenar un modelo desde cero como realizamos en la anterior sección puede ser prohibitivo en términos de recursos<sup>8</sup>; y por otro lado, reducir los inmensos costos computacionales de entrenar puede ser de interés, tanto en términos económicos como ambientales.

Para contestar esta pregunta, realizamos una adaptación de dominio a un modelo BETO sobre textos generados por usuario—concretamente, los mismos datos usados para entrenar *RoBERTuito*—y probamos su performance sobre el benchmark de tareas sociales descrito en la sección anterior.

<sup>8</sup> Con los recursos instalados de nuestro laboratorio es prácticamente imposible pre-entrenar un modelo desde cero

### 7.3.1. Metodología

Para realizar la adaptación de dominio, seguimos las recomendaciones de Gururangan et al. [58]: tomar un gran conjunto de datos no anotado de textos sociales, y correr la tarea de Masked Language-Modeling sobre estos. En términos de ese trabajo, estaríamos realizando *Domain Adaptation Pre-training (DAPT)*, que consta de usar un conjunto de datos relativamente general. Para ello, utilizamos los mismos datos recolectados para entrenar a *RoBERTuito*.

Tomamos las versiones cased y uncased de BERT, y corrimos únicamente la tarea de MLM sobre estos datos. En lugar de correr por 12.5K pasos de optimización como es sugerido en este trabajo, probamos con 2.5K, 5K, 10K y 20K pasos de optimización, para analizar también el impacto de este hiperparámetro. Finalmente, nos quedamos con la configuración que obtuvo el mejor resultado en términos del benchmark analizado.

Para entrenar estos modelos, usamos una TPU v2-8, gracias también al programa TRC de Google. Cada paso de optimización tomó alrededor de 2.5 segundos. Usamos una configuración similar a la descripta para el entrenamiento de *RoBERTuito* (ver tabla 7.2), con un learning rate ligeramente superior ( $5 * 10^{-4}$ ) y limitando también la longitud de secuencia a 128 tokens.

### 7.3.2. Resultados

En la figura 7.2 podemos observar la performance para los modelos de lenguaje (*BETO* y *RoBERTuito*) así como también para las versiones con ajuste de dominio de *BETO*. Podemos observar que, para los modelos *cased*, aumentar el pre-entrenamiento parecía coincidir con una mejor performance; salvo para el caso de la tarea de detección de odio contextualizado. Una posible razón detrás de esto es que la tarea planteada tiene diferencias con el dominio sobre el cual ajustamos: utilizamos pares de tweets, uno de ellos (el contexto) proveniente de un medio periodístico. También puede argumentarse que el dominio de tweets en general es demasiado amplio [36].

En el caso de los modelos *uncased*, esta mejora es un poco menos clara: por ejemplo, podemos observar que a los 5,000 pasos de optimización la performance sufre una caída con respecto a otras tareas. Esto puede deberse a problemas en la recolección de datos en los cuales entrenamos *RoBERTuito*, que se magnifican al realizar pocas actualizaciones. Sin embargo, salvando este caso puntual, observamos mejoras en todas las tareas, otra vez salvando el caso de la detección de odio contextualizada.

La tabla 7.5 muestra las medias de los resultados junto a sus desviaciones estándar para los modelos *uncased*, que tienen los mejores resultados (ver en apéndice E la tabla completa). Seleccionamos como modelos ajustados a dominio (indicados con el subíndice *FT*) a aquellos modelos que obtuvieron una mejor performance entre aquellos que entrenamos con distinta cantidad de pasos de optimización. Haciendo una comparación entre los modelos *BETO* y *RoBERTuito* uncased, vemos en el caso de la tarea de HatEval que el gap es de alrededor de 4.4 puntos de F1, mientras que la versión *FT* achica esa diferencia a 2.26 puntos F1, una reducción del 48 % del gap de performance. En el caso de Sentiment Analysis, pasamos de un gap de 5.8 puntos de F1 a uno de 2.7, una achicando en un 53 % la diferencia de performances.

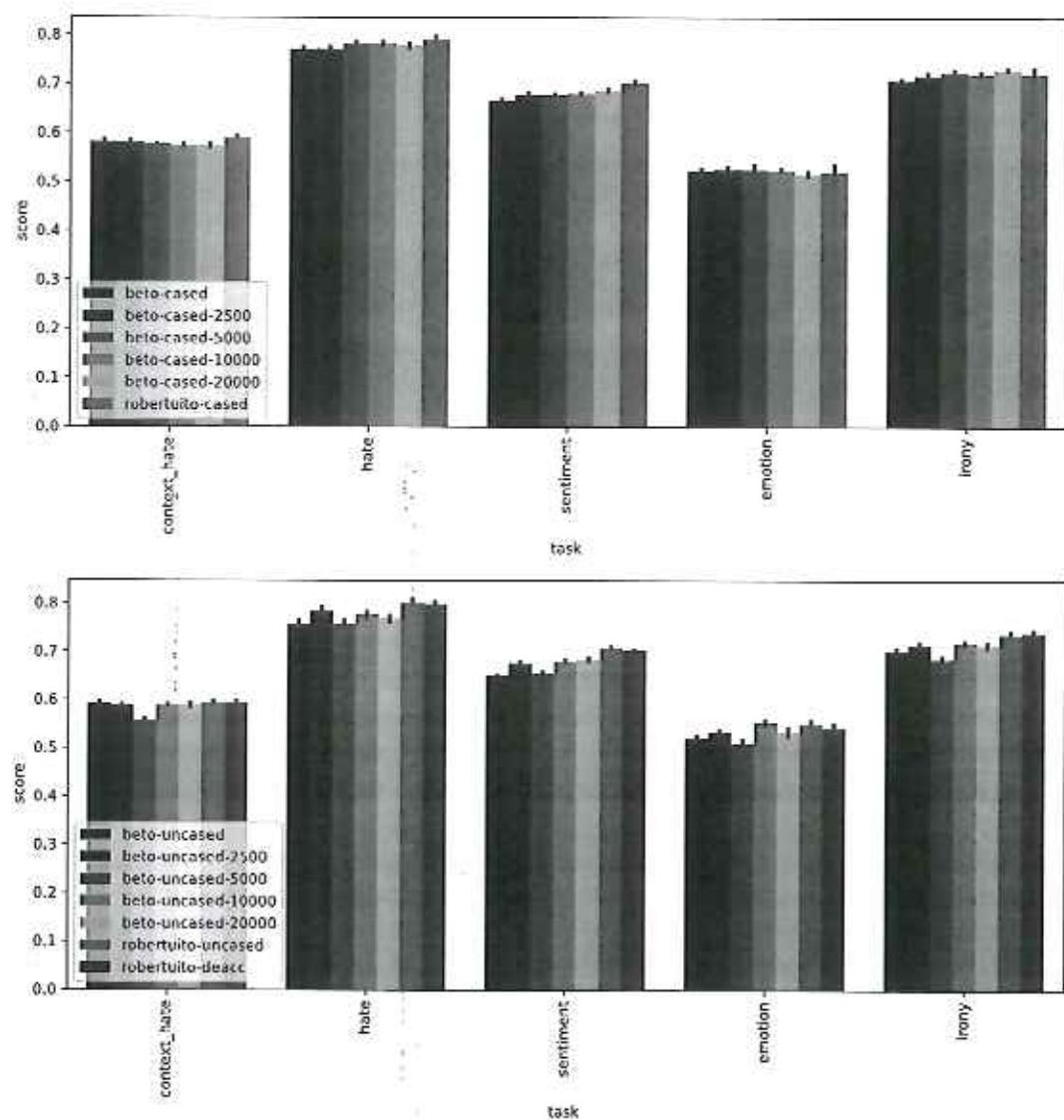


Fig. 7.2: Resultados sobre el benchmark de los modelos BETO y *RoBERTuito* (en versiones cased y uncased). Las barras están agrupadas por tarea y muestran la media de la performance sobre 15 corridas, junto a su intervalo 95 %. En tonalidad de rojo oscuro a amarillo están los modelos ajustados a dominio (más claro, más ajuste de dominio). El número al final de los modelos indica la cantidad de pasos de optimización realizados en el ajuste de dominio. En tonos azules las variantes de *RoBERTuito*

No hagas esto por favor. Hay mucha gente del tonismo.  
usd grises, negados, etc. con algún grado de

Modelo	CONTEXT	HATE	SENTIMENT	EMOTION	IRONY	score
beto-uncased	0,591 ± 0,006	0,757 ± 0,012	0,649 ± 0,005	0,521 ± 0,006	0,702 ± 0,008	0,644
beto-cased	0,582 ± 0,007	0,768 ± 0,012	0,665 ± 0,004	0,521 ± 0,012	0,706 ± 0,007	0,648
beto-cased <sub>FT</sub>	0,572 ± 0,006	0,777 ± 0,009	0,686 ± 0,005	0,517 ± 0,009	0,730 ± 0,004	0,656
beto-uncased <sub>FT</sub>	0,588 ± 0,003	0,775 ± 0,015	0,680 ± 0,004	0,553 ± 0,009	0,717 ± 0,005	0,663
robertuito-cased	0,590 ± 0,005	0,790 ± 0,012	0,701 ± 0,012	0,519 ± 0,032	0,719 ± 0,023	0,665
robertuito-deacc	0,593 ± 0,006	0,798 ± 0,008	0,702 ± 0,004	0,543 ± 0,015	0,740 ± 0,006	0,675
robertuito-uncased	0,593 ± 0,004	0,801 ± 0,010	0,707 ± 0,004	0,551 ± 0,011	0,736 ± 0,008	0,678

Tab. 7.5: Resultados de la evaluación de modelos pre-entrenados y modelos ajustados en dominio para el benchmark de tareas sociales: CONTEXT es contextualized hate speech, HATE es hate speech detection sobre el dataset de hatEval, SENTIMENT, EMOTION e IRONY son análisis de sentimiento, emociones e ironía sobre los corpus de TASS. Todos los scores son Macro F1s. beto-cased-ft y beto-uncased-ft son modulos adaptados al dominio social. Score es la media de cada fila.

En el caso de Emotion Analysis, este gap pasa de 3 puntos de F1 a 0, de hecho logrando mejores resultados. Finalmente, en el caso de detección de ironía, el gap de 3.4 puntos de F1 pasa a 1.9, una reducción de 44%.

#### 7.4. Discusión

En primer lugar, *RoBERTuito* presenta mejoras significativas para casi todas las tareas, con picos de hasta casi 4 puntos de F1 para la tarea de Sentiment Analysis contra la mejor versión de *BETO*. La mejora es muy marginal en el caso de la tarea del dataset construido en el capítulo 5, y no pareciera ser significativa. Esto puede deberse a que este dataset tiene una estructura bastante diferente de la de las otras tareas: cada instancia es una composición de dos tweets, uno de los cuales (el contexto) es en la mayoría de los casos un titular de diarios (formulado como un tweet), y por otro lado el comentario efectivo que se analiza como discurso de odio. Esto puede mitigar las potenciales mejoras de *RoBERTuito* por dos razones: el dominio del contexto no es demasiado distinto que el dominio de entrenamiento de *BETO*, y el tipo de pre-entrenamiento sólo con MLM y no en pares de tweets (recordemos que sólo hacemos MLM y no Next-Sentence Prediction). Sobre la segunda posible razón, podemos observar (para la tarea de detección de discurso de odio contextualizado) que los modelos con pre-entrenamiento basado en *RoBERTa* (*roberta-bne* y *bertin*) tienen peor performance que las dos versiones de *BETO*.

De los distintos tipos de normalización de texto utilizados en *RoBERTuito* (*cased*, *uncased* y *deacc*), podemos observar que los modelos *uncased* y *deacc* obtienen mejores performances en general. Si bien el modelo *uncased* muestra una ligera performance superior al modelo *deacc*, esta mejora no es significativa, y esto indicaría que remover tildes no reporta una degradación de la performance. Esto es esperable ya que usualmente no se utiliza esta marcación en el español “vulgar” utilizado en las redes sociales. Sin embargo, para llegar a esta conclusión sería necesario realizar pruebas más extensivas sobre otras tareas y verificando el correcto pre-entrenamiento de estos modelos.

Con respecto a los experimentos de adaptación de dominio, podemos observar

*Referencia lo mismo con una frase  
que dice que es Robertito.*

que adaptando BETO obtenemos una mejora en todos las tareas contra la versión no ajustada. Comparado con *RoBERTuito*, las versiones a las que les realizamos fine-tuning logran recortar alrededor del 50 % del performance gap entre *RoBERTuito* y BETO; en algunos casos, incluso reduciéndolo a 0. Esta comparación, sin embargo, no es del todo justa ya que BETO fue entrenado de una manera distinta que *RoBERTuito*. Por cuestiones de tiempo no pudieron ser realizados sobre *roberta-base* (principalmente, ya que éste modelo y *bertin* fueron lanzados mientras hacíamos estos experimentos) pero debería esto ser verificado en el futuro. Puede que este gap sea reducido aún más partiendo desde este otro modelo, y analizando algunas otras opciones que no tratamos en este capítulo: por ejemplo, agregar vocabulario en el ajuste de dominio, algo que por ejemplo la librería *fastai* realiza en su implementación de ULM-FIT.

Una consideración práctica de la adaptación de dominio es que permite, en lugar de realizar un costo pre-entrenamiento desde cero (como en el caso de MedBERT y SciBERT que ya relatamos anteriormente), mejorar la performance de un modelo de lenguaje ya entrenado de una manera relativamente económica. En términos concretos, un ajuste de dominio puede realizarse utilizando una placa de GPU en uno o dos días de entrenamiento, mientras que pre-entrenar un modelo desde cero requiere acceso a un hardware más oneroso. Algunos trabajos recientes [70] muestran alternativas para hacer esto con recursos reducidos ajustando varios hiperparámetros y usando algunas técnicas de optimización reciente (como LAMB [152]); sin embargo, muchos de estos setups están lejos del alcance de los recursos disponibles de muchos laboratorios. En este escenario, aplicar una optimización de dominio aparece como una alternativa mucho más factible.

Una de las limitaciones de lo estudiado es que *RoBERTuito* fue entrenado sólo por 600k pasos de optimización, contra los casi 900k pasos de BETO, y los 1M de *BERTweet*. Hay que observar, que la optimización de BETO se da con un batch size menor (512 vs 4k que usa *RoBERTuito*) y la de *BERTweet* se hace un batch size mayor (7k). En el caso de los modelos basados en *RoBERTa* en español, no tenemos disponible esa información. Con lo cual, esta comparación no es totalmente justa. Otra limitación es la escasa disponibilidad de tareas en español para textos sociales por fuera de clasificación: en ?, por ejemplo, se estudian también problemas de POS tagging y de NER para textos sociales en inglés.

puede no ser / quizás no sea

## 7.5. Conclusiones

En este capítulo, hemos abordado la tarea de mejorar la performance de la detección de discurso de odio para las tareas planteadas en esta tesis y en el contexto más general de tareas de clasificación sobre textos sociales en español. Para ello, utilizamos como benchmark varias de las tareas que vimos en esta tesis: detección de discurso de odio (en sus dos versiones, no contextualizada y contextualizada), análisis de sentimiento, análisis de emociones, y detección de ironía.

En primer lugar, y en la corriente de modelos pre-entrenados sobre distintos dominios, generamos un nuevo y valioso recurso para la clasificación de textos sociales: *RoBERTuito*, un modelo de lenguaje basado en *RoBERTa* sobre tweets en español. Para ello, recolectamos una gran base de datos de tweets en español, y utilizando

las TPU provistas por Google realizamos el pre-entrenamiento de este modelo. Los experimentos de clasificación sobre el benchmark arrojaron que *RoBERTuito* obtiene mejores significativas sobre otros modelos en español. Así mismo, observamos que remover tildes en el preprocessado no reporta una degradación significativa en la performance.

Por otro lado, exploramos un ajuste de dominio sobre modelos actuales para comparar la ganancia de performance y compararla contra *RoBERTuito*. Para ello, tomamos los modelos de *BETO* (en sus versiones cased y uncased) y corrimos la tarea de MLM sobre los tweets recolectados para entrenar nuestro modelo anterior. Si bien la performance de estos modelos ajustados mejora con respecto de *BETO*, se mantiene por debajo de *RoBERTuito*, aunque recortando considerablemente el gap de performance entre ambos modelos. De todas formas, este análisis puede ser de consideración para aquellos lenguajes con menos recursos que no pueden pre-entrenar modelos de lenguaje desde cero. Resta como trabajo futuro realizar estos experimentos sobre más tareas más ricas que no sean exclusivamente de clasificación, y realizar los ajustes sobre los modelos *RoBERTa* en español para hacer una comparación más justa.

Todos estos experimentos han sido realizados en español. Publicamos tanto el modelo *RoBERTuito*<sup>9</sup>, el código para entrenarlo y para correr el benchmark con otros modelos pre-entrenados<sup>10</sup> y la base de datos de tweets en español.

## 7.6. Notas

En Pérez et al. [114] puede encontrarse la descripción de la construcción de *RoBERTuito*, aproximadamente la mitad de este capítulo. En el apéndice E puede encontrarse la tabla completa de resultados para las distintas cantidades de pasos de optimización.

<sup>9</sup> <https://huggingface.co/finiteautomata/robertuito-base-uncased>

<sup>10</sup> Ambos en <sup>11</sup>



## 8. CONCLUSIONES

En esta tesis hemos abordado la tarea de la detección de discurso de odio, intentando empujar el estado del arte y proponiendo enfoques superadores a lo que es gran parte de lo que se hace hoy día, basado en la detección binaria de este fenómeno. En ese sentido, propusimos la tarea de detección de odio contextualizada, algo que había sido poco estudiando en la literatura. Construimos un dataset etiquetado para discurso de odio sobre comentarios de usuarios de Twitter sobre artículos periodísticos en español rioplatense. Teniendo en cuenta una breve reseña de algunos trabajos previos, decidimos realizar este trabajo con anotadores nativos de esta variedad dialectal, y no con terceros a través de plataformas, para poder obtener etiquetas de mayor calidad.

En base a los experimentos de clasificación realizados, hemos podido brindar cierta evidencia de que el contexto –en este caso, en forma de tweet de medio periodístico– puede aprovecharse para identificar discursos de odio mejorando la performance de clasificadores basados en técnicas del estado del arte. Si bien los experimentos que realizamos no mostraron una mejora en el rendimiento utilizando un contexto más largo –en forma de artículo periodístico completo– trabajo futuro debería explorar cómo incorporar esta información al clasificador. De manera heurística, podríamos argumentar que los humanos muchas veces utilizamos este contexto “largo” implícitamente, accediendo por otras vías a información sobre la noticia en cuestión –por ejemplo, si la persona sobre la que habla la noticia posee cierta característica protegida que está implícita en la nota.

Este resultado –sobre la posibilidad de utilizar información contextual– va en línea con varios trabajos recientes del área de extracción de opiniones en redes sociales que muestran que la utilización de varias fuentes de información es beneficiosa para los algoritmos de detección. Esto es algo esperable ya que nuestra percepción de la realidad dista de ser unimodal –sólo percibiendo un comentario o un texto en aislamiento– sino que incorpora diversos elementos: desde el tópico de la conversación, quiénes son los interlocutores, entre otras cuestiones. Dentro de estas diversas.

Un punto que observamos es que la predicción de múltiples características además de la mera existencia del discurso de odio no sólo no empeora la performance de los clasificadores sino que la mejora marginalmente para las técnicas del estado del arte basadas en modelos pre-entrenados de lenguaje. Es decir, si en vez de sólo predecir que existe o no discurso de odio predecimos más características –como ser la o las características ofendidas, si existe un llamado a la acción violenta, si está dirigido a un grupo o un individuo– podemos, por un lado, mejorar la interpretabilidad y la riqueza de la salida de los algoritmos de detección; y por otro, mejorar su rendimiento al brindarles una señal más poderosa en su entrenamiento. De esto, podemos fomentar que los próximos datasets de discurso de odio se anoten con más variables a predecir –no sólo la etiqueta binaria– y en lo posible marcando las características protegidas que se estén vulnerando en cada instancia.

Respecto a las limitaciones puntuales de este trabajo, puede marcarse por un lado

el hecho de que sólo realizamos el etiquetado viendo el contexto de cada comentario. Algunos trabajos previos (como [109]) han usado ambos tipos de etiquetado –contextualizado y no contextualizado– para entrenar los clasificadores contextualizados sobre datos anotados sobre etiquetadores que vieron el contexto, y lo mismo para aquellos no contextualizados. → Explíquese un poco de por qué cosa bueva va.

Debemos ser, sin embargo, cautelosos acerca de los resultados obtenidos en este trabajo y, en líneas generales, de la mayoría de los avances en la detección de discurso discriminatorio. El estado del arte actual en NLP está basado en modelos de lenguajes neuronales pre-entrenados de lenguaje, altamente potentes. En términos de lo mencionado en *The Book of Why* de Judea Pearl [110], estos sistemas están en una etapa increíblemente “asociacional”. Es decir, nuestros modelos de lenguaje (muy a pesar de que muchos artículos hablen sobre cierto “entendimiento” por parte de estos) tan sólo detectan regularidades en los datos, como si fueran sólamente el ajuste de una curva que un estadístico realiza hace más de un siglo, sin realizar ningún tipo de razonamiento de lo que se está haciendo.

En nuestro problema concreto, un clasificador puede detectar que decirle “sos hombre” a un artículo relacionado a una mujer (quizás trans) conlleva discurso de odio contra la comunidad LGBTI. Sin embargo, este mismo mensaje ofuscado de alguna manera (por ejemplo, preguntándole el nombre, o alguna otra forma que no hayamos observado en los datos) logra burlar a nuestros sistemas. Ligado a este ejemplo, [12] ilustran este punto: nuestros actuales sistemas, basados en modelos de lenguaje, aún en sus formas más complejas y sobreparametrizadas con miles de millones de parámetros, no son más que “loros estadísticos”, muy hábiles en detectar regularidades y hacernos creer que llevan adentro algún tipo de razonamiento. Sin embargo, la realidad es que no lo tienen, ya que el entrenamiento sobre la mera “forma” del lenguaje –es decir, los gigabytes de texto de entrenamiento de BERT, GPT y amigos– no conlleva ningún entendimiento ni razonamiento. Podemos decir de todo esto –y parafraseando a Mitchell [98]– que la **detección de discurso discriminatorio es más difícil de lo que creemos**.

¿Significa esto que los sistemas actuales no sirven para nada? En absoluto. Los actuales algoritmos de detección, aún con sus defectos y siendo bastante rudimentarios, logran detectar parte del lenguaje discriminatorio que observamos en redes sociales. Sin embargo, es necesario entender sus limitaciones: a medida que estos sistemas puedan encontrar regularidades con más detalle, muchos usuarios ocultarán este discurso de manera más sofisticada para lograr saltar su escrutinio (en caso de que estemos hablando de sistemas que se usen con fines de moderación). Teniendo estas cuestiones en cuenta, planteamos que agregar más información y contexto a nuestros algoritmos puede ayudarlos a mitigar parcialmente sus limitaciones.

Para cerrar, un eje que atraviesa este trabajo es que lo realizamos íntegramente en español. La mayor parte de la literatura sobre este tema es en inglés, y entendiendo que el discurso de odio es un fenómeno social y cultural, es necesario estudiarlo en otros idiomas. Por eso, este trabajo intenta aportar a balancear la asimetría de recursos tanto en el área particular y específica de detección de discurso de odio como así también en la de NLP en general.

¡Muy bueno!

④ Me gustaría pensar en más de un conocido... ☺