

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



“DETECCIÓN DE DISCURSOS DE ODIO EN REDES SOCIALES CHILENAS MEDIANTE MÉTODOS DE APRENDIZAJE AUTOMÁTICO”

DOMINGO BENOIT CEA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Dr. Juan Ricardo Ñanculef Alegría
Profesor Correferente: Dr. Marcelo Mendoza

Julio - 2023

DEDICATORIA

A mi abuela Nella Margarita Marchetti Pareto, quién me enseñó la importancia de la perseverancia, la dedicación, la curiosidad y la satisfacción de un trabajo bien hecho.

AGRADECIMIENTOS

Agradezco a mis profesores de la Universidad Técnica Federico Santa María, cuya formación ha sido fundamental en entregarme los conocimientos y valores que me han convertido en la persona que soy y espero ser, en especial a mi profesor guía Ricardo Ñanculef, cuyo consejo y guía fue fundamental en la realización de este trabajo.

Agradezco a mis amigos y compañeros a lo largo de la carrera con quienes comparto incontables noches en vela, estudiando, programando, sufriendo y tantos otros sentimientos imposibles de retratar en palabras.

Agradezco a los colaboradores de este trabajo Paulina Gómez, Sebastián Campos, Diego Pinto, Axel Reyes, Álvaro Fuentes, Sebastián Alvarado, Ignacio Jorquera, Gabriel Ortega, Sergio Rosales, Ignacia Delaigue, Gerald Méndez, Jorge Juan, Francisco Riveros, Joaquín Concha, Maximiliano Ojeda y Paula Hasson, sin los cuales nada de esto habría sido posible.

Finalmente, agradezco a mi familia, mi madre, mis abuelos, mis hermanas, mi polola y todos mis otros seres queridos, sin los cuales nada de esto habría sido posible o valido la pena lograr.

RESUMEN

Resumen—Desarrollar herramientas para identificar Discursos de Odio en redes sociales es un paso importante para la comprensión de este fenómeno a nivel global. Lamentablemente, la gran mayoría de los recursos disponibles se encuentran orientados al idioma inglés. Por otro lado, los pocos recursos existentes en español no suelen tomar en cuenta la gran diversidad de dialectos existentes, lo que dificulta evaluar el impacto de características socio-culturales específicas al momento de generalizar soluciones de un dialecto o lengua a otra.

En este trabajo se introduce un nuevo recurso orientado al estudio del Discurso de Odio con *tweets* en el dialecto chileno del español. El corpus incluye 4572 registros recolectados de modo semi-automático de *Twitter* y posteriormente validados y anotados manualmente por un grupo de 15 humanos, garantizando 3 anotaciones independientes por *tweet*. El *dataset* incluye un 45,5 % de casos positivos para la clase *Hate Speech*, entregando así datos mejor balanceados que los recursos ya existentes para este idioma. Las anotaciones entregadas permiten también identificar en cada caso si el *tweet* se refiere o no a una de cuatro comunidades que suelen ser objetivo de mensajes provocativos u ofensivos: mujeres, migrantes, pueblos originarios, y comunidad LGBTQ+, permitiendo así análisis más detallados del fenómeno. Junto con lo anterior, se dispone junto a cada *tweet* el hilo o conversación del cual forma parte, enriqueciendo los actuales recursos disponibles en términos contextuales.

Junto a la metodología de recolección y anotación empleada, este trabajo presenta una serie de experimentos orientados a evaluar la calidad del corpus obtenido utilizando tres modelos multi-lingua y tres *datasets* alternativos del estado del arte. En primer lugar, se estudió la *transferibilidad* del recurso propuesto, es decir, su capacidad para mejorar la detección de odio sobre otros *datasets*. En segundo lugar, se realizaron experimentos orientados a detectar la presencia de *sesgos* en los datos recolectados considerando siete grupos o comunidades protegidas. Finalmente, se efectuaron 27 *pruebas funcionales* recientemente propuestas por *HATECHECK*[55] para evaluar modelos en casos en que éstos tienden a fallar o sobre-especializarse.

Los resultados obtenidos demuestran que, con respecto a su único símil en dialecto chileno, el *dataset* generado es más transferible a otros dialectos e idiomas, exhibe menores sesgos, y permite superar con mayor éxito la mayoría de las pruebas funcionales, incluso en dialectos diferentes al entrenado. Comparando con recursos en otros idiomas, el *dataset* generado es altamente competitivo, permitiendo por ejemplo superar el estado del arte en 16 de las 27 pruebas funcionales de *HATECHECK* y logrando la mejor transferencia a español chileno e inglés. Como desventajas, el corpus mostró sesgos ligeramente más altos que *datasets* en otros idiomas y no logró mejorar la transferencia al español castellano, temas que deben ser abordados con mayor profundidad en el futuro. Todo el material generado ha sido publicado para facilitar su uso y reproducción por parte de la comunidad. IEEE Dataport

Palabras Clave— Discursos de Odio; Procesamiento de Lenguaje Natural; Aprendizaje Automático; Lingüística Computacional; Conjuntos de Datos.

ABSTRACT

Abstract—Developing algorithms to identify Hate Speech in social media is essential in studying these phenomena globally. Sadly, most available corpora to build these algorithms are in English. On the other hand, the few available Spanish resources do not cover the great diversity of dialects existing for this language. As a result, studying specific sociocultural attributes of Hate Speech in most Spanish-speaking countries like Chile is challenging because algorithms need to be transferred from other contexts.

This work introduces a new corpus focused on Chilean Hate Speech. The corpus includes 4572 tweets recollected from Twitter in a semi-automatic fashion, which were subsequently annotated and validated by a group of 15 humans, ensuring three independent annotations by tweet. The dataset includes 45.5 % of positive cases of Hate Speech, which corresponds to a significantly better-balanced dataset than existing resources for this language. Our annotations also allow identifying whether the tweet refers to one of four communities often targeted by hate speech: women, migrants, indigenous peoples, and the LGBTQ+ community, thus supporting a more detailed analysis of the phenomenon. In addition, we make available the thread or conversation to which the tweet belongs, enriching the currently available resources in contextual terms.

Besides the data collection and annotation methodology, this manuscript presents a series of experiments to evaluate the quality of the obtained corpus using three multi-lingual models and three alternative state-of-the-art datasets. Firstly, we analyze the transferability of the proposed resource, i.e., its ability to enhance hate detection on other datasets. Secondly, we conduct experiments to detect the presence of biases in the collected data, considering seven protected groups or communities. Finally, 27 functional tests, recently proposed by HATECHECK [55], were carried out to evaluate models in cases where they tend to fail or over-specialize.

The obtained results demonstrate that, compared to its only counterpart in the Chilean dialect, the generated dataset is more transferable to other dialects and languages, exhibits fewer biases, and allows for achieving better results in the majority of functional tests, even though these tests are a different language. Compared with resources in other languages, the generated dataset is highly competitive, outperforming the state-of-the-art in 16 out of 27 functional tests from HATECHECK and achieving the best transferability to

Chilean Spanish and English. Among the disadvantages, we observed slightly more biased predictions and, training state-of-the-art models on the corpus, we could not improve the transference to Standard Spanish. All the generated data and code have been made publicly available to facilitate their use by the community [IEEE Dataport](#) [IEEE Dataport](#).

Keywords— Machine Learning; Hate Speech; Natural Language Processing; Computational Linguistics; Datasets.

GLOSARIO

API: Application programming interface.

ANN: Artificial Neural Network.

CNN: Convolutional Neural Networks (Redes Neuronales Convolucionales).

CV: Computer vision (Visión por computadora).

DI: Departamento de Informática.

GCN: Graph Convolutional Network

GELU: Gaussian Error Linear Unit

IAA: Inter Annotator Agreement (Nivel de acuerdo entre anotadores).

KNN: K -nearest neighbors.

LASER: Language-Agnostic Sentence Representations.

LSTM: Long Short-Term Memory.

MUSE: Multilingual Unsupervised and Supervised Embeddings.

NLP: Natural Language Processing (Procesamiento de Lenguaje Natural).

ONU: Organización de las Naciones Unidas.

ReLU: Rectified linear unit.

RNN: Recurrent Neural Networks (Redes Neuronales Recurrentes).

SVM: Support Vector Machines.

UTFSM: Universidad Técnica Federico Santa María.

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	V
GLOSARIO	VII
ÍNDICE DE FIGURAS	X
ÍNDICE DE TABLAS	XI
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	3
1.1 Objetivos	4
1.1.1 Objetivo General	4
1.1.2 Objetivos Específicos	4
1.2 Impacto inicial de solucionar el problema	5
CAPÍTULO 2: MARCO CONCEPTUAL	6
2.1 Trabajo Relacionado en Metodologías de Muestreo	6
2.1.1 Descripción	6
2.1.2 Estado del Arte	6
2.1.3 Herramientas de Muestreo	9
2.2 Trabajo Relacionado en Metodologías de Anotación	11
2.2.1 Descripción	11
2.2.2 Estado del Arte	11
2.2.3 Plataformas de <i>Crowdsourcing</i>	18
2.3 Trabajo Relacionado en Detección de <i>Hate Speech</i>	20
2.3.1 Procesamiento de Lenguaje Natural o <i>NLP</i>	20
2.3.2 Clasificadores basados en Aprendizaje Automático	21
2.3.3 Estado del Arte Modelos	35
2.4 Métricas	40
2.4.1 Métricas de Rendimiento	40
2.4.2 Métricas de Consenso	43
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	45
3.1 Metodología de Muestreo	45
3.1.1 Criterios y Conceptos Clave	45
3.1.2 Herramienta de Muestreo	46
3.1.3 Esquema General	47
3.2 Metodología de Anotación	47
3.2.1 Criterios y Conceptos Clave	48

3.2.2	Plataforma de <i>Crowdsourcing</i>	48
3.2.3	Esquema General	49
3.2.4	Esquema de anotado	50
3.3	Evaluación de los Datos	51
3.3.1	Diseño Experimental	52
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN		59
4.1	Resultados Recolección	59
4.2	Resultados Anotado	60
4.3	Resultados experimentales	65
4.3.1	Experimentos de transferencia	66
4.3.2	Experimentos de Sesgo	69
4.3.3	Experimentos <i>Hatecheck</i>	76
CAPÍTULO 5: CONCLUSIONES		83
5.1	Definición de Herramientas, Metodología y Criterios de Recolección	83
5.2	Metodologías y Criterios de anotación	83
5.3	Resultados de Anotación	84
5.4	Modelos <i>Baselines</i> y <i>Benchmarks</i>	84
5.5	Publicación y Divulgación	86
5.6	Trabajo Futuro	86
ANEXOS		88
ANEXO A: Palabras claves, cuentas y <i>hashtags</i>		88
ANEXO B: Lineamientos		89
REFERENCIAS BIBLIOGRÁFICAS		93

ÍNDICE DE FIGURAS

1	Taxonomía de Khan <i>et al.</i>	15
2	Clases semánticas de Sharma <i>et al.</i>	15
3	Ejemplo de una neurona.	23
4	Ejemplo de una red <i>Feed Forward</i>	24
5	Arquitectura de las neuronas de una RNN, Fuente	26
6	Diagrama de bloque de una célula LSTM en una RNN, Fuente: Goodfellow <i>et al.</i>	28
7	Esquema de la red convolucional VGG-16	30
8	(a) Es el kernel producido al utilizar una convolución sin dilatación (equivalente a una dilatación igual a 1), (b) Corresponde al kernel (a), pero con dilatación 2 ($3 \times 3 \rightarrow 7 \times 7$), (c) Es generado al aplicar una dilatación de 4 sobre (a) ($3 \times 3 \rightarrow 15 \times 15$). Al utilizar una dilatación de l , esto significa que son $l - 1$ píxeles omitidos entre cada píxel del kernel. En esta figura, los puntos rojos muestran los píxeles de la imagen utilizados por el kernel, y el coloreado azul representa el nuevo campo receptivo alcanzado por el kernel sobre la imagen.	32
9	Arquitectura del modelo <i>Transformers</i>	33
10	(Izquierda) Atención de Producto punto escalado. (Derecha) Atención multicabezal consistente en varias capas de atención corriendo en paralelo.	35

11	(A) Hay dos distribuciones de <i>embeddings</i> , primero palabras en Inglés mostradas en rojo denotadas por X y segundo palabras en Italiano mostradas en azul denotadas por Y , se quiere alinear estos <i>embeddings</i> de tal forma que se obtiene la traducción. Cada punto representa una palabra en el espacio. El tamaño del punto es proporcional a la frecuencia de la palabra en el corpus de entrenamiento. (B) Mediante entrenamiento adversativo se busca aprender una matriz de rotaciones W , tal que, alinee ambas distribuciones con cierto margen de error. Las estrellas verdes, son pares de palabras seleccionadas aleatoriamente, las cuales se entregan al discriminador para discernir si los <i>embeddings</i> pertenecen a una misma distribución. (C) El mapeo W es refinado utilizando análisis de Procrustes ¹ . Este método utiliza palabras frecuentes alineadas por el paso anterior como puntos ancla y minimiza la función de energía correspondiente a un sistema de resortes entre anclas. El mapeo refinado, es luego usado para mapear todas las palabras en el diccionario. (D) Finalmente, se traduce utilizando el mapeo refinado W y una métrica de distancia, denominada CROSS-DOMAIN SIMILARITY LOCAL SCALING, la cual expande regiones del espacio con alta densidad de puntos (i.e. el área alrededor de “cat” en (A)), haciendo así que palabras centralizadas como “cat” se encuentren más separadas de otros vectores de palabras que como lo estarían originalmente en (A).	37
12	Arquitectura Modelo ACL19.	38
13	Arquitectura generadora de <i>embeddings</i> para LASER.	39
14	Arquitectura de clasificación Modelo EMNLP20.	40
15	Esquema de recolección de datos.	48
16	Ilustración del anotado de tweets con su thread en LightTag [47].	49
17	Histogramas de Cantidad de Votos por clases.	61
18	Diagramas circulares de distribución de Clases donde los anotadores lograron consenso (votos ≥ 2).	62
19	Nube de palabras para la categoría de Odio	62
20	Nube de palabras para la categoría de No Odio	63

ÍNDICE DE TABLAS

1	Herramientas de web scrapping	11
---	---	----

2	Tabla resumen metodologías de recolección y anotado por publicación	19
3	Funciones de activación de una Neurona.	23
4	Información estadística de los <i>datasets</i> utilizados. C corresponde al número de clases (etiquetas) correspondiente a cada conjunto de datos. $ Dataset $ representa la cardinalidad de datos del conjunto. Odio representa el porcentaje de Odio en cada <i>dataset</i>	52
5	Configuración de experimentos de transferencia.	54
6	Descripción recurso <i>Hatecheck</i> [55].	56
7	Configuración de experimentos de funcionalidades para <i>Hatecheck</i> y evaluación de sesgos.	57
8	Asignación de criterios de búsqueda para Recolectores.	60
9	Distribución de clases en <i>Tweets</i> con votos mayor o igual a 2.	63
10	Alfa de Krippendorff de grupos por clase	64
11	Kappa de Cohen por clase del Grupo 1	64
12	Kappa de Cohen por clase del Grupo 2	64
13	Kappa de Cohen por clase del Grupo 3	65
14	Kappa de Cohen por clase del Grupo 4	65
15	Kappa de Cohen por clase del Grupo 5	65
16	Resultados transferencia para ACL19. Mejores resultados para transferencia al: Inglés, Español y Chileno resaltados en verde, azul y rosado respectivamente.	66
17	Resultados transferencia para ECML20. Mejores resultados para transferencia al: Inglés, Español y Chileno resaltados en verde, azul y rosado respectivamente.	66
18	Resultados transferencia para EMNLP20. Mejores resultados para transferencia al: Inglés, Español y Chileno resaltados en verde, azul y rosado respectivamente.	67
19	Resultados de transferencia consolidados para AUC. Mejores resultados consolidados y promedio para transferencia al: Inglés, Español y Chileno resaltados en verde, azul y rosado respectivamente.	67

20	Resultados de transferencia consolidados y promedio para valor F1. Mejores resultados para transferencia al: Inglés, Español y Chileno resaltados en verde, azul y rosado respectivamente.	68
21	ACL19: Resultados de métricas de sesgo por comunidad para <i>Hatecheck</i> . Mejores y peores resultados por métrica resaltados en verde y rosado respectivamente para cada <i>dataset</i> . En los casos donde el peor o mejor resultado esta repetido entre comunidades éste no es resaltado.	70
22	ECML20: Resultados de métricas de sesgo por comunidad para <i>Hatecheck</i> . Mejores y peores resultados por métrica resaltados en verde y rosado respectivamente para cada <i>dataset</i> . En los casos donde el peor o mejor resultado esta repetido entre comunidades éste no es resaltado.	71
23	EMNLP20: Resultados de métricas de sesgo por comunidad para <i>Hatecheck</i> . Mejores y peores resultados por métrica resaltados en verde y rosado respectivamente para cada <i>dataset</i> . En los casos donde el peor o mejor resultado esta repetido entre comunidades éste no es resaltado.	72
24	ACL19: Resultados de métricas de sesgo acumuladas para <i>Hatecheck</i> . Mejores y peores resultados por métrica resaltados en verde y rosado respectivamente.	73
25	ECML20: Resultados de métricas de sesgo acumuladas para <i>Hatecheck</i> . Mejores y peores resultados por métrica resaltados en verde y rosado respectivamente.	73
26	EMNLP20: Resultados de métricas de sesgo acumuladas para <i>Hatecheck</i> . Mejores y peores resultados por métrica resaltados en verde y rosado respectivamente.	73
27	Resultados consolidados de métricas de sesgo por comunidad para <i>Hatecheck</i> . Considerando el mejor algoritmo multi-lingua para cada corpus. Mejores y peores resultados por métrica resaltados en verde y rosado respectivamente para cada comunidad.	74
28	Resultados consolidados de métricas de sesgo acumuladas para <i>Hatecheck</i> . Considerando el mejor algoritmo multi-lingua para cada corpus. Mejores y peores resultados por métrica resaltados en verde y rosado respectivamente.	75
29	ACL19: Resultados de Evaluación en <i>Hatecheck</i> utilizando la exactitud como métrica. Mejores y peores resultados por fila resaltados en verde y rosado respectivamente.	76

30	ECML20: Resultados de Evaluación en <i>Hatecheck</i> utilizando la exactitud como métrica. Mejores y peores resultados por fila resaltados en verde y rosado respectivamente.	77
32	ACL19: Resultados de Evaluación en <i>Hatecheck</i> para comunidades utilizando valor F1 Macro como métrica. Mejores y peores resultados por fila resaltados en verde y rosado respectivamente.	77
31	EMNLP20: Resultados de Evaluación en <i>Hatecheck</i> utilizando la exactitud como métrica. Mejores y peores resultados por fila resaltados en verde y rosado respectivamente.	78
33	ECML20: Resultados de Evaluación en <i>Hatecheck</i> para comunidades utilizando valor F1 Macro como métrica. Mejores y peores resultados por fila resaltados en verde y rosado respectivamente.	79
34	EMNLP20: Resultados de Evaluación en <i>Hatecheck</i> para comunidades utilizando valor F1 Macro como métrica. Mejores y peores resultados por fila resaltados en verde y rosado respectivamente.	79
35	Resultados Consolidados de Evaluación en <i>Hatecheck</i> . Considerando el mejor algoritmo multi-lingua para cada corpus. Los datos resaltados en verde corresponden al valor más alto por fila que supera el <i>benchmark</i> (XLMT 2021), mientras que los resaltados en rosado a los que empeoran el <i>benchmark</i> .	80
36	Resultados Consolidados de Evaluación en <i>Hatecheck</i> . Considerando el mejor algoritmo multi-lingua para cada corpus. Mejores y peores resultados por fila resaltados en verde y rosado respectivamente. En los casos donde el peor o mejor resultado esta repetido entre <i>datasets</i> éste no es resaltado.	81
37	Resultados Métricas de Evaluación en <i>Hatecheck</i> por conjunto de datos. Mejores y peores resultados por columna resaltados en verde y rosado respectivamente.	81
38	Pabras claves por comunidad.	88
39	Cuentas y <i>Hashtags</i> relevantes.	89

INTRODUCCIÓN

El fenómeno de Discursos de Odio o *Hate Speech*, no corresponde a un fenómeno particularmente nuevo, basta con recordar algunos de los terribles hechos ocurridos durante la segunda guerra mundial, para ver lo nocivos que pueden ser los efectos de estos en radicalizar, deshumanizar e incentivar el genocidio de otros seres humanos. Por estas razones, la lucha contra este fenómeno se ha vuelto tan relevante en un mundo altamente intercomunicado, permitir que este fenómeno se reproduzca sin restricciones en redes sociales ha llevado nuevamente a genocidios [40] y discriminación [44]. Los principales problemas de este tipo de discursos es que enfrentan y amenazan los valores democráticos por los que tanto se ha luchado, además de la estabilidad social y la paz [21].

Es por estas razones que ante el gran volumen de datos generados constantemente en el internet, la posibilidad de generar filtros manuales o poco sofisticados capaces de llevar a cabo esta tarea sin pasar a llevar derechos sociales o sin caer fuertemente en sesgos resulta muy difícil, pues carecen de la flexibilidad y mayor nivel de complejidad que es posible lograr mediante otros métodos. Ante esto surge la pregunta: ¿Qué herramientas nos permiten automatizar procesos altamente complejos?, y la respuesta es evidentemente el uso de nuevas tecnologías, en específico los algoritmos de aprendizaje automático. En este contexto, el gran volumen de datos disponibles se transforma en una ventaja para combatir el problema, pues estos mismos datos pueden ser utilizados para entrenar y ajustar modelos de clasificación capaces de abstraer y generalizar la tarea con mayor eficacia y eficiencia.

Sin embargo, para generar una primera aproximación a soluciones inteligentes para este problema, es necesario contar con datos pertinentes a la tarea en sí. Por éstas razones, es que en este trabajo se construye un corpus de casos chilenos de *hate speech* anotados por humanos, utilizando casos recolectados de manera semi-automática permitiendo así sortear los problemas asociados al desbalance de clases que se suele encontrar en la literatura, como lo evidenciado por Arango Monnar *et al.* y Albadi *et al.*. Esta metodología semi-automática detalla palabras claves, asignaciones y herramientas usadas, tanto para la parte realizada automáticamente como para la parte realizada por humanos. También se propone una metodología de anotado conforme al estado del arte con tal de ayudar a reproducir los resultados obtenidos. Finalmente, evaluamos nuestros datos en una serie de experimentos enfocados a contrastar las capacidades de generalización de distintos modelos sobre los datos generados con tal de determinar la calidad de éstos mismos y como se comparan con otros conjuntos de datos sobre los mismos modelos.

El presente documento se encuentra dividido en cinco capítulos, donde en el primer capítulo se encuentra la definición del problema, detallando los objetivos generales y específicos de esta memoria. El segundo capítulo aborda los antecedentes teóricos que dan fundamento a los conceptos, metodologías y heurísticas utilizadas para fundamentar la toma de decisiones a lo largo de la confección de esta memoria, pasando por una descripción general del estado del arte relacionado, en cuanto a herramientas, plataformas, metodologías y mode-

los utilizados en esta área de investigación. En el tercer capítulo se formaliza la propuesta de solución, describiendo la metodología que llevaremos a cabo, por qué y en qué nos inspiramos para realizarlo. En el penúltimo capítulo “Validación de la Solución” exponemos los resultados obtenidos una vez aplicada nuestra propuesta, donde planteamos un análisis de los datos anotados y sobre los resultados de experimentación. Finalmente, en el quinto capítulo exponemos nuestras conclusiones frente a los resultados obtenidos, además de sugerir ciertas líneas de investigación posibles relacionadas con trabajos futuros, sobre este mismo recurso.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

La problemática a abordar se encuentra relacionada con la aplicación de la inteligencia artificial a problemas relacionados con la discriminación y la perpetuación de discursos de odio en redes sociales.

En la reciente convención constituyente chilena [15], en el artículo 21 del Reglamento de ética, se definió el discurso de odio de la siguiente manera “Constituye discurso de odio toda comunicación, expresión verbal o de cualquier tipo, que sea un ataque o utilice lenguaje discriminatorio en relación con una persona o un grupo en razón de su origen étnico, raza, color, ascendencia, nacionalidad, credo, religión, espiritualidad u otro factor de identidad, que tenga la entidad suficiente para incitar a la discriminación, la hostilidad o la violencia.” [15]

En los últimos años, el discurso de odio se ha vuelto un tema tan relevante que incluso la ONU ha formulado planes para su lucha [21]. Ante estos antecedentes es fácil notar como el discurso de odio constituye hoy en día uno de los grandes problemas en internet, siendo de gran importancia la generación de herramientas que permitan analizarlo y combatirlo.

Es por esto que se propone una metodología para la creación de un *dataset* con base en el dialecto chileno del español para así ser capaz de diferenciar las características intrínsecas de los textos o imágenes/*memes* con fines humorísticos de aquellas que se refugian detrás de estos para expresar mensajes discriminatorios, siendo de especial relevancia el ser capaz de clasificar un texto o imagen como ofensivo o no de acuerdo a su contenido, no solo explícito sino que también en relación a las componentes más abstractas del texto. Se sugiere la generación de un corpus de casos chilenos de *hate speech* los cuales reflejen las características socio-culturales particulares (e.g. prejuicios implícitos) del país. Permitiendo así la indagación de fenómenos y contrastes socio-culturales entre el chileno y otras lenguas. Resulta de vital importancia la documentación del proceso de creación de este corpus, tanto como modo de transparentar las decisiones tomadas en su creación, al igual que aquellas que lleven a agregar atributos deseables, pero inexistentes actualmente en otros corpus.

Un gran número de publicaciones como Albadi *et al.*, Alotaibi y Hasanat, Marpaung *et al.*, Ibrohim y Budi muestran la necesidad de implementaciones de reconocimiento de texto adaptadas a la realidad socio-cultural del lenguaje utilizado con tal de poder generar mejores modelos de detección para el discurso de odio.

Papers recientes muestran nuevos avances en lo que se refiere a análisis de imágenes multimodales, como en Kiela *et al.* donde se confeccionó un *baseline* de desempeño para el *dataset* utilizado.

Independientemente del formato audiovisual en que se exprese un potencial discurso de

odio, varios trabajos recientes como en Romim *et al.* muestran que debido a lo intrínsecamente ligados que se encuentran el humor y lo que es considerado ofensivo en un lenguaje, señalando la importancia de adaptar la tarea de clasificación al lenguaje/idioma/dialecto utilizado con tal de obtener resultados satisfactorios y poder establecer comparativas entre modelos. Lo anterior se confirma en gran número de artículos publicados recientemente Albadi *et al.*, Mookdarsanit y Mookdarsanit, Luu *et al.*, Sazany y Budi, Alatawi *et al.* . Esto sugiere que liberar un *dataset* especializado en el dialecto chileno permitiría generar y facilitar avances en el área de análisis y procesamiento de lenguaje natural en el país, además de permitir la utilización de estos datos para la investigación de nuevas técnicas y modelos relacionados con estas áreas.

1.1. Objetivos

1.1.1. Objetivo General

El objetivo general de esta memoria consiste en recolectar y anotar datos de redes sociales chilenas que permitan entrenar modelos orientados a la detección y el monitoreo de discursos de odio relevantes para la realidad nacional.

1.1.2. Objetivos Específicos

Para poder llevar a cabo el objetivo propuesto con anterioridad, se propone el completar los siguientes objetivos:

- (i) Estudiar la disponibilidad de herramientas que permitan la recolección automática de datos en redes sociales como *Instagram* o *Twitter*.
- (ii) Definir una metodología de muestreo que permita recolectar ejemplos de discursos de odio en redes sociales como *Instagram* o *Twitter*, usando diferentes criterios de filtro como: menciones a personajes públicos, palabras clave, población objetivo, y eventos de interés nacional (*hashtags*).
- (iii) Definir criterios afines a estándares internacionales que permitan anotar los datos obtenidos usando anotadores humanos.
- (iv) Publicar una base de datos tratada y anotada que permita entrenar fácilmente métodos de aprendizaje supervisado para clasificar automáticamente potenciales discursos de odio en redes sociales chilenas (dialecto chileno del español).
- (v) Seleccionar, entrenar y evaluar al menos un modelo del estado del arte que permita contar con un nivel de referencia (*baseline*) para futuras investigaciones.

(vi) Publicar y divulgar los resultados de esta investigación.

1.2. Impacto inicial de solucionar el problema

Este proyecto busca contribuir generando un nuevo corpus de datos en el dialecto chileno del español mediante el diseño y ejecución de una metodología de recolección y anotado de datos, que permita poner a disposición de la comunidad académica nuevos recursos que fomenten el desarrollo de estudios sobre discursos de odio adaptados a nuestra realidad nacional. Junto al corpus de datos tratados y anotados por humanos, el proyecto propone la selección, implementación y evaluación de tres modelos del estado del arte que permitan contar con puntos de referencia (*baselines*) para futuras investigaciones.

CAPÍTULO 2

MARCO CONCEPTUAL

A continuación, se procederá a describir el marco conceptual que actuará como fundamento teórico para la utilización y descripción de las metodologías y herramientas utilizadas. Este se encuentra compuesto por los siguientes ejes:

- Trabajo Relacionado en Metodologías de Muestreo
- Trabajo Relacionado en Metodologías de Anotación
- Trabajo Relacionado en Detección de *Hate Speech*
- Métricas

2.1. Trabajo Relacionado en Metodologías de Muestreo

2.1.1. Descripción

Una metodología de muestreo constituye un conjunto de instrucciones las cuales definen el tipo de herramientas y técnicas utilizadas para llevar a cabo la obtención de los datos requeridos por una investigación. En esta se definen qué tipos de datos se desean recolectar, y como se realizará la captura. En estudios basados en redes sociales, las estrategias de recolección usualmente involucran *keywords*, *hashtags*, eventos de interés nacional, personajes públicos, grupos objetivo, etc., señalando como éstos se encuentran relacionados con los objetivos de la investigación, además de qué herramientas y/o técnicas de *webscrapping* son utilizadas.

2.1.2. Estado del Arte

Debido a la carencia actual de metodologías de recolección formales, se investigó la literatura relacionada con el tema y se documentó qué decisiones se tomaron en la recolección de estos datos. A partir de esto, se confecciona la siguiente descripción publicación a publicación, para así poder determinar y formalizar la metodología a utilizar:

- En Suryawanshi *et al.*, se recolectaron datos mediante el uso de 16 voluntarios, los cuales donaron *memes* encontrados en sus redes sociales (*Instagram*, *WhatsApp*, *Pinterest*, *Facebook*) a lo largo de dos meses. Se utilizaron dos categorías: *troll* y *no-troll*.

La categoría *troll* se define como una imagen o *meme* que intenta ofender o disminuir a una persona. No se establecen filtros o criterios sobre el tipo de ofensa o comunidad ofendida.

- En Albadi *et al.*, se concentraron en los 4 grupos religiosos más populares en el oriente medio (islam, cristianismo, judaísmo y ateísmo), debido a que en esta zona es donde el islam es mayormente practicado. Los autores decidieron agregar las 2 mayores sectas musulmanas, *Sunni* y *Shia* respectivamente. Se utilizó la API de búsqueda de *Twitter* para recolectar 6000 *tweets*, 1000 por cada religión. Debido a la ausencia de un léxico de odio y para evitar una recolección sesgada, se optó por integrar en el *query* de búsqueda solo términos imparciales que se refieran a cada religión y sus practicantes. Intencionalmente, evitaron usar cualquier clase de insulto asociado a las religiones como *keyword* de búsqueda. Con tal de reducir la redundancia en el *dataset* de entrenamiento, se desestimaron *retweets* y *replies* de los *queries* con tal de facilitar la tarea de anotación. Para el *dataset* de testeo se realizó la misma acción anterior, sin embargo, se recolectaron solo 600 *tweets*, 100 por cada religión, 2 meses después de obtenidos los *tweets* relacionados al conjunto de entrenamiento, con tal de asegurar que los clasificadores puedan generalizar bien nuevos datos.
- En Alfina *et al.* utilizan la API de *Twitter* para recopilar datos asociados a la elección de gobernador de *Jakarta*, utilizando *hashtags* y *keywords* como “*#DebatPilkadaDKI*”, “*#SidangAhok*”, “*Pilkada Jakarta 2017*”. Finalmente, tras un filtrado se obtuvieron 1100 *tweets* para ser anotados.
- En Alatawi *et al.* primero se examinó, un foro de discusión sobre supremacía blanca para construir un diccionario. Luego utilizaron la API de *Twitter* para obtener *tweets* asociados a supremacistas blancos. La metodología utilizada fue la siguiente: en primera instancia se recolectaron *tweets* con *hashtags* asociados a la supremacía blanca. Luego, a partir de las cuentas que más utilizaban estos *hashtags*, se realizó una recopilación de *tweets* sobre estas cuentas. Los *tweets* recolectados se unieron al *stormfront dataset* [17], ya existente.
- En Mozafari *et al.* utilizaron 2 *datasets* generados por otros trabajos:
 - El primer *dataset* de Waseem y Hovy se confeccionó en un periodo de 2 meses y dio como resultado 136,052 *tweets* de *Twitter* de los cuales se anotaron 16,914 (3,383 sexistas, 1,972 racistas y 11,559 ni sexistas ni racistas). Se construyó manualmente un vocabulario de términos ofensivos con el cual se hizo una primera búsqueda. Después de analizar esos *tweets*, se determinaron manualmente términos inocuos, usualmente presentes en discursos de odio. Luego, para completar el *dataset* se utilizó la API de *Twitter* para obtener *tweets* usando el diccionario con palabras ofensivas e inocuas.
 - El segundo *dataset* de Davidson *et al.*, se recolectó utilizando el diccionario predefinido de *HateBase* en conjunto con la API de *Twitter*. Se buscó la ocurrencia de estas palabras en *tweets* obteniéndose 33,458 cuentas/usuarios que utilizaron estas *keywords*. Luego extrajeron todos los *tweets* emitidos por cada uno de

estos usuarios, resultando en un conjunto de $\sim 85,4$ millones de *tweets*, sobre el cual se aplicó un muestreo aleatorio para seleccionar 25,000 *tweets* que contenían palabras ofensivas.

- En Ibrohim y Budi, utilizaron *tweepy* para buscar *tweets* que contengan palabras y frases utilizadas en discursos de odio en redes sociales indonesias a lo largo de 7 meses. Se identificaron las siguientes 4 categorías: “no hate speech”, “no hate speech but abusive”, “hate speech but no abusive” y “hate speech and abusive”. Para determinar qué exactamente constituye discursos de odio, realizaron un *Focus Group* interno determinando las siguientes 5 temáticas que suelen ser objetivo de *hate speech*:

- Religion/creed
- Race/ethnicity
- Physical/disability
- Gender/sexual orientation
- Other invective/slander

El problema se plantea como multiclase a menos que se trate del caso *Other invective/slander*: un *tweet* no puede pertenecer a esta clase y a otras simultáneamente.

- En Khan *et al.*, se utilizó la herramienta *tweepy* con *keywords* de búsqueda como política, protesta pública, disturbios, tópicos populares, localización geográfica y otros *hashtags* infames, llegándose a obtener 90,000 *tweets* recolectados en un periodo de 3 meses en India y Pakistan.
- En Romim *et al.*, se recolectaron comentarios de *TikTok*, *YouTube* y *Facebook*. En *Facebook*, se utilizó la *Graph API* para extraer comentarios de páginas de celebridades, pero por sus limitaciones no se pudo obtener una gran cantidad de datos relevantes. En *Youtube*, buscaron vídeos en el periodo 2017-2020 correspondientes a los tópicos más controvertidos de Bangladesh, seleccionando vídeos con al menos 700,000 vistas y se extrajeron los comentarios de ellos. Se terminó utilizando la aplicación *Facepager* para extraer comentarios de *Facebook*.
- En Mathur *et al.*, utilizaron la *API* de *Twitter* para seleccionar *tweets* con al menos 3 palabras en *Hinglish* (mezcla del hindú con inglés), los cuales fueron recolectados a lo largo de 4 meses utilizando filtros de geolocalización y *hashtags* populares. Se obtuvieron 25,667 *tweets* para la tarea de anotado.
- En Samarasinghe *et al.*, recurrieron a sitios de *prensa amarilla* para hacer *webscraping* y tras hacer un limpiado de datos se obtuvieron unos 7000 comentarios. Lamentablemente, los autores no reportan los criterios utilizados para la recolección ni tampoco cómo seleccionaron las palabras utilizadas para determinar la presencia de odio.
- En Sabat *et al.*, los autores utilizaron *Google images download* recolectando 5,020 imágenes y utilizándolas en conjunto con *Reddit Memes Dataset* para tener *memes* que no

sean de odio. Para recolectar los *memes* de odio se utilizaron las siguientes *keywords*: *racist meme*, *jew meme* y *muslim meme*. Para las clases asumieron estas de acuerdo al *query* utilizado, es decir, si el meme era encontrado usando la *keyword*: *racist meme*, entonces el meme queda clasificado como *racist* en el *dataset*.

- En Suryawanshi *et al.*, los obtuvieron de *kaggle*, pero en dicha plataforma no indican cómo se generó el *dataset*.
- En Alotaibi y Hasanat, utilizaron *Twython* y *Tweepy* para obtener los *tweets* a partir de una lista de 60 palabras racistas.

Basándose en esto se pueden determinar las siguientes necesidades que nuestra metodología de muestreo deberá suplir. Primero, definir palabras claves pertinentes al contexto del problema, es decir, que éstas estén ligadas no solo a contextos meramente discriminatorios, sino que también permitan identificar contenido ofensivo y neutral, con tal de permitirle generalizar bien al modelo, siendo así capaces de obtener predicciones de mayor fidelidad. Segundo, es necesario definir claramente qué herramienta se utilizará, indicando ventajas y desventajas con respecto a otras herramientas actuales. Tercero, se debe definir cómo se generará el diccionario de palabras claves y en base a qué criterios. Finalmente, se deben especificar los tiempos de muestreo y qué conceptos serán utilizados.

A modo de resumen, lo que debe realizar nuestra metodología de muestreo es lo siguiente:

- Definir criterios y conceptos claves del problema, es decir, conceptos de discurso de odio, lenguajes ofensivos y qué técnicas teóricas utilizan con tal de poder contrastar fácilmente con otras publicaciones.
- Definir diccionario de palabras ofensivas generalizables, es decir, que puedan ser encontradas en contextos discursivos normales o de odio con similar frecuencia. Se deben documentar los criterios para la inclusión de las palabras.
- Definir herramientas de muestreo indicando las razones para su elección, además de ventajas y desventajas.
- Indicar tiempos de muestreo señalando criterios para su definición
- Definir criterios para el reconocimiento de comunidades identificadas.

2.1.3. Herramientas de Muestreo

En esta sección se detallan algunas de las herramientas utilizadas en la literatura, al igual que otras herramientas que pueden llegar a ser relevantes para la realización del muestreo de datos. La Tabla 1 incluye referencias y detalles técnicos.

- *Beautifulsoup*: Es una librería de *Python* la cual permite la extracción de datos de archivos, *htmls* y *xml*. Generando una estructura de datos anidada, la cual representa el árbol gramatical del archivo y el cual se puede manipular como si tuviera una estructura de diccionario.
- *Facepy*: Permite interactuar de forma programática con la *API Graph* de *Facebook*. La *API Graph* es la principal herramienta que permite a las apps leer y escribir en la red social de *Facebook*, permitiendo así la extracción de datos relevantes dentro de la aplicación, como personas, grupos, publicaciones, imágenes y vídeos.
- *Google Images Download: Script* de *Python* que permite buscar y descargar miles de imágenes de Google imágenes en el disco local, usando *keywords* para la búsqueda.
- *Instagramy*: Extrae información de usuario, *Posts*, *Hashtags* y datos de localización en *Instagram*.
- *Meme API: JSON API* permite obtener N memes de *Reddit*, pudiendo especificar el *subreddit*.
- *Reddit API*: Permite obtener información sobre la red social de *reddit*, permitiendo recuperar *posts* en *reddit*, incluyendo contenido (urls de las imágenes/vídeos mostrados o texto), comentarios y puntaje (*upvotes/downvotes*).
- *Request*: Permite enviar *requests* de *HTTP/1.1* y obtener los *htmls* de los *posts* buscados, por lo cual teóricamente es posible ingresar a cualquier red social a través de un *script* y extraer datos de ella. Sin embargo, esto conlleva mucho tiempo debido a que los servidores poseen un límite máximo de *requests* por minuto, ralentizando la extracción de datos.
- *Selenium*: Es un proyecto general para una gama de herramientas y librerías que permiten la automatización de procesos en navegadores *web*, pudiendo estos ir desde la extracción de datos a la automatización de procesos en un navegador.
- *Tweepy*: Permite acceso a la *API* de *Twitter* mediante *Python*.
- *Twitter API*: Permite extraer *tweets* por *hashtag*, persona, localización.
- *Twython*: Permite acceso a la *API* de *Twitter* mediante *Python*.

Herramienta	Red Social	Documentación	Repositorio
<i>Beautifulsoup</i>	<i>Facebook, Twitter</i>	Link Docs	Link Repo
<i>Facepy</i>	<i>Facebook</i>	Link Docs	Link Repo
<i>Google Images Download</i>	<i>Todas²</i>	Link Docs	Link Repo
<i>Instagramy</i>	<i>Instagram</i>		Link Repo
<i>Meme API</i>	<i>reddit</i>		Link Repo
<i>Reddit API</i>	<i>reddit</i>	Link Docs	
<i>Request</i>	<i>Facebook, Twitter</i>	Link Docs	Link Repo
<i>Selenium</i>	<i>Facebook, Twitter</i>	Link Docs	Link Repo
<i>Twitter API</i>	<i>Twitter</i>	Link Docs	
<i>Tweepy</i>	<i>Twitter</i>	Link Docs	Link Repo
<i>Twython</i>	<i>Twitter</i>	Link Docs	Link Repo

Tabla 1: Herramientas de *web scrapping*
Fuente: Elaboración propia

2.2. Trabajo Relacionado en Metodologías de Anotación

2.2.1. Descripción

Una metodología de anotación constituye un conjunto de instrucciones que definen el conjunto de herramientas y técnicas a utilizar para llevar a cabo la anotación de datos por humanos. Normalmente, se busca disminuir el sesgo entre anotadores y obtener un conjunto de datos anotados lo más fiel posible a la realidad del fenómeno estudiado. La metodología define también las categorías en que se quieren clasificar los datos, el tipo de plataforma o herramienta que se usará para automatizar la tarea (si es que es utilizada), qué instrucciones se les entregará a los anotadores, en qué conceptos deben basarse para generar la clasificación (*guidelines*), y qué técnicas se utilizarán para disminuir el sesgo de anotación.

2.2.2. Estado del Arte

Al igual que en el caso de las metodologías de muestreo, existe poca o baja formalización sobre cómo llevar a cabo la anotación de los datos, dificultando así la reutilización de estas técnicas. Por lo anterior, resulta de especial relevancia determinar cuáles son buenas y malas prácticas a la hora de anotar, para así poder diferenciar entre ellas y obtener un anotado de buena calidad. Con este objetivo, se confeccionó la siguiente lista de literatura relevante que

²imágenes en *Google*: es posible añadir de *keyword* el nombre de la red social de la cual queremos extraer imágenes.

detalla las metodologías utilizadas, para rescatar buenas y malas prácticas de anotación, y, con esto, poder generar una metodología de anotación formalizada:

- Suryawanshi *et al.* utilizó *Google Forms* para mostrar los datos a los anotadores, presentándose 10 memes por página y un total de 100 memes por formulario. Para cada meme los anotadores debían decidir si era *troll* o no. A modo de entrenamiento, a los anotadores se les mostró una serie de ejemplos de memes *troll* y no *troll*. Cada meme fue asignado a una anotadora femenina y a uno masculino. El *meme* es anotado como perteneciente a una categoría si, y solo si, ambos anotadores llegaron a un acuerdo, es decir, ambos determinan la misma categoría. No se menciona el número total de anotadores.
- Albadi *et al.* utilizaron *CrowdFlower*, plataforma de *crowdsourcing* para realizar el anotado. Solo se le permitió acceder a personas con IP's asociadas a países de habla árabe. Se entregó la definición de discursos de odio utilizada por la publicación. Con el fin de reducir el sesgo, se le pidió a cada anotador ser lo más imparcial posible en sus anotaciones. Se les pidió clasificar los *tweets* en alguna de las 3 siguientes categorías: contiene discurso de odio, no contiene discurso de odio y poco claro. En caso de contener, se les pidió a los anotadores que distinguieran a cuáles comunidades religiosas se encontraban dirigidos los ataques.

Con tal de asegurar anotaciones de buena calidad, se realizaron pruebas a los anotadores, siendo aceptados solo aquellos que puntuaran al menos 70 % en un proceso de evaluación preliminar. Además, se añadió una pregunta por página de trabajo con lo cual quienes fallaran en mantener un 70 % de desempeño a lo largo de la tarea eran descalificados y expulsados de ésta. La pregunta de validación correspondía a un *tweet* previamente anotado, indistinguible de los otros 4, puesto que eran 5 *tweets* por página de trabajo.

- En Alfina *et al.*, se utilizaron 2 categorías (discurso de odio o no discurso de odio). Los anotadores fueron 30 estudiantes universitarios voluntarios de 17 a 24 años (43 % hombres). Se describe la composición étnica y religiosa de los voluntarios. Se prepararon conjuntos de 50 *tweets* para ser anotados por 3 anotadores de diferente sexo, etnia y religión. Si los 3 anotadores hacían la misma clasificación, se dejaba la anotación, en caso contrario se eliminaba.
- En Alatawi *et al.*, se anotaron 1,999 *tweet* con 3 anotadores obtenidos de *Amazon Mechanical Turk (AMT)*. Se consideraron 4 categorías: *explicit white supremacy (EWS)*, *implicit white supremacy (IWS)*, *other hate speech (O)*, y *neutral(N)*. En este trabajo utilizaron el Kappa de Cohen como métrica de medición de acuerdo entre anotadores. Los fundamentos teóricos de esta y otras métricas se encuentran detallados más adelante, en la sección 2.4.2. Obtuvieron un Kappa de Cohen de 0,070 (muy bajo), debido principalmente al poco acuerdo logrado entre los anotadores frente a las categorías. Estos desacuerdos fueron medidos contando el número de conflictos entre pares de

anotadores, descubriéndose que, en general, esto ocurría cuando la intención del autor no era explícita. Como resultado, las cuatro categorías originales se colapsaron en *white supremacist* y *not white supremacist*. Finalmente, como método de resolución de desacuerdos, se optó por mantener la categoría más votada como categoría final.

- Waseem y Hovy generaron la siguiente lista de criterios para clasificar un *tweet* ofensivo (basado en McIntosh y Plous):

1. Usa un insulto sexista o racial.
2. Ataca a una minoría.
3. Busca silenciar a una minoría.
4. Critica a una minoría (sin argumentos claros).
5. Promueve la discriminación, pero no usa directamente el discurso de odio o los crímenes violentos.
6. Critica a una minoría y utiliza falacia del hombre de paja [48].
7. Tergiversa descaradamente la verdad o busca distorsionar las opiniones sobre una minoría con afirmaciones infundadas.
8. Muestra apoyo a hashtags problemáticos. e.g. “#BanIslam”, “#whoriental”, “#whitengenocide”.
9. Estereotipa negativamente a una minoría.
10. Defiende xenofobia o el sexismo.
11. Contiene un nombre de pantalla (*screen name*) que es ofensivo, según los criterios anteriores. El *tweet* es ambiguo (en el mejor de los casos) y además se refiere a un tema que satisface cualquiera de los criterios anteriores.

- Davidson *et al.* utilizaron *CrowdFlower* para la anotación, determinando 3 categorías: ofensivo, discurso de odio, y ninguno de los anteriores, utilizando la definición de discurso de odio mencionada en la publicación. Se les pidió a los anotadores fijarse en el contexto del *tweet* para determinar su categoría. Se usaron al menos 3 anotadores por *tweet* y se logró un Kappa de Cohen de 92 %. Se anotaron 24,802 *tweets* de los cuales un 76 % era ofensivo.

- Ibrohim y Budi utilizaron *crowdsourcing* (no señalan qué plataforma), separando el proceso de anotación en dos fases.

En la primera fase se determinó si los *tweets* correspondían a discurso de odio, lenguaje abusivo o no, en concordancia con las definiciones entregadas en la publicación.

En la segunda fase, se les pidió a los anotadores definir el objetivo (*individual* y *group*), categoría (*Religion/creed*, *Race/ethnicity*, *Physical/disability*, *Gender/sexual orientation* y *Other invective/slander*) y nivel del *tweet* (*Weak hate speech*, *Moderate hate speech* y *Strong hate speech*).

Para el anotado, generaron un sistema *web* de anotación propio. Con el fin de facilitar la tarea a los anotadores, se generaron “*guidelines*”(no documentadas) para la tarea. Luego, hicieron una prueba de anotación para verificar qué anotadores comprendieron la tarea y cuáles no. Utilizaron a un lingüista como experto para consultar y discutir las metodologías y resultados. En la selección de anotadores, se aseguraron de que vinieran de distintas esferas religiosas, étnicas y sociales. Se utilizaron además los siguientes criterios de selección:

- 20 – 30 años
- Nivel nativo de la lengua de Indonesia
- Experimentado en el uso de *Twitter*
- No ser miembro de ningún partido político

El proceso de reclutamiento seleccionó 30 anotadores, 14 hombres y 16 mujeres, 25 estaban en el rango 20 – 24 años y 5 en el rango 25 – 32 años. Se fijó un nivel mínimo de “*agreement*” o si no el texto era eliminado. Posteriormente, el *label* final se determinó usando mayoría de votos.

- Khan *et al.* utilizan una metodología iterativa para la generación de los *guidelines* de anotación partiendo con los *guidelines* originales sobre los que se generarán los nuevos. Para generar esta metodología, se seleccionó un 10 % de los *tweets* y se les entregaron a dos anotadores los *tweets* para anotar, a los cuales se les solicitó desarrollar “*guidelines*” que facilitarían la tarea. Luego se realizó el proceso nuevamente con otros anotadores, los cuales generaron nuevos *guidelines*, los cuales fueron combinados con el anterior, tras lo cual determinaron el estadístico Kappa para evaluar la consistencia de las anotaciones, lo que fue repetido un par de veces (no mencionan cuántas). Tomando esto en cuenta, fueron capaces de generar la taxonomía de clasificación mostrada en la figura 1, desarrollando “*guidelines*” para detectar cada clase.

Para la segunda ronda de anotado, se tomó el 25 % del 10 % anterior con el fin de evaluar la efectividad de los “*guidelines*” generados. Estas definiciones junto con los datos seleccionados fueron entregados a otros 2 anotadores, los cuales alcanzaron un 91 % para el estadístico Kappa de las anotaciones. Para el proceso de anotado final, se entregaron los “*guidelines*” validados a otro grupo de 2 anotadores (no explicitan si es un solo grupo) y se les indicó a qué “*guideline*” debían referirse en caso de considerar que el *tweet* se relacionara con ese elemento de la taxonomía.

- En Sharma *et al.*, se recurrió a *Amazon Mechanical Turk* (AMT) para anotar si las imágenes pertenecían a alguna de las siguientes categorías: *Humorous*, *Sarcasm*, *Offensive*, *Motivation* y para cuantificar la intensidad de la clase indicada. Además, se solicitó indicar la clase de sentimiento al cual estaba relacionada la imagen (*very negative*, *negative*, *neutral*, *positive*, *very positive*). Para lograr todo esto, se identificaron las tareas que debiese ser capaz de realizar el modelo:
 - Clasificación de sentimientos: dado un *meme*, la primera tarea es clasificarlo como positivo, negativo o neutral.

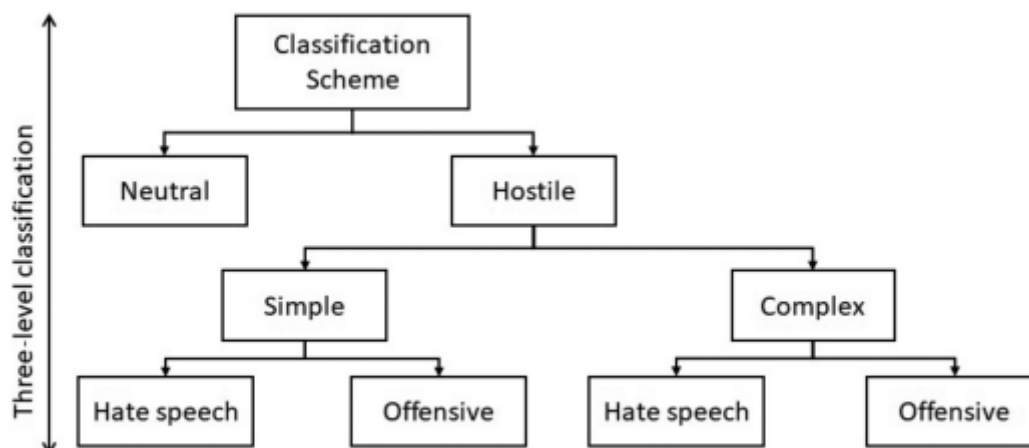


Fig. 3. Taxonomy of the offensive sentences.

Figura 1: Taxonomía de Khan *et al.*
Fuente: Khan *et al.*

	sarcastic	humorous	offensive	Motivation
<i>not</i> (0)	😐	😐	😐	😐
<i>slightly</i> (1)	😊	😄	😏	👍
<i>mildly</i> (2)	😏	😄	😏	NA
<i>very</i> (3)	😏	😄	😏	NA

Table 1: Semantic classes for the Memotion Analysis

Figura 2: Clases semánticas de Sharma *et al.*
Fuente: Sharma *et al.*

- Clasificación de humor: dado un *meme*, el sistema debe identificar el tipo de emoción expresada. Los *memes* pueden poseer más de una categoría.
 - Escala semántica: la tercera tarea consiste en cuantificar qué tanto se expresa cada clase (i.e. fuertemente sarcástico o no ofensivo). A continuación, se muestra la figura 2 qué detalla la escala de cuantificación.
- Kiela *et al.* documentaron una definición de discursos de odio que luego fue entregada a una compañía encargada de la anotación, siendo los anotadores entrenados por 4 horas para reconocer discursos de odio según la definición especificada. Se llevaron a cabo tres pilotos con los anotadores con el fin de generar *feedback* para mejorar su desempeño. El proceso fue dividido en 3 fases:
1. Filtrado: en esta etapa los anotadores se encargaron de filtrar las imágenes de acuerdo con el idioma, verificando que cumplieran con no tener contenido ex-

plícito (contenido suicida, explotación infantil, desnudez, llamados a la violencia, contenido sexual para adultos, invitación a actos de terrorismo y trata de personas).

2. Construcción de *memes*: se les pidió a los anotadores cambiar la imagen de los *memes* por una que compartiera el mismo significado, pero que fuera de uso liberado.
3. Puntaje de odio: se les pidió a los anotadores que anotaran la imagen en una escala de 1 – 3 donde 1 indica que el caso es definitivamente de odio, 2 indica que es ambiguo, y 3 indica que definitivamente no se trata de odio. Para cada *meme*, se recolectaron 5 anotaciones.
4. Confusores benignos (*benign confusers*): Para el subconjunto de *memes* anotados como de odio, se recolectaron confusores. Esto se hizo con el fin de agregar sesgos que el modelo fuera capaz de aprender con facilidad. Los confusores se definieron como “imagen o texto mínimo de reemplazo capaz de invertir el significado del *meme* de odio a no odio y viceversa” [30].

Los anotadores alcanzaron un estadístico Kappa de 68,4 %, lo cual indica un acuerdo moderado según McHugh.

- Kehat y Pustejovsky detallan un conjunto de reglas y buenas prácticas a seguir para generar el anotado, tales como: *author bias*, *Limited or sparse vocabulary*, *Action/visual sense is not well defined*, y *Annotator’s Attention*.
- Oriola y Kotzé redactaron las siguientes reglas para la anotación (no explicitan cuántas de las siguientes debe cumplir):
 1. Un *tweet* es de odio si:
 - Está dirigido contra una persona o grupo de personas.
 - Usa palabras despectivas o insultos raciales repetitivamente dentro del *tweet*.
 - Hace uso de términos despectivos con la intención de dañar o incitar al daño.
 - Hace referencia y apoya otros hechos de odio, *tweets* de odio y organización.
 - Hace uso de expresiones idiomáticas, metafóricas, de colocación o cualquier otro medio indirecto que resulte perjudicial o pueda incitar al daño.
 - Expresa, comunicaciones violentas.
 2. Un *tweet* es ofensivo si:
 - Está dirigido contra una persona, grupo de personas u organización.
 - No es discurso de odio.
 - Abusa de un objetivo utilizando palabras soeces, despectivas o insultantes.
 3. Un *tweet* está acorde a la libertad de expresión si:
 - Está dirigido o no contra una persona, grupo de personas u organización.

- No es discurso de odio ni discurso ofensivo.
- Es expresado por el gobierno o una agencia autorizada de gobierno con la intención de una promoción masiva e iluminación (“*enlightment*” [45]).

En el proceso de anotación utilizaron 6 anotadores divididos en 3 grupos de 2 anotadores, asignando cada grupo a “inglés con Afrikaans”, “inglés con IsiZulu” y “inglés con Sesotho”, respectivamente. Los anotadores alcanzaron estadísticos Kappa de 0,837, 0,579 y 0,633 para los datos en *Afrikaans*, *IsiZulu* y *Sesotho*, respectivamente. Debido al bajo nivel de acuerdos logrado en los últimos dos *datasets*, se determinó utilizar solo los tweets en que ambos anotadores lograron acuerdos.

- Romim *et al.* utilizan 50 anotadores a los cuales les indicaron seguir las reglas definidas en el documento para llevar a cabo la anotación. La tarea de anotación presenta las siguientes clases, discurso de odio, no discurso de odio, además de las siguientes categorías: *sports*, *entertainment*, *religion*, *politics*, *crime*, *celebrity* y *TikTok & meme* (estas últimas se anotaban independiente de si fuese discurso de odio o no). Todos los comentarios fueron anotados en tres ocasiones donde se determinaba la clase por mayoría de votos. Finalmente, realizaron una extracción aleatoria de 30 comentarios pertenecientes a cada categoría donde el autor verificó manualmente la validez de estas clasificaciones. No reportan *agreement*.
- Suryawanshi *et al.* documentan las reglas a utilizar por los anotadores para la tarea. Se utilizaron 8 anotadores voluntarios, 6 hombres y 2 mujeres con tal de reducir el sesgo de género. El proceso de anotación consistía de 2 pasos, primero se les entregaban 50 *memes* a cada uno de los anotadores para clasificar como ofensivo o no y la categoría más votada era la elegida. Se realizaron cambios a la metodología del proceso de acuerdo a la retroalimentación de los anotadores y se repitió el paso 1, con lo cual se obtuvo un kappa de Cohen moderado (entre 0,4 a 0,5) y se tomó la decisión de enviar todos los *memes* restantes a los anotadores, donde cada *meme* fue anotado solo por un anotador.
- Samarasinghe *et al.* realizaron el anotado en dos fases a través de *crowdsourcing* (no reportan herramienta o plataforma utilizada). En la primera fase se determina si el texto constituye discurso de odio y en la segunda fase se determina el nivel de odio, pudiendo ser desde el 1 al 4 dónde los niveles corresponden a:
 1. Plantea preocupaciones en términos de intolerancia.
 2. Discurso de odio que está prohibido.
 3. Constituye incitación a discriminación, hostilidad o violencia. Alguna idea que promuevan el odio racial o lo justifiquen ideas.
 4. Incitación directa y pública al genocidio.
- En Alotaibi y Hasanat no describen método de anotación. Describen que trabajan con un grupo de individuos, sin embargo, no describen cuántos anotadores son en total, cuántos por *tweet*, ni cómo validan el proceso.

- En Mathur *et al.*, el anotado fue realizado por 3 anotadores con conocimientos de NLP. Anotaron como *hate speech* si los *tweets* satisfacían una o más de las siguientes condiciones:

1. Utiliza un insulto (*slur*) sexista, racista o discriminador hacia una minoría.
2. Estereotipos indignos.
3. Apoyando hashtags problemáticos como *#ReligiousSc*m*.

No reportan reglas para las otras clases. En 2803 *tweets* las clases fueron determinadas por mayoría de votos, los 386 restantes fueron anotados por un experto. Finalmente, lograron anotar 3189 *tweets* en las clases de *hate-inducing*, *abusive*, *non-offensive*.

En general, notamos que las características principales que debe cumplir un método de anotación son las siguientes:

- Definir *guidelines* claros y sencillos para los anotadores.
- Idealmente, utilizar al menos 3 anotadores con tal de evitar empates.
- Utilizar test ocultos es una buena práctica para asegurar el comportamiento de los anotadores.
- Utilizar una herramienta web adecuada, cumpliendo con criterios de facilidad de uso y relevancia en el problema.
- Definir clases o categorías claras.
- Utilizar una métrica de medición de *agreement* adecuada. Kappa de Cohen si son al menos 2 personas y Alfa de Krippendorff en caso de ser más.

2.2.3. Plataformas de *Crowdsourcing*

Una herramienta de anotación constituye una plataforma web de trabajo colaborativo donde distintos etiquetadores trabajan en equipo para completar una tarea de anotación. Estas tareas pueden ir desde clasificación de textos e imágenes, hasta reconocimiento de entidades en textos, imágenes o videos mediante el uso de técnicas de etiquetado de palabras y selección de polígonos, respectivamente. A partir de las plataformas indicadas por la literatura, se determinó la siguiente lista de herramientas de anotación utilizadas:

- *Crowdfower*³: Actualmente, es pagada y no posee licencia académica.

³Actualmente *Appen*.

AUTOMÁTICO

Publicación	Tipo de Dato	Metodología de Recolección	Esquema de Clases	Metodología de Anotación	Anotadores por Dato	Validación
Suryawanshi et al., 2020b	imagen	Voluntarios	troll	Entrenamiento y clasificación	2	Acuerdo total entre anotadores
Albadi et al. 2018	texto	Palabras clave	Discurso de odio, poco claro y comunidades	Entrenamiento y pruebas secretas en clasificación	No reportan	Acuerdo con datos ya anotados
Alfina et al. 2017	texto	Palabras clave	Discurso de Odio	Introducción, clasificación	3	Acuerdo total entre anotadores
Alatawi et al. 2021	texto	Palabras clave	explicit white supremacy, implicit white supremacy, other hate speech y neutral	Introducción, clasificación	3	Más votado
Mozafari et al. 2020	texto	Datos pre-existentes	"no hate speech", "no hate speech but abusive", "hate speech but no abusive" y "hate speech and abusive" y comunidades	No utilizan	No utilizan	No utilizan
Waseem y Hovy 2016	texto	Palabras clave	Ofensivo	Anotado por autores	No reportan	Verificación por colaboradores
Davidson et al. 2017	texto	Palabras clave + muestreo de Cuentas	Ofensivo, Discurso de odio, y ninguno de los anteriores	Introducción y clasificación	3	más votado
Ibrohim y Budi 2019	texto	Palabras clave	Discurso de odio, lenguaje abusivo, objetivo, categoría y nivel de odio en el tweet	Introducción y dos fases de clasificación	3	más votado
Khan et al. 2021	texto	Palabras clave	Discurso de odio, ofensivo, neutral	Entrenamiento, 2 rondas de clasificación	2	No reportan
Romim et al. 2021	texto	Palabras clave	discurso de odio y categoría	Introducción, clasificación	3	más votado
Mathur et al. 2018	texto	Palabras clave	non-offensive, abusive y hate-inducing	clasificación (no reportan introducción)	3	más votado + anotador experto
Samarasinghe et al. 2020	texto	No reportan	discurso de odio, categorizado en niveles del 1 al 4 (menos a más grave)	2 rondas de clasificación	No reportan	No reportan
Sabat et al. 2019	imagen	Palabras clave	discurso de odio	No utilizan	No utilizan	No utilizan
Suryawanshi et al. 2020a	imagen	No reportan	ofensivo	Entrenamiento y clasificación	1	acuerdo en entrenamiento
Alotaibi y Hasanat 2020	texto	Palabras clave	racista	No reportan	No reportan	No reportan
Sharma et al. 2020	imagen	Palabras clave	sarcástico, humor, ofensivo y motivación	Clasificación (sin guilines)	5	más votado
Kiela et al. 2020	imagen	No reportan	rating de odio	Entrenamiento, clasificación	5	más votado + anotador experto para desacuerdos

Tabla 2: Tabla resumen metodologías de recolección y anotado por publicación

Fuente: Elaboración propia

- *Amazon Mechanical Turk*: Pagada y sin beneficios para académicos.

En base a la inspección del estado del arte, se generó la siguiente tabla de resumen de la literatura relevante con autores, tipo de dato de la publicación, metodología de recolección, esquema de clases, metodología de anotado, anotadores por dato y como realizan la validación del anotado (ver tabla 2).

2.3. Trabajo Relacionado en Detección de *Hate Speech*

En esta sección se presentarán en el siguiente orden las bases conceptuales de *NLP*, aprendizaje automático y profundo, para finalmente exponer 3 modelos del estado del arte sobre los cuales se evaluará posteriormente su desempeño en el corpus generado.

2.3.1. Procesamiento de Lenguaje Natural o *NLP*

El procesamiento de lenguaje natural representa una intersección entre los campos de la lingüística, informática e inteligencia artificial, donde el principal objeto de estudio corresponde al lenguaje humano mediante técnicas relacionadas a la informática e inteligencia artificial. En particular, se trata de ser capaces de analizar grandes cantidades de datos mediante el uso de computadoras. La principal meta de esta área consiste en generar un programa que sea capaz de comprender en cierta medida los contenidos de documentos y textos, incluyendo la diversidad de contextos presentes.

Hoy en día se ha demostrado que, en general, métodos de aprendizaje automático basados en redes neuronales profundas pueden lograr resultados del estado del arte [18], por lo cual uno de los objetivos de este trabajo será la obtención de un *baseline* de desempeño para el *dataset* a recolectar usando este tipo de técnicas.

Para atacar problemas de *NLP* con redes neuronales se suele recurrir a redes de tipo *Transformers*, debido a que las características intrínsecas de este tipo de arquitectura permiten reconocer patrones en secuencias, comprensión del contexto mediante los mecanismos de atención, y altas capacidades de paralelización.

Tareas del Procesamiento de Lenguaje Natural El lenguaje humano como fenómeno se encuentra caracterizado por el uso de simbologías con tal de transmitir un mensaje desde el emisor hacia el receptor. Esta simbología se encuentra principalmente conformada por palabras. Sin embargo, una parte importante de la comunicación ocurre fuera de lo textual o de lo evidente, es decir, existen elementos de los mensajes emitidos, como la entonación, la gesticulación, el uso de referencias o contextos históricos y socioeconómicos, etc., que nos permiten discernir no solo qué nos están tratando de comunicar, sino también inferir

las intenciones del emisor. Por estas razones, al ser el lenguaje un sistema de comunicación en el cual las fuentes de información no son solo verbales y obedecer fuertemente a un contexto social cambiante, el generar algoritmos capaces de determinar el significado de textos o voces ha resultado extremadamente complejo.

La existencia de “homonimias”, sarcasmo y metáforas, entre otras, son algunas de las tantas irregularidades del lenguaje que les toma a los humanos años aprender. Sin embargo, si se quiere generar algoritmos orientados al procesamiento de lenguaje natural, el ser capaces de comprender correctamente estas irregularidades es un requisito inicial indispensable si se busca generar aplicaciones útiles y prácticas.

Existen diversas tareas en *NLP* orientadas a digerir datos textuales y de sonido con tal de ayudar al computador a comprenderlos. Algunas de las tareas asociadas a *NLP* con mayor relevancia para el presente texto son:

- **Desambiguación de palabras:** Corresponde a la selección de significados de una palabra mediante un proceso de análisis semántico con el cual se determina el significado adecuado de la palabra en relación al contexto. Por ejemplo, “sentido” puede referirse tanto al verbo sentir como a la cualidad de dirección u orientación. Por lo tanto, el uso de desambiguación nos permite determinar el significado correcto de la palabra dentro de una oración.
- **Análisis de sentimientos:** Corresponde a tratar de determinar cualidades subjetivas de los datos analizados, tales como sarcasmo, emociones, humor, confusión, entre otros.

Las primeras aplicaciones de *NLP* consistían en la programación de sistemas basados en reglas capaces de realizar tareas simples. Sin embargo, la falta de escalabilidad y la baja adaptabilidad de los algoritmos ante excepciones hicieron que estas nunca gozaran de relativo éxito.

Ante estas problemáticas surge *NLP* estadístico, que combina el uso de algoritmos complejos con modelos de aprendizaje de máquinas capaces de clasificar, extraer o etiquetar elementos de los datos de manera automática con relativa facilidad, indicando la probabilidad de significado de los elementos relevantes. Hoy en día, los modelos de aprendizaje profundo basados en redes neuronales, como *RNN*, *CNN*, *transformers* ó *GCN* constituyen el estado del arte [70] para el tratamiento de tareas y problemas asociados al área de *NLP*, permitiéndole a estos sistemas aprender y evolucionar, al mismo tiempo que trabajan sobre grandes volúmenes de datos.

2.3.2. Clasificadores basados en Aprendizaje Automático

Este tipo de algoritmos permiten identificar patrones en los datos ingresados, construir modelos capaces de identificar y clasificar fenómenos sin la necesidad de contar con conjuntos

de reglas explícitas programadas. En otras palabras, son algoritmos capaces de aprender a resolver problemas a través de los datos. De manera más rigurosa, el concepto lo explica Mitchel en su libro “Machine Learning” donde define Aprendizaje Automático como: “Un programa de computador se dice que aprende de una experiencia E con respecto a una clase o tipo de tarea T y una medida de rendimiento P , si su rendimiento en T medido por P , mejora con la experiencia E ” [41]

La experiencia E puede ser interpretada como un *dataset* o conjunto de datos. Un *dataset* es una recolección de ejemplos concretos, compuestos por características medidas cuantitativamente, que reflejan atributos de una problemática por resolver mediante la utilización de un algoritmo de Aprendizaje Automático [19]. Formalmente, un *dataset* es descrito como un conjunto Z de m instancias o $Z := \{z_i; i := 1, \dots, m | \forall m \in \mathbb{N} \wedge z_i \in \mathcal{Z}\}$, siendo $\mathcal{Z}$ un espacio.

En el área del aprendizaje automático existen dos grandes paradigmas: Aprendizaje Supervisado y no Supervisado, los cuales se diferencian por las anotaciones a las que se asume que se tiene acceso.

- **Aprendizaje Supervisado:** Esta categoría obedece a problemas donde los datos disponibles constituyen ejemplos anotados, lo cual implica que cada dato contiene una serie de atributos y una etiqueta asociada. La meta de esta clase de algoritmos es aprender una función que mapea un vector de atributos a la etiqueta, usando los ejemplos entregados [54]. Formalmente, X e Y denotan los conjuntos de vectores de entrada y salida respectivamente. El objetivo en un problema de aprendizaje supervisado, dado un conjunto de datos de entrenamiento, es aprender una función $F : X \rightarrow Y$, la cual mapea las características de entradas con las de salida, donde los ejemplos pueden ser representados como $\{(x^{(i)}, y^{(i)})\}_{i=1}^n = \{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$ con $n \in \mathbb{N} \wedge x^{(i)} \in \mathcal{X} \wedge y^{(i)} \in \mathcal{Y}$, con $\mathcal{X}$ espacio de entrada e $\mathcal{Y}$ espacio de salida.

Cuando la variable objetivo y que se está tratando de predecir es continua, se llamará al problema de aprendizaje como uno de regresión. Por otro lado, cuando y puede tomar valores discretos, el problema corresponderá a uno de clasificación.

- **Aprendizaje No Supervisado:** Este paradigma recibe un conjunto de datos de entrada X , pero sin el conjunto de salida Y . La idea principal detrás de esto es que mediante la imitación de patrones dentro de grandes conjuntos de datos, el algoritmo sea capaz de construir una interpretación concisa de este y así ser capaz de generar interpretaciones interesantes sobre las mediciones [25].

Redes Neuronales Artificiales Las redes neuronales artificiales o ANN, por sus siglas en inglés, son un modelo de aprendizaje automático inspirado en el comportamiento de las redes neuronales cerebrales, emulando el comportamiento de una neurona y tratándola como un nodo de un grafo, donde los arcos representan las conexiones sinápticas entre neuronas.

Una neurona representa la unidad computacional básica de una red neuronal (ver figura 3), la cual recibe n atributos $x_0, \dots, x_n$ produciendo:

$$\begin{aligned} f(\mathbf{x}) &= \Phi(\langle \mathbf{w}, \mathbf{x} \rangle + b) \\ &= \Phi \left(\sum_{i=0}^n w_i x_i + b \right) \end{aligned} \quad (1)$$

Donde:

- $\mathbf{w} \in \mathbb{R}^n$ corresponde al vector de pesos y representa la relevancia de cada atributo en la función de salida.
- b es el sesgo asociado.
- $\Phi()$ es la función de activación de la neurona.

Existen distintos tipos de funciones de activación (ver tabla 3) dependiendo de qué se quiere lograr con el modelo o que se quiere que realice una capa oculta.

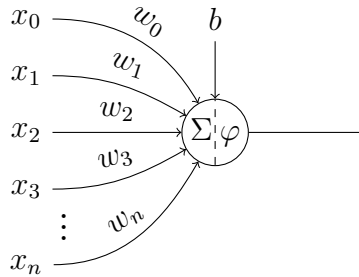


Figura 3: Ejemplo de una neurona.

Fuente: elaboración propia.

Nombre	Función, $\phi(x)$
sigmoidea	$\frac{1}{1+e^{-x}}$
Tangente hiperbólica	$\frac{e^{-x}-e^x}{e^x+e^{-x}}$
ReLU	$\max(0, x)$
GELU	$\frac{x}{2} \left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) \right)$

Tabla 3: Funciones de activación de una Neurona.

Fuente: Elaboración Propia.

Red Feed Forward Una arquitectura tradicional de redes neuronales son las denominadas redes *Feed Forward*, representadas como grafos acíclicos. La característica predominante de estas arquitecturas es la disposición de las neuronas en capas donde cada capa recibe de entrada solo las salidas de la capa anterior y así sucesivamente hasta la capa de salida. La inexistencia de ciclos dentro de las capas corresponde a una decisión de diseño orientada al evitar la formación de bucles infinitos, que impidan el correcto flujo de la información dentro de la red. En la figura 4 es posible observar la arquitectura tradicional de estas redes con una capa de entrada que entrega su *output* a una capa oculta, la cual repite el proceso anterior y así sucesivamente hasta llegar a la capa de salida.

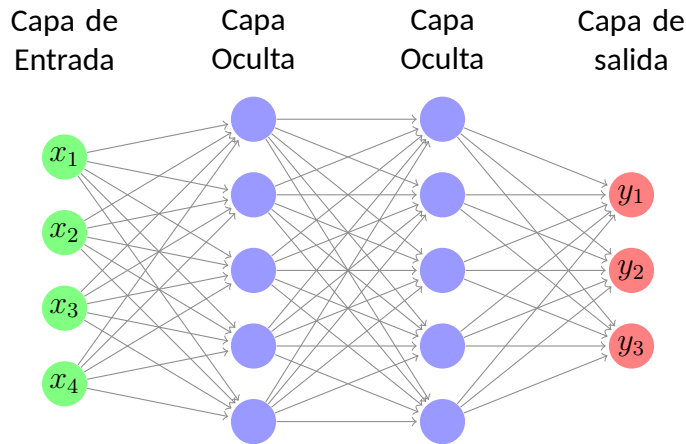


Figura 4: Ejemplo de una red *Feed Forward*.

Fuente: $\text{\LaTeX}$ *neuralnetwork package*.

A modo de resumen, podemos describir una red neuronal de n_l capas, donde L_i corresponde a la i -ésima capa de la red, por lo que L_1 y L_{n_l} corresponden a las capas de entrada y salida respectivamente. Los parámetros de la red son $W^{(l)}$, la matriz de pesos de los arcos, y $b^{(l)}$, el vector de *bias* de la l -ésima capa. El *output* de la capa l con función de activación $\sigma^{(l)}$ viene dado por

$$a^{(l)} = \sigma^{(l)} (W^{(l)} a^{(l-1)} + b^{(l)}) \quad (2)$$

Es decir, para obtener $a^{(l)}$ es necesario primero resolver $a^{(l-1)}$. Entonces, la activación de la capa de salida es determinada por:

$$\hat{y} = \sigma^{L_{n_l}} (W^{(L_{n_l})} a^{(L_{n_l}-1)} + b^{(L_{n_l})}) \quad (3)$$

Entrenamiento Como es posible observar en las ecs. (2) y (3), para obtener el *output* de cada capa es necesario primero resolver el de la capa anterior, y así sucesivamente, hasta llegar a la primera capa. Resulta fácil deducir que, entonces, si el entrenamiento de estas redes será basado en el uso de gradientes, resultará intrínsecamente costoso en términos

computacionales, por lo cual se vuelve necesario que la forma de entrenar sea eficiente en términos computacionales. Es por estas razones que en el entrenamiento de redes neuronales artificiales se utiliza el algoritmo denominado *Backpropagation* [53], el cual calcula el gradiente de la función de pérdida $\mathcal{L}$ respecto a los parámetros de la red, de manera eficiente. Luego, los pesos de la red son actualizados utilizando gradiente descendente, gradiente descendente estocástico o algunas de sus variantes. El funcionamiento en sí del algoritmo es relativamente simple, este va hacia atrás en las capas de la red, calculando el gradiente de la función de pérdida mediante regla de la cadena, una capa a la vez. Debido a que $\mathcal{L}$ no es convexa, no se puede asegurar la existencia de un mínimo global y que este pueda ser encontrado utilizando *Backpropagation*.

Formalmente, tenemos un conjunto de n ejemplos de pares ordenados $\{(x^{(i)}, y^{(i)})\}_{i=1}^n$, donde $x^{(i)}$ corresponde al i -ésimo vector de características y $y^{(i)}$ al vector de etiquetas correspondiente. Se define la función de costo (o pérdida) respecto a un ejemplo como J el costo de la diferencia en el valor predicho $g(x^{(i)})$ y el valor real $y^{(i)}$. El objetivo del algoritmo es minimizar el costo. Por lo tanto, en una iteración, el gradiente descendente y sus variantes actualizan los parámetros de la siguiente forma para la capa l :

$$w_{ij}^{(l)} = w_{ij}^{(l)} - \alpha \frac{\partial}{\partial w_{ij}^{(l)}} J(w, b) \quad (4)$$

$$b_i^{(l)} = b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} J(w, b) \quad (5)$$

- α : *learning rate* o tasa de aprendizaje.
- $w_{ij}^{(l)}$: coeficiente en la posición i, j en la matriz de pesos W en la capa l .
- $b_i^{(l)}$: coeficiente i del vector de sesgo b en la capa l .

Dado un ejemplo de entrenamiento, primero se realiza un *Forward pass* o pase hacia adelante con el fin de calcular las activaciones a lo largo de la red. Luego, una vez procesado un lote de ejemplos, utilizando las ecs. (4) y (5) se determinan, capa por capa, la contribución de errores en la salida, partiendo por la última capa hasta llegar a la primera.

El algoritmo de *Backpropagation* se aplica hasta cumplir algún criterio de término como máximo de iteraciones, convergencia, error, etc.

Redes Neuronales Recurrentes A pesar del poder de las redes *Feed Forward*, estas tienen bastantes limitaciones. Una de estas, es que no pueden aprender relaciones sobre secuencias de datos. En general, esto no es un problema muy grave, sin embargo, al trabajar con

datos que se encuentran ligados de forma temporal, espacial o contextual en el caso de *NLP*, es necesario utilizar métodos de aprendizaje que puedan tomar en cuenta estas relaciones de alguna manera. Es aquí donde las redes neuronales recurrentes o *RNN* por sus siglas en inglés dejan ver sus maravillas, ya que su arquitectura con la presencia de *loops* les permiten capturar de mejor manera las dinámicas de estos datos. A diferencia de las redes *Feed Forward*, las *RNN* pueden retener estados que representan información en ventanas de tiempo de tamaño arbitrario [33]. Más formalmente, una *RNN* es una red especializada en el procesamiento de secuencias temporales. En la figura 5 se muestra un esquema básico del funcionamiento de este tipo de redes.

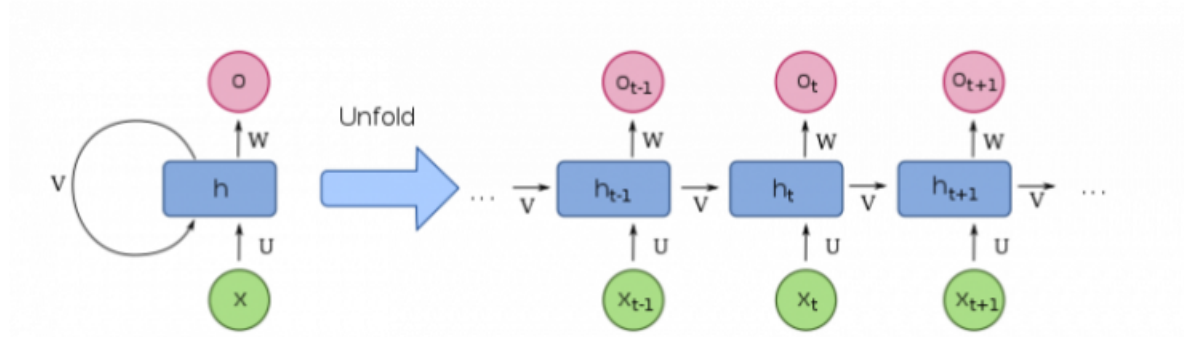


Figura 5: Arquitectura de las neuronas de una *RNN*, Fuente

Definiendo formalmente tenemos las siguientes ecuaciones y variables que representan el funcionamiento general de las neuronas en una *RNN*. Por simplicidad, se definirá una *RNN* con 1 capa oculta, sin embargo, es posible extender este modelo básico apilando capas recurrentes una sobre otra. Es decir, usando como secuencias de entrada de la i -ésima capa, el *output* de la capa $i - 1$.

- $h_t = f_h(h_{t-1}, x_t) = \sigma_h(Ux_t + Vh_{t-1} + b_h)$
- $o_t = f_o(h_t) = \sigma_o(Wh_t + b_w)$
- h_t : nuevo estado en el instante t .
- h_{t-1} : estado en el instante $t - 1$.
- o_t : output en el instante t .
- x_t : vector de input en el instante t .
- V : matriz de pesos para las capas ocultas recurrentes.
- U : matriz de pesos para las capas *inputs* a ocultas.

- W : matriz de pesos para las capas ocultas a *outputs*.
- b : vectores de *bias*.
- σ_o : función de activación de la capa de salida.
- σ_h : función de activación de la capa oculta.
- $t \in [1, \tau], \tau \in \mathbb{N}$

En cada *timestep* (posición en la secuencia de entrada), la red recibe salida del paso previo junto con el vector de entrada x_t en la capa oculta para producir la salida o_t y el estado oculto h_t .

En estos tipos de redes, la utilización de *Backpropagation* debe realizarse a través de los *timesteps*. Lamentablemente, debido a la utilización de manera repetitiva de *Backpropagation* estocástico, los valores del gradiente pueden explotar o desvanecerse conforme el entrenamiento se propaga a instantes anteriores. Este fenómeno es popularmente conocido como el “Problema de desvanecimiento de gradiente” o “*Vanishing gradient problem*” [22].

Long short-term memory (LSTM) En Hochreiter y Schmidhuber, introducen por primera vez la arquitectura *LSTM* como una solución para el “Problema de desvanecimiento de gradiente”. Este tipo de modelos son muy similares a los de redes recurrentes como el presente en la figura 5. Sin embargo, los nodos son reemplazados por células de memoria *LSTM* como la presente en el diagrama 6. La innovación de estas redes yace en la presencia de recurrencia interna o un ciclo interno en la célula, independiente de la recurrencia externa. El aumento de complejidad en los nodos lleva consigo un aumento en la cantidad de parámetros entrenables dentro de la red. Las compuertas presentes en la célula tienen como objetivo activar ciertos comportamientos como el obligar a olvidar estados o controlar el flujo de la información. Se utilizan funciones de activación sigmoidea en estas compuertas, con tal de que si llega a cero se interrumpa el flujo de información y viceversa.

Resulta importante aclarar que las siguientes ecuaciones son válidas asumiendo que cada elemento de la secuencia se representa como un vector. En *NLP* esto se logra mediante la adición de una capa adicional denominada capa de *Embeddings* la cual mapea palabras ó frases a vectores [26].

En estas unidades, el componente más importante es la unidad de estado $s_i^{(t)}$, la cual posee un ciclo interior. Sin embargo, el peso de este ciclo está controlado por la puerta de olvido o *forget gate* $f_i^{(t)}$ la cual asigna el peso como cero o uno en base a la unidad sigmoidea dada por:

$$f_i^{(t)} = \sigma \left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)} \right) \quad (6)$$

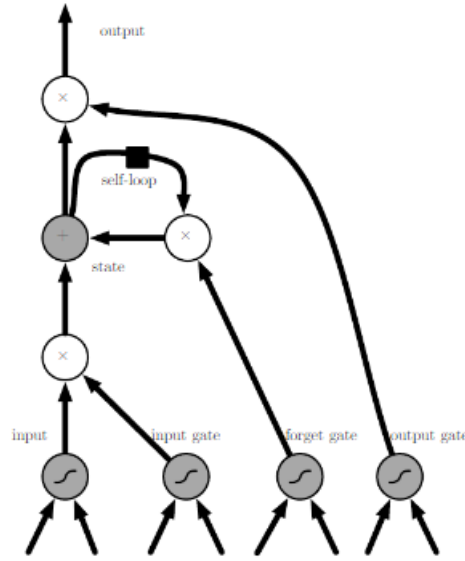


Figura 6: Diagrama de bloque de una célula *LSTM* en una *RNN*, Fuente: Goodfellow *et al.*

donde

- h_t : nuevo estado en el instante t .
- h_{t-1} : estado en el instante $t - 1$.
- x_t : vector de input en el instante t .
- U^f : matriz de pesos de *inputs* para las compuertas de olvido.
- W^f : matriz de pesos para las capas recurrentes de las compuertas de olvido.
- b^f : vectores de *bias* de las compuertas de olvido.

De acuerdo a esta función, el estado interno de la *LSTM* es actualizado como se muestra en la Ec.(7), pero con un peso condicionado al ciclo interno:

$$s_i^t = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma \left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)} \right) \quad (7)$$

donde

- U : matriz de pesos de *inputs* interiores en la *LSTM*.

- W : matriz de pesos para las capas recurrentes interiores en la *LSTM*.
- b : vectores de *bias* interiores en la *LSTM*.
- $s_i^{(t-1)}$: valor de la unidad de estado en el *timestep* anterior.

La compuerta de entrada externa $g_i^{(t)}$ (Ec.(8)) es obtenida de manera similar a la compuerta de olvido (utilizando una función sigmoidea para lograr valores entre 0 y 1), sin embargo, esta posee sesgos y pesos propios:

$$g_i^{(t)} = \sigma \left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)} \right) \quad (8)$$

Nótese como el *output* de la célula (Ec.(9)) puede ser siempre filtrado por la compuerta de salida y también utiliza una función sigmoidea:

$$h_i^{(t)} = \tanh \left(s_i^{(t)} \right) q_i^{(t)} \quad (9)$$

$$q_i^{(t)} = \sigma \left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)} \right) \quad (10)$$

Redes Neuronales Convolucionales Las Redes Neuronales Convolucionales o *CNN* por sus siglas en inglés, son una red neuronal especializada en problemas de visión computacional. Sin embargo, conforme han ganado popularidad, se han comenzado a utilizar en otras tareas donde han probado ser efectivas, como *NLP* [70]. Estas consisten en redes profundas con capas de entrada, salida y ocultas, siendo la principal característica de estas la presencia de capas convolucionales, es decir, capas que realizan la operación de convolución, la cual es equivalente integrar la multiplicación de la entrada a la capa de convolución y el kernel de convolución. El mapa de características generado a partir de la operación de convolución suele ser entregado a una capa de *pooling*, siendo esta seguida por capas completamente conectadas y de normalización hasta llegar a las capas de *output* del modelo.

En la figura 7 se muestra como ejemplo la red *VGG-16* [63], que corresponde a una red convolucional, donde podemos apreciar las capas de convolución, *pooling* y completamente conectadas, tal como se mencionó anteriormente.

Convolución Esta operación es bastante utilizada en el procesamiento de señales, ya que permite obtener la respuesta de una señal frente a un impulso para todo instante de tiempo, siempre y cuando el sistema cumpla con ser homogéneo y de tiempo-invariante. Formalmente, una convolución discreta para una señal unidimensional frente a un impulso se define de la siguiente manera:

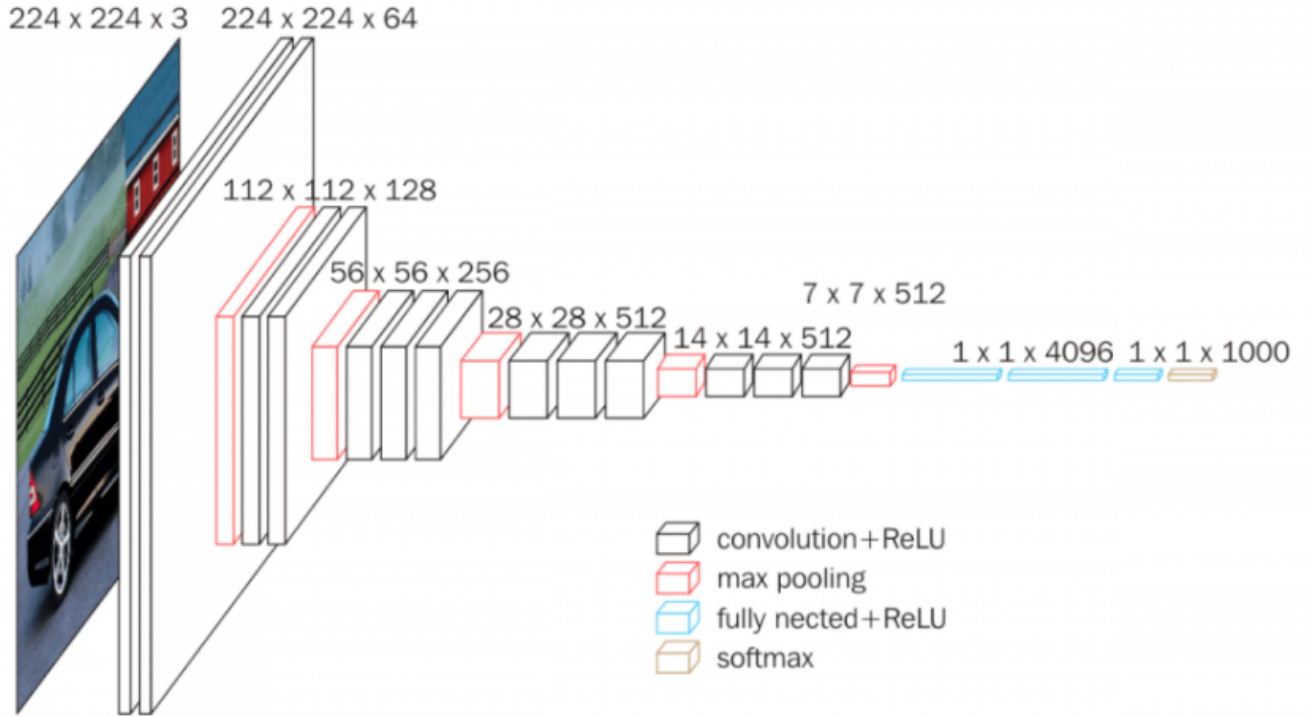


Figura 7: Esquema de la red convolucional VGG-16

Fuente: neurohive

$$s(i) = (x * k)(i) = \sum_{j=-\infty}^{\infty} x[j]k[i - j] \quad (11)$$

donde

- $x[i]$: señal en un tiempo discreto i .
- $k[j]$: Impulso en el tiempo discreto j .
- $s(i)$: Señal resultante en el tiempo discreto i (superposición del impulso frente a la señal).

Para redes neuronales, x corresponde al *input* de la capa, mientras que k representa un filtro. Considerando nuevamente el ejemplo de clasificación de imágenes. La aplicación de la operación de convolución se puede interpretar como el efecto de aplicar el filtro k sobre el vector de entrada x generando un *feature map* formado por neuronas que comparten parámetros. En el caso particular, como la imagen corresponde a una estructura $2D$, el filtro

a utilizar también tiene dicha forma. Matemáticamente, la operación queda definida de la siguiente manera:

$$S(i, j) = (X * K)(i, j) = \sum_{\substack{m=-\infty \\ n=-\infty}}^{\infty} X[m, n]K[i - m, j - n] \quad (12)$$

- X : señal de entrada, en este caso una imagen $2D$.
- K : filtro de la señal.
- S : Resultado de la convolución.

Capa Convolutiva Como ha sido explicado, las redes convolucionales consideran la aplicación de operaciones de convolución dentro de ellas. Esto lo realizan mediante las denominadas capas convolucionales. Este tipo de capas utiliza K filtros de tamaños preestablecidos, aplicándolos sobre los datos de entrada X , generando así lo que se podría denominar en el caso de imágenes como efecto de ventana móvil que recorre la imagen.

Para denotar la aplicación de una operación de convolución sobre una capa l denotamos la siguiente notación para expresar la salida de esta capa como:

$$a^{(l)} = \sigma_l (a^{(l)} * K^{(l)} + b^{(l)}) \quad (13)$$

- $a^{(l)}$: salida de la l -ésima capa.
- σ_l : función de activación de la l -ésima capa.
- $K^{(l)}$: filtros o arreglo de kernels para la l -ésima capa.
- $b^{(l)}$: vector de sesgos de la l -ésima capa.

Hiperparámetros Cada capa convolutiva posee los siguientes hiperparámetros:

- **Tamaño del Kernel:** Constituye la cantidad de píxeles procesados juntos, es decir, las dimensiones que tiene el kernel (i.e. 2×2 , 4×5).
- **Padding:** Constituye la adición de píxeles de valor 0 en los bordes de las imágenes, esto se realiza con la intención de no perder los valores bordes de los píxeles al realizar la convolución.

- **Stride:** Corresponde al número de píxeles en los que se mueve la ventana de análisis en cada iteración. Un *stride* de 2 significa que el *kernel* actual tiene una diferencia de 2 píxeles de posición con respecto al que lo presidió.
- **Tamaño del filtro:** Se refiere al tamaño que tendrán los filtros utilizados en la convolución. De acuerdo a la literatura actual, lo más común es utilizar filtros de 3×3 y compensar con profundidad.
- **Tipo de Pooling y tamaño:** Usualmente, se utiliza *Maxpooling*, de tamaño 2. Esto genera un *downsampling* significativo sobre los datos de entrada.
- **Dilatación:** Corresponde a ignorar una cantidad determinada de píxeles de la imagen en cada *kernel*, reduciendo potencialmente el uso de memoria sin pérdida significativa de la señal. Una dilatación de 2 sobre un *kernel* de 5×5 lo expande a 11×11 , sin embargo, mantiene el procesamiento de 25 píxeles en memoria.(e.g. figura 8)

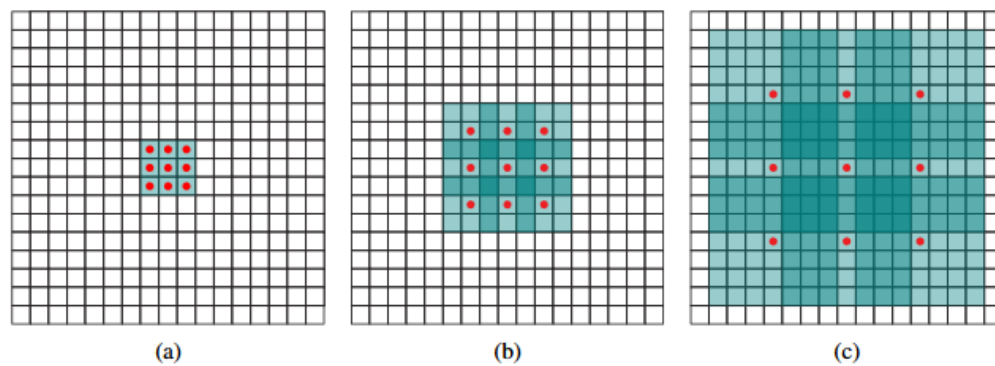


Figura 8: (a) Es el kernel producido al utilizar una convolución sin dilatación (equivalente a una dilatación igual a 1), (b) Corresponde al kernel (a), pero con dilatación 2 ($3 \times 3 \rightarrow 7 \times 7$), (c) Es generado al aplicar una dilatación de 4 sobre (a) ($3 \times 3 \rightarrow 15 \times 15$). Al utilizar una dilatación de l , esto significa que son $l - 1$ píxeles omitidos entre cada píxel del kernel. En esta figura, los puntos rojos muestran los píxeles de la imagen utilizados por el kernel, y el coloreado azul representa el nuevo campo receptivo alcanzado por el kernel sobre la imagen.

Fuente: Yu y Koltun

Este tipo de redes son comúnmente utilizadas para el procesamiento de imágenes y videos, aunque en los últimos años se han comenzado a utilizar en tareas de procesamiento de lenguaje natural, obteniendo relativo éxito en tareas como *semantic parsing* [20], recuperación de *queries* de búsqueda [62], modelado de sentencias [27], clasificación [72], predicción de sentencias [13], entre otras tareas. La intuición detrás de esto viene de que las CNN, a diferencia de las redes recurrentes, mediante el uso de filtros en las capas de convolución serán capaces de generar abstracciones contextuales locales para los ejemplos entregados, las cuales se encontrarán ligadas al tamaño de los filtros utilizados, pese a no tener la ventaja teórica de memoria infinita de las RNN [9].

Transformers Las redes *Transformers* constituyen una arquitectura de red neuronal relativamente simple donde en lugar de utilizar redes con arquitecturas complejas como *RNN* o *CNN*, recurren a una arquitectura basada únicamente en mecanismos de atención. El modelo fue propuesto originalmente por Vaswani *et al.* como una alternativa a las arquitecturas reinantes de la época en las áreas de *NLP* y *Computer Vision*.

Para esta arquitectura, un *encoder* mapea una secuencia de símbolos de entrada $(x_1, \dots, x_n)$ a una secuencia de representaciones continuas $z = (z_1, \dots, z_n)$. Dado z , el *decoder* genera una secuencia de símbolos de salida $(y_1, \dots, y_n)$ para cada elemento. En cada paso el modelo de predicción, el decoder es autorregresivo, consumiendo los símbolos previamente generados como entrada adicional para generar el siguiente.

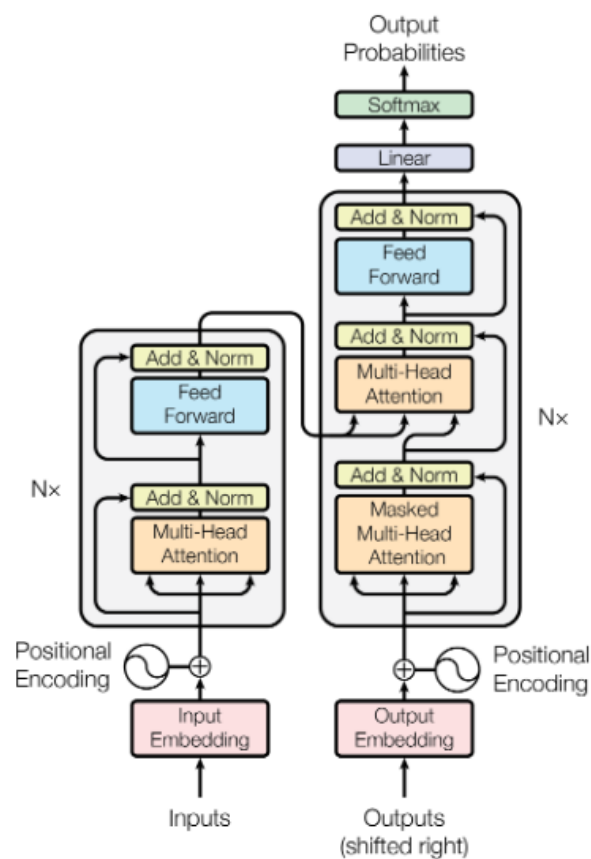


Figura 9: Arquitectura del modelo *Transformers*
Fuente: Vaswani *et al.*

Encoder y Decoder Stacks

Encoder: El *encoder* consiste en un *stack* de N capas idénticas. Cada capa posee 2 subcapas. La primera capa es un mecanismo de autoatención multicabezal y la segunda es una

completamente conectada red *feed forward*. Se emplea una conexión residual entre cada una de las sub-capas, seguida por una capa de normalización. Es decir, la salida de cada sub-capas es $\text{LayerNorm}(x + \text{Sublayer}(x))$, donde $\text{Sublayer}(x)$ es la función implementada por la sub-capas. En pos de facilitar estas conexiones residuales, todas las sub-capas del modelo, al igual que las capas de *embeddings* generan salidas de dimensión $d_{\text{model}} = 512$ [68].

Decoder: El *decoder* también se encuentra compuesto por un *stack* de N capas. Además de las ya presentes 2 sub-capas en el codificador (*encoder*), el *decoder* agrega una tercera capa, la cual realiza el proceso de atención multicabezal sobre la salida del *stack* de codificación. Análogo al *encoder*, se utilizan conexiones residuales sobre cada sub-capas seguidas por una capa de normalización. Asimismo, se modifica la sub-capas de autoatención en el *stack* de decodificación, con tal de prevenir que se preste atención a posiciones subsiguientes. El proceso de enmascarado, combinado con el hecho de que, en tiempo de entrenamiento los *embeddings* correspondientes a la secuencia de salida se entregan al *decoder* con un *offset*, aseguran que la predicción para la posición i dependa únicamente de las salidas conocidas en posiciones menores a i .

Mecanismos de Atención

La función de atención puede ser descrita como un mapeo de un *query*/consulta y un conjunto de pares de llave valor hacia una salida, donde la consulta, las llaves, valores y la salida son vectores. La salida es calculada como la suma pesada de los valores, donde el peso asignado a cada valor es determinado por una función de compatibilidad de la consulta con la llave correspondiente.

Atención de Producto punto escalado En un mecanismo de atención de producto punto escalado (*Scaled Dot-Product Attention*, figura 10). La entrada consiste en una serie de consultas, llaves (de dimensiones d_k) y valores (de dimensiones d_v). Se determina el producto punto de consulta con todas las llaves, dividiendo cada una por $(\sqrt{d_k})$ y aplicando una función *softmax* con tal de obtener los pesos en los valores.

La función de atención es aplicada simultáneamente sobre un conjunto de *queries*, agrupados en la matriz Q . Las llaves y valores son también agrupados en las matrices K y V respectivamente. Por consiguiente, la matriz de *outputs* se encuentra determinada de la siguiente forma:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (14)$$

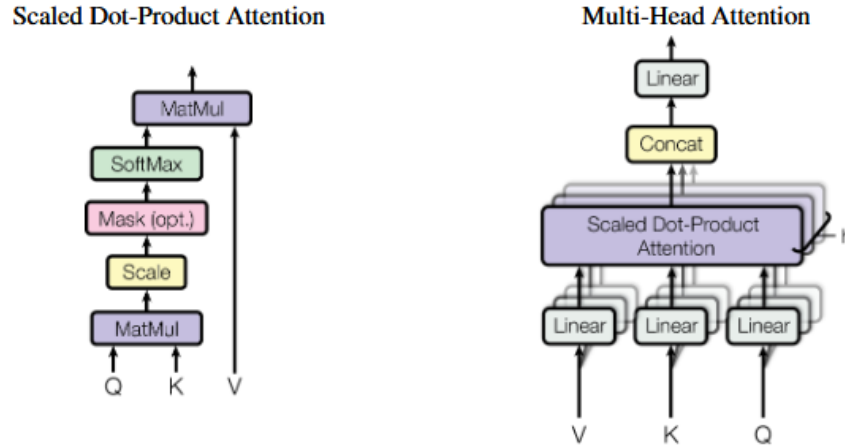


Figura 10: (Izquierda) Atención de Producto punto escalado. (Derecha) Atención multicabezal consistente en varias capas de atención corriendo en paralelo.

Fuente: Vaswani et al.

Atención Multicabezal En lugar de realizar una sola función de atención con llaves d_{model} -dimensional, valores y *queries*. Se realiza una proyección lineal de llaves, valores y *queries* h veces, con diferentes proyecciones aprendidas a espacios de dimensiones d_k , d_v y d_k , respectivamente. Sobre cada una de estas proyecciones es posible aplicar la función de atención en paralelo, retornando valores de salida d_v -dimensionales. Estos son concatenados y nuevamente proyectados, generando así los valores finales, como se puede evidenciar en la parte derecha de la figura 10 [68].

El uso de este multicabezal permite al modelo conjuntamente prestar atención a diferentes subespacios de representación en diferentes posiciones. Por otro lado, para el caso donde se utiliza un solo cabezal es evidente que no se presentará este comportamiento.

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O$$

donde $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$

donde las proyecciones son matrices de parámetros $W_i^Q \in \mathcal{R}^{d_{model} \times d_k}$, $W_i^K \in \mathcal{R}^{d_{model} \times d_k}$, $W_i^V \in \mathcal{R}^{d_{model} \times d_v}$ y $W^O \in \mathcal{R}^{hd_v \times d_{model}}$.

2.3.3. Estado del Arte Modelos

Modelos Multi-lingua Con el fin de evaluar el *dataset* generado, ver las capacidades de transferencia, generalización y medir la eficacia de este en otros dialectos del español, se

sigue el diseño experimental propuesto recientemente por Arango Monnar *et al.* donde utilizan los modelos de Pamungkas *et al.*, Aluru *et al.* y Ranasinghe y Zampieri denominados como ACL19, ECML20 y EMNLP20 respectivamente, para evaluar las capacidades de transferencia de Discursos de Odio entre distintos idiomas y dialectos. A continuación se entregan mayores detalles respecto a la arquitectura de los modelos utilizados.

ACL19 Pamungkas *et al.* utilizan distintos modelos para su clasificación. Sin embargo, en Arango Monnar *et al.* determinaron utilizar un modelo de aprendizaje conjunto basado en LSTM con MUSE⁴ embeddings entre los 4 modelos planteados por Pamungkas *et al.*.

MUSE corresponde a “multi-lingual Unsupervised and Supervised Embeddings” [14] y funciona construyendo un diccionario bilingüe entre dos lenguajes sin utilizar “corpus paralelos”⁵ sino que alineando directamente los espacios de representación (*embeddings*) de palabras monolingües de manera no supervisada. Sean $\mathcal{X} = \{x_1, \dots, x_n\}$ e $\mathcal{Y} = \{y_1, \dots, y_m\}$, dos conjuntos de *embeddings* de palabras de largos n y m , ambos pertenecientes a un lenguaje fuente y a otro objetivo respectivamente. El modelo es entrenado de forma no supervisada para ser capaz de diferenciar entre elementos aleatorios obtenidos de $W\mathcal{X} = \{Wx_1, \dots, Wx_n\}$ u obtenidos de $\mathcal{Y}$. Como resultado, el entrenamiento resulta en un juego de dos agentes, donde el discriminador busca maximizar la habilidad para discriminar el origen de *embeddings*, mientras que, por otro lado, la transformación W busca confundir al discriminador haciendo $W\mathcal{X}$ e $\mathcal{Y}$, lo más similares posible.

Objetivo del discriminador Refiriéndose a los parámetros del discriminador como θ_D . Se considera la probabilidad $P_{\theta_D}(\text{origen} = 1|z)$, qué un vector z pertenece al mapeo de *embeddings* de origen (a diferencia de *embeddings* objetivo) de acuerdo al discriminador. La función de pérdida del discriminador puede ser descrita como:

$$\mathcal{L}(\theta_D|W) = -\frac{1}{n} \sum_{i=1}^n \log P_{\theta_D}(\text{origen} = 1|Wx_i) - \frac{1}{m} \sum_{i=1}^m \log P_{\theta_D}(\text{origen} = 0|y_i) \quad (15)$$

Objetivo del mapeo En el ambiente no supervisado, W es ahora entrenado para que el discriminador no sea capaz de predecir con precisión el origen de los *embeddings*:

$$\mathcal{L}(W|\theta_D) = -\frac{1}{n} \sum_{i=1}^n \log P_{\theta_D}(\text{origen} = 0|Wx_i) - \frac{1}{m} \sum_{i=1}^m \log P_{\theta_D}(\text{origen} = 1|y_i) \quad (16)$$

⁴<https://github.com/facebookresearch/MUSE>

⁵En NLP corpus paralelos hace referencia a un recurso lingüístico, consistente en textos de dos lenguas alineados/traducidos con cierto nivel de granularidad; generalmente a nivel de párrafo, aunque también a nivel de sección, página o incluso a veces de palabra. [64]

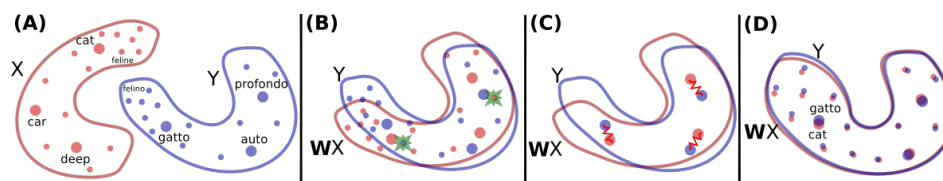


Figura 11: **(A)** Hay dos distribuciones de *embeddings*, primero palabras en Inglés mostradas en rojo denotadas por X y segundo palabras en Italiano mostradas en azul denotadas por Y , se quiere alinear estos *embeddings* de tal forma que se obtiene la traducción. Cada punto representa una palabra en el espacio. El tamaño del punto es proporcional a la frecuencia de la palabra en el corpus de entrenamiento. **(B)** Mediante entrenamiento adversativo se busca aprender una matriz de rotaciones W , tal que, alinee ambas distribuciones con cierto margen de error. Las estrellas verdes, son pares de palabras seleccionadas aleatoriamente, las cuales se entregan al discriminador para discernir si los *embeddings* pertenecen a una misma distribución. **(C)** El mapeo W es refinado utilizando análisis de Procrustes⁷. Este método utiliza palabras frecuentes alineadas por el paso anterior como puntos ancla y minimiza la función de energía correspondiente a un sistema de resortes entre anclas. El mapeo refinado, es luego usado para mapear todas las palabras en el diccionario. **(D)** Finalmente, se traduce utilizando el mapeo refinado W y una métrica de distancia, denominada *CROSS-DOMAIN SIMILARITY LOCAL SCALING*, la cual expande regiones del espacio con alta densidad de puntos (i.e. el área alrededor de “cat” en **(A)**), haciendo así que palabras centralizadas como “cat” se encuentren más separadas de otros vectores de palabras que como lo estarían originalmente en **(A)**.

Fuente: Conneau *et al.* 2017

La figura 12 muestra la arquitectura del modelo ACL19, el cual es entrenado para la tarea de detección de discursos de Odio modelada como un problema de clasificación binaria. Para utilizarlo es necesario el uso de la *API* de *Google translate* con tal de traducir los textos de entrenamiento en español al inglés y viceversa. La arquitectura del modelo consiste en dos redes *LSTM* seguidas por una capa densa con activación *RELU* y una de *dropout* del 0,3. El funcionamiento es el siguiente: una de las redes aprende la tarea en el lenguaje de origen, mientras la otra en el lenguaje objetivo. El *output* de las redes es concatenado y entregado a una capa densa con activación sigmoidea como capa de *output*. Esta arquitectura es optimizada por un optimizador *RMS* con parámetros por defecto, utilizando 9 *epochs* y un *batch size* de 16.

⁷El análisis de Procrustes, constituye una técnica estadística de análisis de formas, el cual es utilizado para encontrar la rotación y/o reflexión óptima de un objeto con respecto a otro. La técnica fue propuesta por Schönemann para el denominado “Problema Ortogonal de Procrustes”.

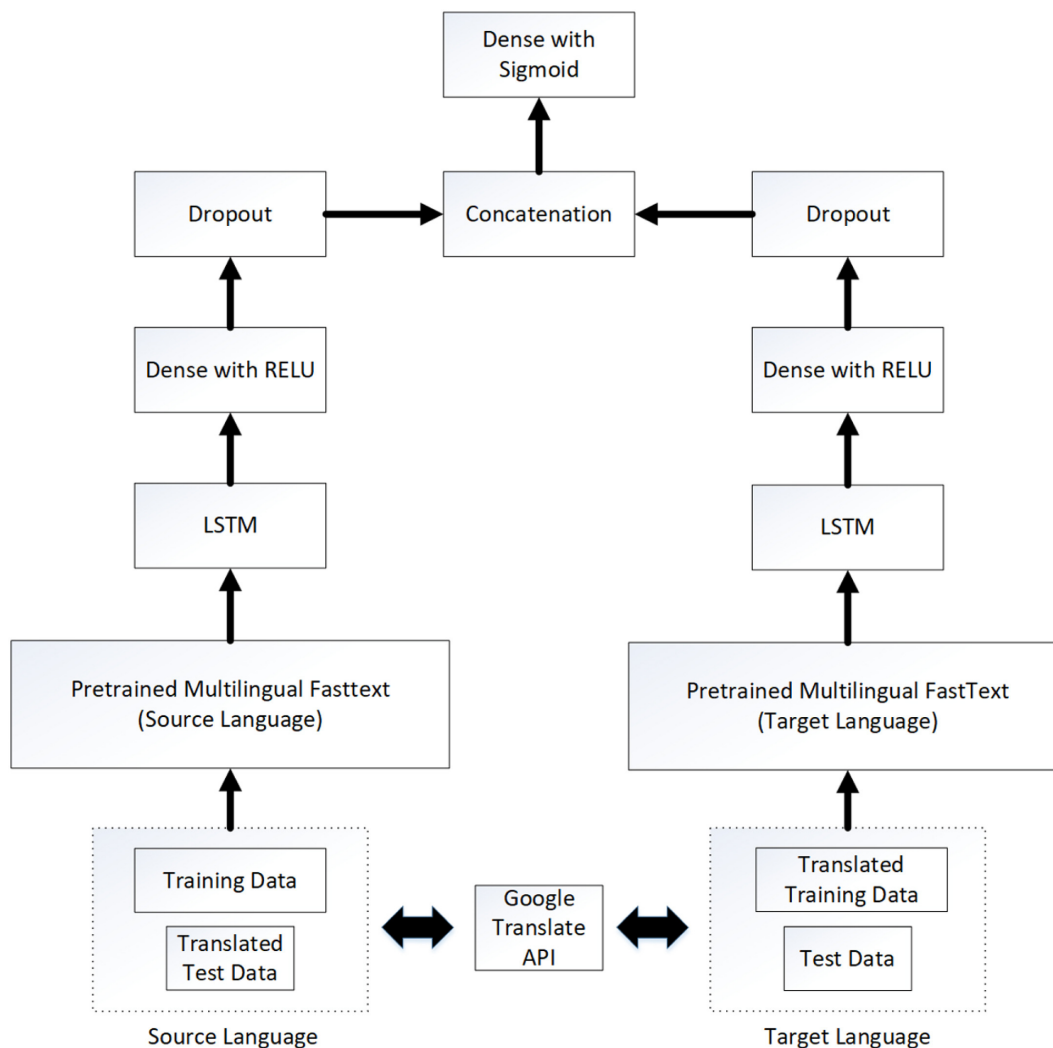


Figura 12: Arquitectura Modelo ACL19.
Fuente: Pamungkas *et al.*

ECML20 Los autores Aluru *et al.* plantean 4 modelos, *MUSE + CNN-GRU*, *Translation + BERT*, *LASER + LR*, y *multi-lingual BERT*. Después de una comparación experimental, Arango Monnar *et al.* llegan a la conclusión que el mejor desempeño es obtenido con *LASER + LR*, es decir, utilizando los *embeddings* de *LASER* y un regresor logístico, entrenado para la tarea de detección de discursos de Odio modelada como un problema de clasificación binaria. *LASER* es un acrónimo para *Language-Agnostic Sentence Representations*⁸. El cual funciona de la siguiente manera: dada una sentencia de *input*, *LASER* genera *embeddings* aplicando *max-pooling* sobre el largo de secuencias del *output* de una BiLSTM de 5 capas, con *embeddings* de palabras de tamaño 320 y capas ocultas con 512 neuronas (ver figura 13), resultando así en unos *sentence embeddings* de tamaño 1024. El sistema utiliza un único *encoder* con un

⁸<https://github.com/facebookresearch/LASER>

vocabulario BPE⁹ compartido para los distintos lenguajes.

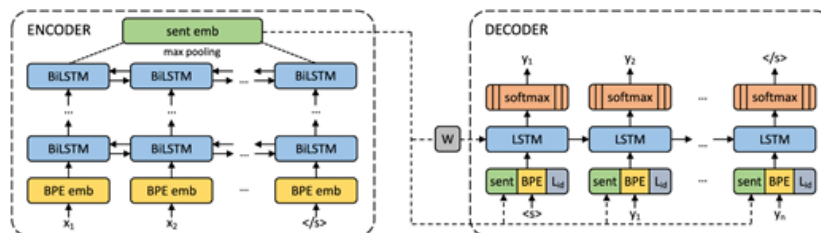


Figura 13: Arquitectura generadora de *embeddings* para LASER.

Fuente: Artetxe y Schwenk

EMNLP20 Ranasinghe y Zampieri proponen utilizar el modelo *transformer* XLM-Roberta. XLM-R o XLM-Roberta es entrenado sobre 100 idiomas con multi-lingual Masked Language Modeling como tarea de entrenamiento, su arquitectura consiste en 270M (millones) de parámetros, con 12 capas, 768 estados ocultos, 3072 estados ocultos *feed forward* y 12 *attention heads*. Esta arquitectura toma un *input* de a lo más 512 tokens y retorna una representación de la secuencia. El primer *token* es siempre [CLS] el que contiene los *embeddings* especiales de clasificación. Para clasificación de texto XLM-R toma el último estado oculto del primer token [CLS] como representación de la secuencia. Finalmente, un clasificador *softmax* es añadido con tal de predecir la probabilidad de la clase “*c*” para una tarea específica.

El ajuste fino del modelo es realizado entrenando sobre la tarea de clasificación de discursos de odio sobre un lenguaje de origen y luego para evaluar la transferencia de conocimiento se realiza la clasificación sobre otro lenguaje. Como el problema de clasificación de “Discursos de Odio” es binario, la capa de clasificación final debiese ser reemplazada por una sigmoide, sin embargo, al observar que los experimentos de Arango Monnar *et al.* no realizan esto, con tal de mantener el marco comparativo se conservó *softmax* como capa final con dos neuronas.

⁹Algoritmo Byte Pair Encoding para tokenización.

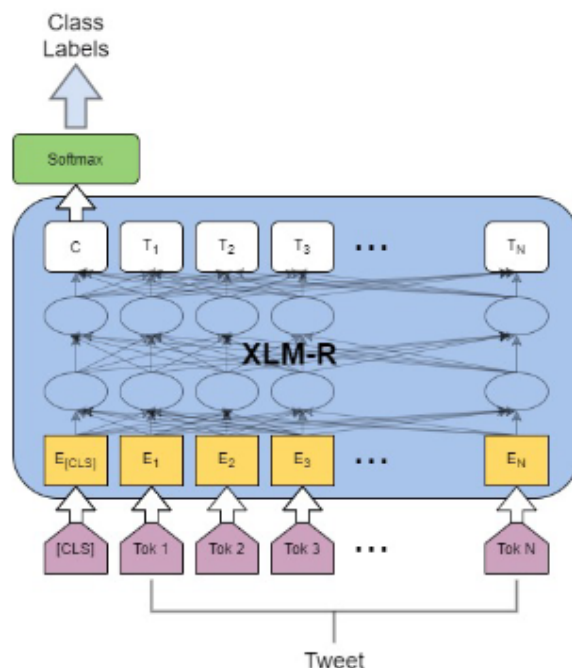


Figura 14: Arquitectura de clasificación Modelo EMNLP20.

Fuente: Ranasinghe y Zampieri

2.4. Métricas

En distintas áreas de la ciencia resulta necesaria la utilización de distintos descriptores estadísticos e indicadores cuantitativos capaces de reflejar el desempeño de distintas tareas. Estos descriptores e indicadores suelen ser denominados como métricas o métricas de desempeño dependiendo del área científica en la cual se trabaje.

2.4.1. Métricas de Rendimiento

Siempre que se trata con algoritmos de aprendizaje automático para resolver un problema, es necesario determinar cómo se evaluará el desempeño del algoritmo. Más específicamente, qué tan buena es la capacidad de generalización de este, entendiéndose esto como la capacidad del algoritmo de hacer frente a nuevos datos sobre los cuales no fue entrenado explícitamente y generar ante ellos el output correcto. Con el fin de evaluar esto, es que se utiliza el llamado conjunto de pruebas o *test Set*. Este conjunto de datos constituye un conjunto aparte de los datos de entrenamiento, el cual permite evaluar el desempeño del modelo generado sobre datos anotados, pero sobre los cuales el modelo no fue entrenado explícitamente.

Con base en lo anterior y con el objetivo de tener una medida cuantitativa del algoritmo

diseñado, es que se debe utilizar o diseñar una métrica de rendimiento. En algoritmos de clasificación, es importante primero introducir los conceptos de: Verdadero positivo, Verdadero negativo, Falso positivo, Falso negativo definidos a continuación:

- VP = Verdadero positivo: Cantidad de clasificaciones realizadas correctamente, que presentan la condición o característica evaluada.
- VN = Verdadero negativo: Cantidad de clasificaciones realizadas correctamente, que no presentan la condición o característica evaluada.
- FP = Falso positivo: Cantidad de clasificaciones realizadas incorrectamente, que no presentan la condición o característica evaluada.
- FN = Falso negativo: Cantidad de clasificaciones realizadas incorrectamente, que sí presentan la condición o característica evaluada.

Usualmente, para los algoritmos de clasificación binaria se utilizan las siguientes métricas [67, 25]:

■ **Accuracy/Exactitud:**

$$\text{exactitud} = \frac{VP + VN}{VP + VN + FP + FN} \quad (17)$$

donde

- VP = Verdadero positivo
- VN = Verdadero negativo
- FP = Falso positivo
- FN = Falso negativo

■ **Precision/Precisión:**

$$\text{precisión} = \frac{VP}{VP + FP} \quad (18)$$

donde

- VP = Verdadero positivo
- FP = Falso positivo

■ **Recall/Exhaustividad:**

$$\text{Recall} = \frac{VP}{VP + FN} \quad (19)$$

donde

- VP = Verdadero positivo
- FN = Falso negativo

■ **F1-Score/Valor-F1:**

$$F_1 = 2 \cdot \frac{\text{Precisión} \cdot \text{Recall}}{\text{Precisión} + \text{Recall}} = \frac{2VP}{2VP + FP + FN} \quad (20)$$

donde

- VP = Verdadero positivo
 - FP = Falso positivo
 - FN = Falso negativo
- **AUROC:** Constituye el área bajo la curva ROC, que representa el *tradeoff* entre la tasa de verdaderos positivos y la tasa de falsos positivos, donde los valores mayores a 0,5 son indicativos de un desempeño mayor al de un clasificador aleatorio y un desempeño igual a 1 es indicativo de un clasificador perfecto. Sea X una variable aleatoria continua, dada un parametro de corte T , la instancia es clasificada como “positiva” si $X > T$, y negativa en caso contrario. X sigue una densidad de probabilidad $f_1(x)$ si realmente pertenece a la clase “positiva”, y sigue una densidad $f_0(x)$ en caso contrario. Por lo tanto, la tasa de verdaderos positivos (TVP) y la tasa de falsos positivos (TFP) son determinadas por:

$$\begin{aligned} \text{TVP}(T) &= \int_T^{\infty} f_1(x)dx \\ \text{TFP}(T) &= \int_T^{\infty} f_0(x)dx \end{aligned}$$

$$\text{TVP}(U) : U \mapsto y(x)$$

$$\text{TFP}(U) : U \mapsto x$$

$$\text{AUROC} = \int_{x=0}^1 \text{TVP}(\text{TFP}^{-1}(x)) dx = P(X_1 > X_0) \quad (21)$$

donde

- U representa el valor del umbral de corte determinado por el eje x .
- X_1 probabilidad de clasificar una instancia como positiva (VP, FP)
- X_0 probabilidad de clasificar una instancia como negativa (VN, FN)

2.4.2. Métricas de Consenso

El consenso entre medidas hace referencia al grado de concordancia entre uno o más conjuntos de mediciones. Los métodos estadísticos para evaluar el consenso son utilizados con tal de juzgar adecuadamente la calidad de los datos obtenidos, la variabilidad entre anotadores/instrumentos y para decidir si la forma de evaluar variables es correcta o no.

En general, es posible estar interesado en determinar si las mediciones realizadas por dos o más individuos o incluso por dos o más técnicas distintas producen resultados similares estadísticamente hablando. Esto puede ser definido como consenso, concordancia o reproducibilidad de las mediciones dependiendo del área de estudio y la naturaleza de los datos.

La necesidad de medir estas características de los datos ha generado las siguientes métricas:

- **Cohen's Kappa/Kappa de Cohen [12]:** Constituye una medida de consenso la cual trabaja bajo el supuesto: "Si los anotadores operaran de manera aleatoria, entonces obtendríamos distribuciones distintas para cada anotador" [12], buscando así la generación de medidas de consenso con una corrección de consenso aleatorio. Para calcular el *Agreement/Consenso* esperado A_e en el coeficiente K , se asume que la asignación aleatoria de categorías a un ítem se encuentra gobernada por distribuciones a-priori únicas e independientes para cada anotador y estas muestran el sesgo individual del anotador. La probabilidad $P(\kappa|c_i)$ de que un anotador c_i clasifique un ítem arbitrario en la categoría κ , es estimada usando la proporción $\hat{P}(\kappa|c_i)$ de ítems asignados por el anotador c_i a la categoría κ es decir, $n_{c_i\kappa}$ dividido por la cantidad de ítems m .

$$P(\kappa|c_i) = \hat{P}(\kappa|c_i) = \frac{n_{c_i\kappa}}{m} \quad (22)$$

Para una categoría κ , la probabilidad de que dos anotadores distintos c_1 y c_2 asignen la misma categoría a un mismo ítem viene dada por la probabilidad conjunta $\hat{P}(\kappa|c_1) \cdot \hat{P}(\kappa|c_2)$. Entonces, el consenso esperado constituye la suma de probabilidades conjuntas sobre todo el conjunto de categorías $\kappa \in \mathcal{K}$.

$$K = A_e = \sum_{\kappa \in \mathcal{K}} \hat{P}(\kappa|c_1) \cdot \hat{P}(\kappa|c_2) = \sum_{\kappa \in \mathcal{K}} \frac{n_{c_1\kappa}}{m} \frac{n_{c_2\kappa}}{m} = \frac{1}{m^2} \sum_{\kappa \in \mathcal{K}} n_{c_1\kappa} n_{c_2\kappa} \quad (23)$$

- **Krippendorff's Alpha/Alfa de Krippendorff [31]:** El alfa de Krippendorff es calculado tomando en cuenta que el *Agreement/Consenso* esperado es calculado observando la distribución total de clasificaciones sin tomar en cuenta qué anotadores generaron que anotaciones. Esto se aplica para múltiples anotadores y permite medir distintas magnitudes de desacuerdo/*disagreement*.

$$D_o = \frac{1}{n \cdot c \cdot (c - 1)} \sum_{i \in I} \sum_{j=1}^{\kappa} \sum_{l=1}^{\kappa} n_{i\kappa_j} n_{i\kappa_l} d_{\kappa_j\kappa_l} \quad (24)$$

$$d_{\kappa\kappa} = 0 \quad \forall \kappa$$

$$D_e = \frac{1}{nc(nc-1)} \sum_{j=1}^{\kappa} \sum_{l=1}^{\kappa} \mathbf{n}_{\kappa_j} \mathbf{n}_{\kappa_l} \mathbf{d}_{\kappa_j \kappa_l} \quad (25)$$

$$\mathbf{d}_{\kappa\kappa} = 0 \quad \forall \kappa$$

$$\alpha = 1 - \frac{D_0}{D_e} \quad (26)$$

Donde

- D_o : desacuerdo/*disagreement* observado.
- D_e : desacuerdo/*disagreement* esperado.
- $i \in I$: conjunto de ítems clasificados.
- $\mathbf{n}_{i\kappa}$: cantidad de veces que el ítem i es asignado el valor κ , es decir, la cantidad de anotadores que indican que el ítem i tiene la categoría κ .
- $\mathbf{n}_{\kappa}$: cantidad de veces que el valor κ es asignado a cualquier ítem, por cualquier anotador.
- $\mathbf{n}$: cantidad total de ítems.
- $\mathbf{c}$: cantidad de anotadores c .
- $\mathbf{d}$: métrica de distancia en un espacio.

En el caso particular, donde queremos discretizar y considerar la distancia entre distintas clases igual para todas, podemos ver la distancia como:

$$\mathbf{d}_{ab} = \begin{cases} 0 & a = b \\ 1 & a \neq b \end{cases} \quad (27)$$

Si bien ambas métricas buscan determinar niveles de consenso, el alfa de Krippendorff, trabaja bajo suposiciones como que el acuerdo es calculado observando la distribución total de juicios sin tomar en cuenta que anotadores realizaron la codificación, es decir, si de 3 anotadores, el primero y el segundo anotaran la misma clase, generaría igualmente el mismo valor si ahora las respuestas del segundo y el tercer anotador fueran las mismas. Es decir, a diferencia del kappa de Cohen, el alfa de Krippendorff no asume que las distribuciones aleatorias serán necesariamente distintas entre anotadores, adquiriendo así mayor robustez ante valores faltantes a diferencia del Kappa de Cohen [8].

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

En este capítulo se presenta la propuesta de solución para el problema de recolección y anotado de datos en redes sociales chilenas que permitan entrenar modelos orientados a la detección y el monitoreo de discursos de odio, al igual que la propuesta de evaluación de los datos generados junto con su diseño experimental, lo cual permitirá contrastar los resultados con *benchmarks* del estado del arte.

3.1. Metodología de Muestreo

A partir de la literatura investigada, se determinó que las publicaciones sobre la creación y evaluación de conjuntos de datos relacionados a discursos de odio poseen las siguientes falencias: poca formalización de los criterios de muestreo (fechas, palabras claves, *hashtags*, etc.), definiciones conceptuales poco claras o ambiguas, carencia de razones para el uso de las herramientas de muestreo (facilidad de uso, acceso académico, código abierto), etc. Es con esto en mente que determinamos los siguientes puntos que la metodología de muestreo a seguir debe cumplir:

- Definiciones conceptuales claras y concretas, lo más apegadas a los estándares internacionales.
- Definir criterios de búsqueda de manera clara y relevante capaces de responder a las siguientes interrogantes: ¿Por qué usar estas palabras?, ¿Por qué es relevante este periodo de tiempo?
- Definir claramente la elección de herramientas de muestreo para responder: ¿Por qué utilizamos esta herramienta?, ¿Cuáles son sus ventajas y desventajas respecto a otras?.

3.1.1. Criterios y Conceptos Clave

Esta sección busca describir y explicar los criterios y conceptos utilizados con tal de acotar el problema, dar mayor formalidad y generar resultados reproducibles. En cuanto a la forma de definirlos, se recurrió a lo indicado por el estado del arte, tomando conceptos ya descritos por la literatura o, en caso de no existir, recurriendo a entidades internacionales relacionadas con el tema específico para así utilizar sus definiciones.

Para la comprensión del concepto “discurso de odio”, se optó por atenerse a definiciones internacionales, con tal de evitar objeciones sobre la arbitrariedad de estas, en particular,

se utilizó la definición de discurso de odio generada por la ONU como: **“Cualquier forma de comunicación de palabra, por escrito o a través del comportamiento, que sea un ataque o utilice lenguaje peyorativo o discriminatorio en relación con una persona o un grupo sobre la base de quiénes son o en otras palabras, en razón de su religión, origen étnico, nacionalidad, raza, color, ascendencia, género u otro factor de identidad”** [21].

Específicamente, para la confección del diccionario de palabras claves se recurrió a *HateBase* [16], sitio web parte de *Sentinel Project for Genocide Prevention*, el cual consiste en una base de datos de insultos definidos en 98 idiomas asociados a las respectivas comunidades discriminadas. Estos datos fueron enriquecidos mediante el uso de conocimiento experto de hablantes del dialecto chileno del español. Tomando en cuenta que *Hatebase* entrega las comunidades asociadas a los insultos, se identificaron las comunidades más propensas a ser objetivos de ataques como la comunidad LGBTQ+, comunidades indígenas, migrantes y los grupos feministas en su calidad de mujeres. Se determinaron éstas y no otras comunidades, ya que además de encontrarse disponibles en *Hatebase* se tomó en cuenta su categoría histórica de grupos discriminados (ciudadanos de segunda categoría, poco acceso a derechos, precarización de condiciones de vida, etc., hechos que incluso es posible evidenciar en artículos noticiosos recientes) [44, 60, 51]. Emulando lo realizado por Poletto *et al.* se generó un diccionario de palabras claves compuesto por insultos y palabras neutrales, disponibles en el anexo 38. Resulta importante destacar que debido a diferencias culturales intrínsecas del contexto chileno, algunas palabras o frases que a simple vista pueden ser calificadas como insultos, presentan una carga negativa altamente variable, dependiendo de la relación que tienen el emisor y receptor del mensaje, pudiendo significar desde ser un término afectuoso hasta directamente un ataque.

Para diversificar la búsqueda, se replicó lo indicado por [3], añadiendo *hashtags* recolectados manualmente asociados a discursos de odio en *Twitter*, al igual que cuentas asociadas, que fueran objetivos de discursos de odio o generadoras de discursos de odio. Estas cuentas y *hashtags* se encuentran disponibles en el anexo 39.

Se determinó como intervalos de búsqueda temporales, desde el primero de enero del 2020 hasta el primero de junio del 2022 tomando en cuenta los procesos históricos relevantes ocurridos a lo largo de estos años, además de la pandemia, como lo fue la quema de campamentos migrantes [44], la promulgación de la Ley de Matrimonio Igualitario, entre otros hechos políticos y noticiosos relevantes reconocidos por generar revuelo entre grupos y personas asiduas a la propagación de discursos de odio.

3.1.2. Herramienta de Muestreo

En primera instancia, se generaron experimentos con tal de determinar qué herramientas presentes en el estado del arte son actualmente funcionales y, por lo tanto, viables para el muestreo de la investigación. Finalmente, se utilizó *Tweepy*, librería de *python* para la utilización de la API de *Twitter*. Entre las razones para la selección de *Tweepy*, en lugar de *Twython*,

se debe a que la escogida corresponde a una librería mantenida activamente por la comunidad y con documentación clara sobre los casos de uso y ejemplos de implementación. Se descarta *Twython*, pues posee escasa documentación, además de no ser un repositorio mantenido activamente. Se recurrió a *Twitter*, como plataforma de recolección, debido a ser la única que a la fecha mantenía el *Academic Research Access*. También se intentó utilizar *Crowdtangle*, plataforma para la transparencia de *Meta* para tener acceso a los datos para académicos, sin embargo, el acceso a la *API* fue negado, pues, actualmente *Meta* se encuentra cerrando la plataforma sin planes de crear una alternativa [32].

3.1.3. Esquema General

Para la etapa de muestreo y recolección de datos se realizó en primera instancia la recolección en los periodos señalados (enero del 2020 hasta el primero de junio del 2022) utilizando únicamente las palabras, cuentas y *hashtags* detallados en los anexos 39 y 38. Imitando la metodología indicada por Poletto *et al.*, se recolectaron la mayor cantidad de *tweets* posibles asociados al periodo de búsqueda, tras lo cual se extrajo una muestra del 10 % de los datos para ser analizada manualmente, con tal de verificar el balance de los datos bajo la presencia o no de odio. Al fallar este enfoque y obtener una proporción de odio cercana al 6 %, muy inferior a la esperada de al menos un 20 % – 30 % de *tweets* asociados al odio dentro de la muestra. En base a esto se propone un esquema de recolección manual, utilizando idealmente más de un recolector con tal de enriquecer los datos y lograr generar un *dataset* de ≈ 5000 *tweets* balanceados como odio y no odio. El esquema manual en caso de constar con más de 1 recolector se plantea el asignar una categoría objetivo y un periodo específicos a los recolectores con tal de evitar la duplicación de datos, entregándoles a los recolectores palabras claves, cuentas y *hashtags* relevantes; además de una breve introducción sobre los conceptos y criterios necesarios para comprender el problema de los discursos de odio. Con tal de facilitar la comprensión, se visualiza en la figura 15 el esquema de recolección resumido.

3.2. Metodología de Anotación

En concordancia con lo detallado en el estado del arte con respecto a las metodologías de anotación, observamos la utilización de las siguientes reglas o “buenas prácticas”:

- Definir *guidelines* claros y sencillos para los anotadores.
- Idealmente, utilizar al menos tres anotadores con tal de evitar empates.
- Utilizar tests ocultos es una buena práctica para evaluar el comportamiento de los anotadores a lo largo del anotado.

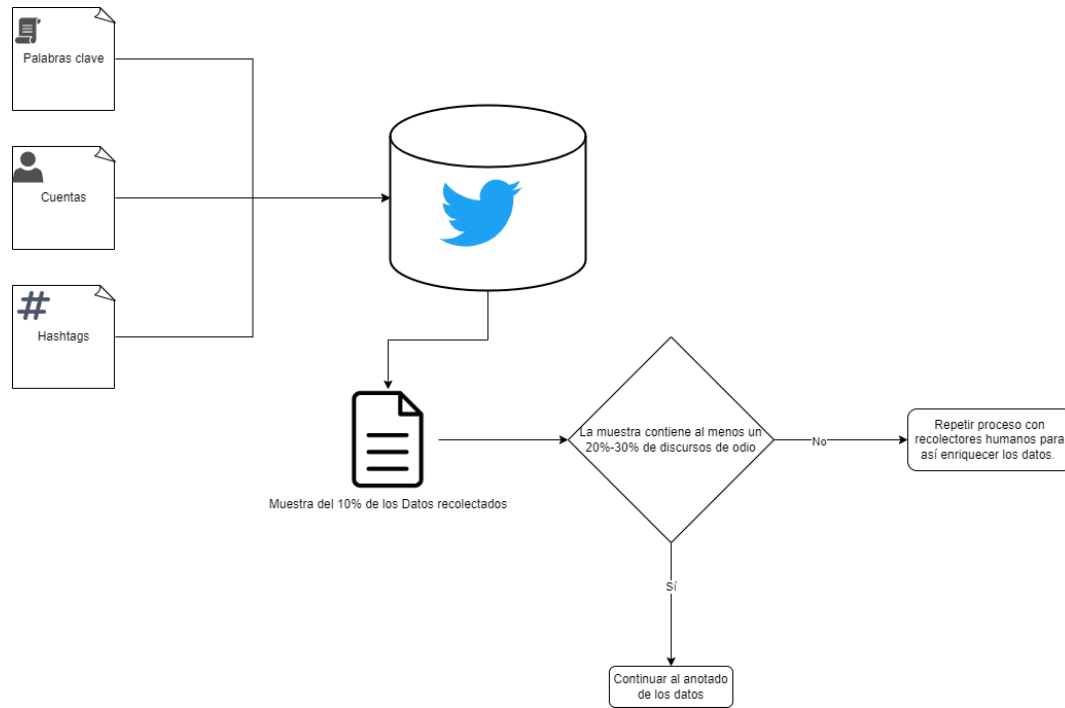


Figura 15: Esquema de recolección de datos.
Fuente: Elaboración propia.

- Utilizar una plataforma de *Crowdsourcing* adecuada, cumpliendo con criterios de facilidad de uso y relevancia en el problema.
- Definir clases o categorías claras.
- Utilizar una métrica de medición de *agreement* adecuada: Kappa de Cohen si son al menos 2 personas y Alfa de Krippendorff en caso de ser más.

3.2.1. Criterios y Conceptos Clave

Se utilizaron los mismos conceptos asociados a la sección 3.1.1. Sin embargo, se ahondó con mayor profundidad en ellos con tal de entregarles a los anotadores una base sólida sobre la cual poder consultar y tomar decisiones, generando así los *guidelines* de la tarea, disponibles en el anexo 5.6.

3.2.2. Plataforma de *Crowdsourcing*

Para encontrar una herramienta de anotación adecuada, se investigaron las utilizadas por la literatura. Sin embargo, al tratar de utilizarlas, surgieron diversos problemas como la falta de

demo (pruebas gratuitas) para verificar que las herramientas posean las especificaciones necesarias para el problema, la eliminación de herramientas útiles o absorción por otras compañías, la desaparición de las plataformas, la imposibilidad de asignar o crear equipos de anotación manualmente y finalmente la falta de accesos académicos. En vista de todos estos problemas encontrados en los sistemas recomendados por la literatura, se decidió realizar una búsqueda exhaustiva sobre plataformas que ofrecen los servicios de anotado, finalmente llegando a *LightTag* [47] plataforma que al momento de realizar el anotado cumplía todos nuestros requisitos, menos la realización de pruebas ocultas, por lo cual se optó por descartar la realización de esta última.

3.2.3. Esquema General

Basándose en la literatura expuesta, se generó un esquema de anotación para un total de 15 anotadores, dividiéndolos en 5 equipos de 3, donde cada equipo anotará un total de ≈ 1100 tweets a lo largo de 4 semanas, dividiendo los tweets en 3 particiones de 250, 250, 600 tweets, respectivamente. En caso de recurrir a la recolección manual, se asignarán tweets no recolectados por los integrantes del equipo a estos mismos, con tal de no inducir sesgo en la clasificación.

Resulta importante notar que los tweets recolectados son anotados tomando en cuenta el contexto en el que se encuentran, es decir, estos tweets muestran el *thread* relacionado al tweet, en caso de tenerlo. La figura 16 ilustra como se realizaba el proceso de anotado en la plataforma.

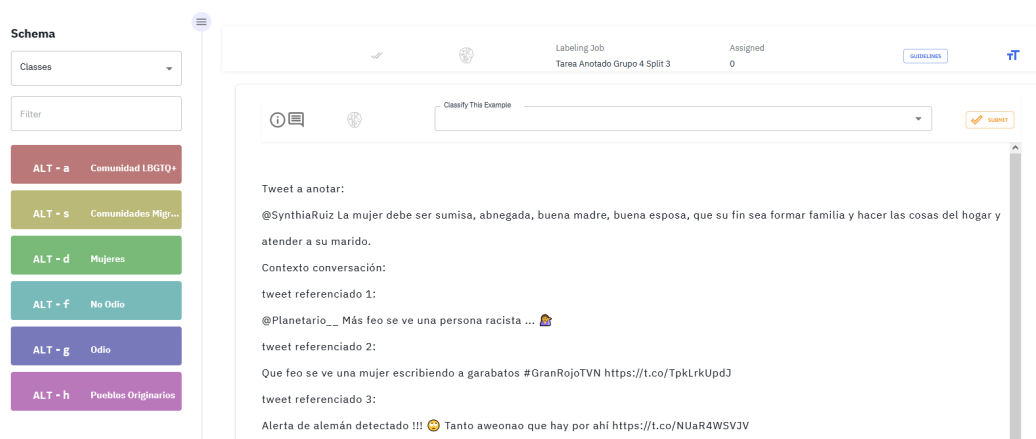


Figura 16: Ilustración del anotado de tweets con su *thread* en *LightTag* [47].
Fuente: Elaboración propia.

Para la redacción del esquema de anotado se hizo énfasis especial en la metodología/Framework generado por Krippendorff en [31] donde señala las siguientes reglas básicas para poder utilizar su métrica, así como para poder garantizar la robustez de los resultados, especialmente en el área de NLP:

- Se debe emplear un esquema de anotación exhaustivo, claro y con facilidad de uso, con instrucciones paso por paso sobre su utilización.
- Se deben utilizar criterios claros en la elección de codificadores para que así otros puedan recrear los datos.
- Se debe asegurar que los anotadores que generan los datos trabajen de forma independiente, sin comentar resultados ni ítems clasificados.

También se hizo especial uso de las consideraciones entregadas por Artstein y Poesio, quienes entregan reglas para la determinación de puntos de corte en cuanto a métricas de evaluación de resultados.

3.2.4. Esquema de anotado

En concordancia a lo visto en la literatura, se generaron las siguientes clases a identificar por los participantes del anotado.

Hate Speech/Discursos de Odio El *tweet* incita al odio por sí mismo o en el contexto del hilo en el que se encuentra, constituyendo un ataque o promoviendo la discriminación de una persona o grupo, solo sobre la base de pertenencia a uno de los siguientes grupos: comunidades migrantes, comunidad LGBTQ+, mujeres y pueblos originarios.

Por ejemplo: “*Es que no se como la vamos a nombrar, porque no es persona menstruante ni persona eyaculante, le podremos decir maricon a secas?*”, es anotado como discurso de odio, pues el autor está haciendo burla sobre las formas de denominar a las personas transgénero y promueve finalmente, el referirse a ellas atacándolos verbalmente, utilizando el insulto final.

Comunidades El *tweet* hace referencia a una comunidad si se refiere a un individuo de acuerdo con su pertenencia a esta misma o si directamente hace referencia a la comunidad en sí.

Por ejemplo: “**Chicas expresando lo que sufren y el acoso que deben aguantar al tener pechos grandes* La tonta de turno: problemas que quiero*”, no constituye discurso de odio, pues no constituye un ataque en base a su pertenencia a un grupo, sino que está haciendo burla de quienes ven los problemas de otras personas como deseables. Por otro lado el *tweet* hace referencia a las mujeres, por lo cual debiese ser anotado como relacionado con esta comunidad objetivo.

Las comunidades en que se puede categorizar un *tweet* son las siguientes:

- Comunidad LGBTQ+
- Comunidades indígenas
- Migrantes
- Mujeres

3.3. Evaluación de los Datos

Tras una revisión bibliográfica relacionada con el problema, se observa una clara tendencia a evaluar el desempeño de nuevos recursos utilizando métricas y descripciones estadísticas que evidencian la calidad intrínseca de los datos. No obstante, existen pocas propuestas que consideren las cualidades extrínsecas de estos recursos, como su capacidad de transferencia a otros dominios y su capacidad para mitigar sesgos en relación a otros recursos. La necesidad de evaluar estas características de los datos mediante el uso de otros recursos alternativos, surge debido a que las técnicas de validación habituales como la utilización de **Hold-out data** extraída aleatoriamente del corpus con tal de generar un conjunto de validación [19], no permiten evidenciar los sesgos ya existentes en el corpus y por lo tanto son poco útiles para evidenciar la capacidad de generalización del modelo hacia otros dominios, así como su robustez ante sesgos presentes en otros recursos. Por estas razones, la evaluación de nuevos recursos de datos debe tomar en cuenta un diseño experimental capaz de reflejar no solo el desempeño de modelos del estado del arte en ese recurso, si no que también revelar como se comportan los modelos entrenados sobre este nuevo recurso en recursos considerados *benchmarks* por el actual estado del arte.

Además de lo mencionado anteriormente, y con tal de generar claridad a la hora de realizar esta evaluación es necesario explicitar qué se busca responder con estas evaluaciones y cómo estas preguntas ayudarán a generar el diseño experimental. Las interrogantes son las siguientes:

- ¿Es transferible el nuevo recurso a otros dominios?
- ¿Permite este nuevo recurso alcanzar mejoras funcionalmente relevantes en la detección de *hate speech* en español?
- ¿Mitiga este nuevo recurso los niveles de sesgo respecto de recursos similares?

Las respuestas a estas interrogantes permitirán evaluar no solo la capacidad de los datos para reflejar correctamente las diferentes formas en que ocurre y no ocurre un discurso de odio, sino que también contrastar su desempeño con el de otros dominios similares. Además de permitir dilucidar si las capacidades funcionales del recurso cumplen con las competencias mínimas necesarias que le permitan ser un aporte valioso para el problema. En tercer lugar,

el estudio del comportamiento del recurso ante sesgos es de especial interés, pues permitirá visualizar problemáticas que puedan surgir en caso de encontrarse sesgos prevalentes hacia comunidades o para evidenciar las capacidades para distinguir entre distintas comunidades de los modelos.

3.3.1. Diseño Experimental

En esta sección se describen y detallan de forma general los conjuntos de datos, modelos a utilizar y la configuración de los experimentos de evaluación de: transferibilidad, funcionalidad y sesgos realizados.

Datasets Los *datasets* utilizados son los generados en Basile *et al.* 2019 (ES y EN), Arango Monnar *et al.* 2022 (CL₁) y el presentado en este trabajo (CL₂).

Los cuales se encuentran detallados a continuación:

- Basile *et al.* 2019 (ES y EN): Es una colección de 6100 *tweets* en español (ES) y 12000 *tweets* en inglés (EN), los cuales se encuentran clasificados en las siguientes categorías *Hate Speech* (discurso de odio), *Targeted* (dirigido) y *Aggressive* (agresivo).
- Arango Monnar *et al.* 2022 (CL₁): Es un *dataset* de 9834 *tweets* en el dialecto Chileno del español (CL₁), los cuales se encuentran clasificados en 18 categorías entre las cuales se encuentra discursos de odio.
- El presentado en este trabajo (CL₂): Es un *dataset* de 4547 *tweets* en el dialecto Chileno del español (CL₂), los cuales se encuentran clasificados en 5 categorías, discurso de odio, Mujeres, Comunidades Migrantes, Comunidad LGBTQ+ y Pueblos Originarios.

A continuación, se presenta la tabla 4 con información estadística relevante de ambos conjuntos de datos.

<i>Dataset</i>	Abreviación	<i>C</i>	$ Dataset $	Odio
Basile <i>et al.</i> 2019 Español	ES	3	6100	41 %
Basile <i>et al.</i> 2019 Inglés	EN	3	12000	42 %
Arango Monnar <i>et al.</i> 2022	CL ₁	18	9834	6 %
Datos Propios	CL ₂	5	4547	46 %

Tabla 4: Información estadística de los *datasets* utilizados. *C* corresponde al número de clases (etiquetas) correspondiente a cada conjunto de datos. $|Dataset|$ representa la cardinalidad de datos del conjunto. Odio representa el porcentaje de Odio en cada *dataset*.

Fuente: Elaboración Propia.

Experimentos de Transferencia Siguiendo la configuración experimental descrita por Arango Monnar *et al.* 2022, se plantea utilizar los mismos modelos (ver sección 2.3.3) y *datasets* del estado del arte ya mencionado, añadiendo el recurso generado. La configuración experimental se encuentra descrita en la tabla 5 donde se muestran los modelos, *datasets* de entrenamiento y evaluación, buscando con esto contrastar las capacidades de transferencia de cada conjunto de datos hacia otro.

Resulta importante indicar que al momento de la recreación de los experimentos en diciembre de 2022, el *framework* de experimentación propuesto por Arango Monnar *et al.* 2022 no se encontraba disponible en forma funcional en el repositorio <https://github.com/aymeam/hate-speech-framework>, sin embargo, guiándose por lo descrito en su publicación y otras publicaciones relacionadas, estos fueron replicados utilizando los repositorios asociados a Pamungkas *et al.* 2021, Aluru *et al.* 2020 y Ranasinghe y Zampieri 2020. Estos experimentos y sus códigos asociados se encuentran disponibles en el siguiente repositorio Chilean-Hate-Speech. A continuación se entregan mayores detalles respecto al *setting* experimental sobre el que se realizaron los experimentos de transferencia.

Modelo	Entrenamiento	Evaluación
ACL19	es	en cl ₁ cl ₂
	en	es cl ₁ cl ₂
	cl ₁	es en cl ₂
	cl ₂	es en cl ₁
ECML20	es	en cl ₁ cl ₂
	en	es cl ₁ cl ₂
	cl ₁	es en cl ₂
	cl ₂	es en cl ₁
EMNLP20	es	en cl ₁ cl ₂
	en	es cl ₁ cl ₂
	cl ₁	es en cl ₂
	cl ₂	es en cl ₁

Tabla 5: Configuración de experimentos de transferencia.
Fuente: Elaboración propia

Experimentos de Funcionalidades En la literatura existen pocas propuestas de evaluación respecto a funcionalidades de los modelos entrenados para clasificación de discursos de odio. Sin embargo, recientemente Röttger *et al.* 2022 proponen un recurso de evaluación funcional denominado *Hatecheck* el cual incluye de conjuntos de datos en distintos idiomas.

Cada conjunto se encuentra anotado como discurso de odio y categorizado respecto a la comunidad y funcionalidad que se busca evaluar.

Los autores mencionados previamente proponen *Hatecheck* [55] conjunto de tests funcionales orientados a discursos de odio en inglés a diferencia de su versión más nueva [55], la cual está disponible en 11 idiomas distintos. La evaluación de los modelos entrenados en este corpus conlleva una gran serie de ventajas por sobre la evaluación mediante un conjunto reservado de datos. Permitiendo contrastar las ventajas funcionales explícitas que generan nuestros datos frente a otros conjuntos de entrenamiento, lo cual no es fácilmente demostrable mediante el uso de datos reservados o mediante el uso de conjuntos de datos que no evalúen funcionalidades.

En la tabla 6 se encuentra descrito el recurso respecto sus funcionalidades categorizadas. Nótese que en esta tabla son de especial importancia para el presente trabajo las funcionalidades en las categorías de “Uso de Obscenidades” (F8 y F9), “Negación” (F12 y F13), “Expresión” (F14 y F15) y “Contra Discurso” (F18 y F19), pues son estas funcionalidades las que engloban las características más complejas de la detección de discursos de Odio. Estas funcionalidades nos permiten dimensionar que tan bien logra el modelo, en base a los datos, generar estas distinciones no triviales. En la tabla 7 se encuentran las configuraciones experimentales para evaluar las funcionalidades de *Hatecheck* por modelo y *dataset* de entrenamiento. La categorización de funcionalidades se realiza tal y como indican en Röttger *et al.* 2022, agrupando las predicciones por categoría y calculando las métricas por categoría.

Categoría	Funcionalidad	Clasificación
Despectivo	F1: Expresa emociones negativas fuertes (explícito)	Odio
	F2: Describe utilizando atributos muy negativos (explícito)	Odio
	F3: Deshumanización (explícito)	Odio
	F4: Implícitamente Despectivo	Odio
Lenguaje Amenazante	F5: Amenaza directa	Odio
	F6: Amenaza mediante un juicio de valor	Odio
Insultos	F7: Odio expresado mediante insultos	Odio
Uso de Obscenidades	F8: Odio expresado utilizando obscenidades	Odio
	F9: Uso de obscenidades sin odio	No Odio
Referencia al Pronombre	F10: Odio expresado mediante referencias en cláusulas subsecuentes	Odio
	F11: Odio expresado mediante referencias en frases subsecuentes	Odio
Negación	F12: Odio expresado usando declaraciones positivas negadas	Odio
	F13: No odio expresado mediante declaraciones odiosas negadas	No Odio
Expresión	F14: Odio expresado como pregunta	Odio
	F15: Odio expresado como opinión	Odio
identificador de grupo no odioso	F16: Declaraciones neutrales utilizando identificadores de grupos protegidos	No Odio
	F17: Declaraciones positivas utilizando identificadores de grupos positivos	No Odio
Contra Discurso	F18: Denuncias contra discursos de odio que lo citan	No Odio
	F19: Denuncias contra discursos de odio que lo referencian	No Odio
Abuso contra objetivos no protegidos	F20: Abuso dirigido a objetos	No Odio
	F21: Abuso dirigido a individuos objetivos (no miembros de grupos protegidos)	No Odio
	F22: Abuso dirigido a grupos no protegidos (e.g. profesiones)	No Odio
Variaciones ortográficas	F23: Cambio de caracteres adyacentes	Odio
	F24: caracteres faltantes	Odio
	F25: Faltan límites de palabras	Odio
	F26: espacios añadidos entre caracteres	Odio
	F27: Escritura <i>leet</i> ¹⁰	Odio

Tabla 6: Descripción recurso *Hatecheck* [55].

Fuente: Elaboración propia

¹⁰*Leet speak* o *leet* (1337 5p34k o 1337 en la escritura *leet*) es un tipo de escritura compuesta de caracteres alfanuméricos, usada por algunas comunidades y usuarios de diferentes medios de internet.

Modelo	Entrenamiento	Evaluación
ACL19	es	<i>Hatecheck</i>
	en	<i>Hatecheck</i>
	cl ₁	<i>Hatecheck</i>
	cl ₂	<i>Hatecheck</i>
ECML20	es	<i>Hatecheck</i>
	en	<i>Hatecheck</i>
	cl ₁	<i>Hatecheck</i>
	cl ₂	<i>Hatecheck</i>
EMNLP20	es	<i>Hatecheck</i>
	en	<i>Hatecheck</i>
	cl ₁	<i>Hatecheck</i>
	cl ₂	<i>Hatecheck</i>

Tabla 7: Configuración de experimentos de funcionalidades para *Hatecheck* y evaluación de sesgos.

Fuente: Elaboración propia

Experimentos de Sesgo Hasta el día de hoy, en la literatura existen pocos ejemplos de análisis de sesgos para el problema de *hate speech*, por lo cual la creación y estudio de métricas dedicadas al análisis de sesgos involuntarios es bastante reciente para este problema en específico. Borkan *et al.* 2019, generan una propuesta de métricas enfocadas a la medición de sesgos involuntarios orientados a comunidades inspiradas en la métrica AUC. Las métricas propuestas son las siguientes:

$$\text{Subgroup AUC} = \text{AUC}(D_g^- + D_g^+) \quad (28)$$

$$\text{BPSN AUC} = \text{AUC}(D^+ + D_g^-) \quad (29)$$

$$\text{BNSP AUC} = \text{AUC}(D^- + D_g^+) \quad (30)$$

- **Subgroup AUC:** Representa la capacidad del modelo para comprender y separar correctamente dentro del subgrupo.
- **Background Positive Subgroup Negative (BPSN) AUC:** Representa la capacidad del modelo para reducir falsos positivos en el subgrupo.
- **Background Negative Subgroup Positive (BNSP) AUC:** Representa la capacidad del modelo para evitar falsos negativos en el subgrupo.
- D^+ : Ejemplos positivos en el *dataset* que no pertenecen al subgrupo.

- D^- : Ejemplos negativos en el *dataset* que no pertenecen al subgrupo.
- D_g^+ : Ejemplos positivos en el *dataset* que pertenecen al subgrupo.
- D_g^- : Ejemplos negativos en el *dataset* que pertenecen al subgrupo.

Para evaluar más objetivamente la presencia de sesgos en el *dataset* propuesto se propone utilizar el recurso *Hatecheck*, con tal de evidenciar sesgos presentes en los modelos entrenados sobre los recursos presentados. Las razones que motivan el uso de un *dataset* externo, en lugar de la utilización de un conjunto de datos reservados pertenecientes al *dataset* propuesto, tienen como objetivo el descartar una posible ventaja que tendría nuestro conjunto por sobre otros al ser evaluado sobre datos que fueron generados con los mismos criterios manteniendo así la imparcialidad en la evaluación de sesgos. Una posible desventaja que tendrá este enfoque es que al ser *Hatecheck* un recurso confeccionado a partir del español hablado en España y del hablado en Argentina, éste no será capaz de reflejar con la misma fidelidad el español hablado por el resto de Latinoamérica lo cual puede ser una eventual fuente de errores en los modelos entrenados y evaluados sobre este recurso. Como configuración experimental se reutilizará la descrita en la tabla 7.

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

En este capítulo se presenta la validación del recurso propuesto en esta memoria. Esto incluye el análisis de los datos obtenidos del proceso de anotado y los resultados de los 3 tipos de experimentos propuestos en la sección anterior: transferibilidad, análisis funcional, y presencia de sesgos.

4.1. Resultados Recolección

En la etapa de recolección se obtuvieron 45000 *tweets* de la *API* de *Twitter* los cuales se filtraron por lenguaje “Español” y por *Retweets*, tras lo cual se obtuvo un conjunto de 5721 *tweets* en español que no eran *Retweets*, pudiendo estos ser de odio o no. Sobre éstos se extrajo una muestra aleatoria del 10 %, la cual se etiquetó de manera rápida siguiendo la definición de discursos de odio entregada anteriormente y presente en el anexo 5.6. De esta muestra se determinó que, contrario a lo indicado por la literatura, la recolección automática no logró generar una muestra con un porcentaje de datos relevantes (Odio en una proporción del 20 % – 30 %) por lo cual se decidió recurrir a un proceso de recolección manual con el motivo de enriquecer los datos. Para esto se reunió a 15 voluntarios, quienes debieron realizar una búsqueda en *Twitter* según lo detallado en la sección 3.1.3. Se realizó la siguiente asignación de criterios y fechas de búsqueda (Ver tabla 8) donde los recolectores tuvieron la tarea de recolectar al menos 100 *tweets* por comunidad.

Ayudantes	Fechas Búsqueda	Comunidad Búsqueda
1	['2020-01-01', '2020-04-20']	Comunidad LGBTQ+
2	['2020-04-20', '2020-08-08']	Mujeres
3	['2020-08-08', '2020-11-26']	Pueblos Originarios
4	['2020-11-26', '2021-03-17']	Comunidades Migrantes
5	['2021-03-17', '2021-07-05']	Comunidades Migrantes
6	['2021-07-05', '2021-10-23']	Mujeres
7	['2021-10-23', '2022-02-10']	Pueblos Originarios
8	['2022-02-10', '2022-06-01']	Comunidades Migrantes
9	['2020-01-01', '2020-05-06']	Comunidades Migrantes
10	['2020-05-06', '2020-09-09']	Comunidad LGBTQ+
11	['2020-09-09', '2021-01-13']	Mujeres
12	['2021-01-13', '2021-05-19']	Pueblos Originarios
13	['2021-05-19', '2021-09-22']	Pueblos Originarios
14	['2021-09-22', '2022-01-26']	Comunidades Migrantes
15	['2022-01-26', '2022-06-01']	Comunidad LGBTQ+

Tabla 8: Asignación de criterios de búsqueda para Recolectores.
Fuente: Elaboración Propia.

De los resultados de la recolección se obtuvieron 2692 *tweets* que poseen odio y resultan ambiguos o interesantes, es decir, poseen casos no triviales de odio o palabras que harían fácil malinterpretar su contenido. Lamentablemente, para este trabajo, por temas de censura por parte de *Twitter*, se perdió alrededor de un $\approx 15\%$ de los *tweets*, recolectados, llegando así a la cifra final de 2288 *tweets* recolectados manualmente, de los cuales también se recolectó su contexto asociado o *thread*.

4.2. Resultados Anotado

En la etapa del anotado se enriquecieron los 2288 *tweets* recolectados manualmente con 2284 *tweets* de los recolectados de forma automática, obteniéndose 4572 *tweets* anotados por 15 anotadores, grupos conformados por 2 mujeres y 13 hombres. En primer lugar, se decidió visualizar cómo se comporta el consenso de acuerdo a las clases, obteniendo los histogramas de la figura 17, en la que se puede observar que al menos la distribución de odio y no odio se encuentra bastante balanceada, siendo esta de 2078 casos de odio y 2494 casos de no odio, donde el consenso entre los 3 anotadores es mayor a o igual a 2. En la figura 17, si miramos la barra amarilla (votos ≥ 2) del tercer panel ("Tweets de Odio por Clase"), vemos que "Comunidades Migrantes" es la más detectada en los *tweets* de Odio. Sin embargo, se observa que las otras clases mantienen distribuciones similares cuando la medida de consenso es utilizando votos ≥ 2 . Ahora repitiendo el mismo procedimiento para el cuarto panel ("Tweets de No Odio por Clase") observando la barra celeste (votos ≥ 2), la comunidad de "Mujeres" se encuentra mucho más representada que las otras

comunidades dentro de los tweets clasificados como no Odio. Resulta importante notar que las clases de las comunidades no son excluyentes entre sí, es decir, es posible que un *tweet* pueda tener más de una clase objetivo. En la tabla 9 se detallan mayormente los resultados que lograron el consenso mayor o igual a 2 para cada una de las clases. De la figura 17, es posible notar cómo los *tweets* clasificados como no odio, si bien lograron consenso, la distribución total de clases es mucho menor que la obtenida en el panel de “*Tweets de Odio por Clase*”. Es decir, será más probable que una comunidad sea claramente identificable en un *tweet* cuando este sea de odio.

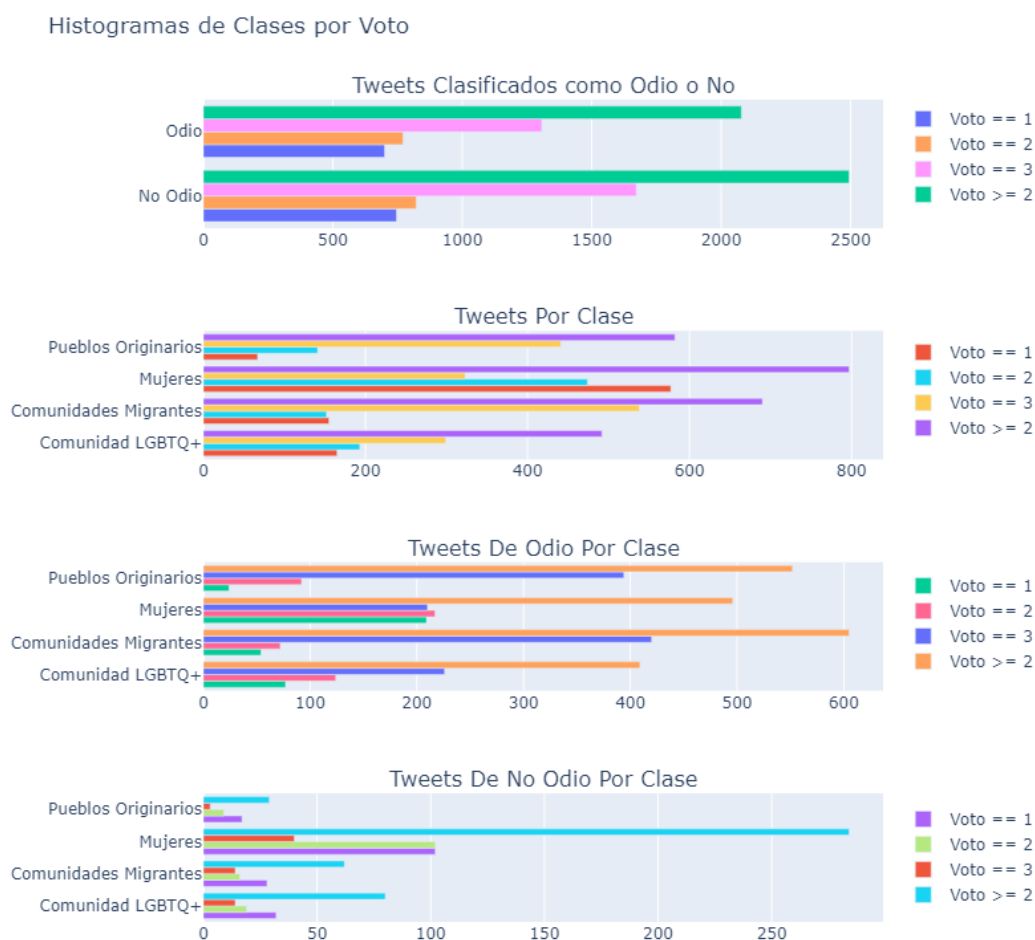


Figura 17: Histogramas de Cantidad de Votos por clases.
Fuente: Elaboración propia.

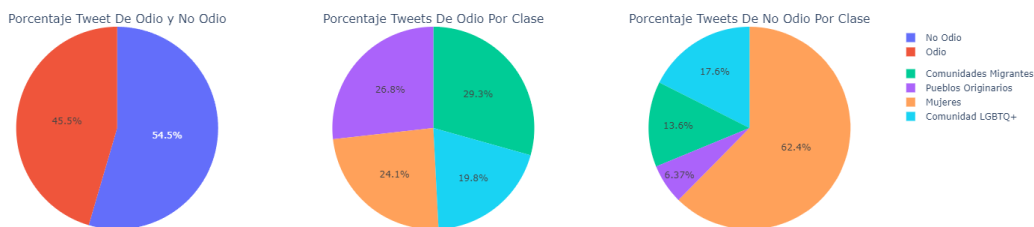


Figura 18: Diagramas circulares de distribución de Clases donde los anotadores lograron consenso (votos ≥ 2).

Fuente: Elaboración propia.



Figura 19: Nube de palabras para la categoría de Odio

Fuente: Elaboración propia

breve introducción al problema al momento de ayudar con la recolección, además de los *guidelines* especificados en 5.6.

Uno de los objetivos que tuvo la utilización del método de recolección manual era evitar que la presencia de palabras ofensivas no fuesen determinantes de la clasificación. Siguiendo esta idea, con tal de visualizar la distribución de palabras en las categorías de Odio y No Odio, se generó un *wordcloud* para cada categoría respectivamente. De estos se observa que, tal y como se esperaba, el balance de palabras ofensivas no se encuentra solamente en Odio, lo cual es justamente lo que se planteaba obtener con tal de que el recurso fuera un aporte para determinar estos casos no triviales.

Grupo	Odio	Comunidad LGBTQ+	Comunidades Migrantes	Mujeres	Pueblos Originarios
grupo 1	0,829805	0,603354	0,931164	0,437693	0,907402
grupo 2	0,631430	0,707811	0,315317	0,637562	0,940599
grupo 3	0,669694	0,588875	0,670913	0,442914	0,818433
grupo 4	0,737578	0,786370	0,722068	0,477485	0,881704
grupo 5	0,560369	0,738637	0,821299	0,606308	0,743526

Tabla 10: Alfa de Krippendorff de grupos por clase
Fuente: Elaboración Propia.

Clase	Anotador 1-2	Anotador 1-3	Anotador 2-3
Comunidad LGBTQ+	0,6989197428590364	0,5463246760230145	0,5283375823364518
Comunidades Migrantes	0,9309181852442494	0,9312570196775218	0,9312990566999396
Mujeres	0,6819755197364306	0,26171660175231093	0,25097406260033495
Pueblos Originarios	0,9052665382145151	0,9044675057039817	0,9125350024707626
Odio	0,819966385457631	0,879916584855101	0,7921698914213595

Tabla 11: Kappa de Cohen por clase del Grupo 1
Fuente: Elaboración Propia.

Clase	Anotador 4-5	Anotador 4-6	Anotador 5-6
Comunidad LGBTQ+	0,6468283152036707	0,5872368782093998	0,8586520916824275
Comunidades Migrantes	0,3584018801410106	0,397838394235718	0,2188841201716738
Mujeres	0,5948036356080254	0,6136709841594097	0,6972404893835782
Pueblos Originarios	0,929846173972767	0,9486982734995889	0,943219196548935
Odio	0,5414320193081256	0,5333906409498738	0,8045783005432776

Tabla 12: Kappa de Cohen por clase del Grupo 2
Fuente: Elaboración Propia.

Clase	Anotador 7-8	Anotador 7-9	Anotador 8-9
Comunidad LGBTQ+	0,47181217489104454	0,7211063203281103	0,588021605085798
Comunidades Migrantes	0,5734857482185274	0,6392020815264527	0,7888631090487239
Mujeres	0,37826263636647806	0,5489165385153083	0,4608589065827001
Pueblos Originarios	0,7663918335296427	0,775627818089522	0,9041803663691874
Odio	0,6008666213441558	0,6859024973492454	0,7237192221934445

Tabla 13: Kappa de Cohen por clase del Grupo 3
Fuente: Elaboración Propia.

Clase	Anotador 10-11	Anotador 10-12	Anotador 11-12
Comunidad LGBTQ+	0,7245042682437315	0,7638784840613600	0,8631694439289407
Comunidades Migrantes	0,7252010032098587	0,6831858797086918	0,7556141979493706
Mujeres	0,37767404372502267	0,4472338686819234	0,585917357195053
Pueblos Originarios	0,8606661590019355	0,8561892132388329	0,9252426502964131
Odio	0,6710015747319564	0,7570569760228338	0,7870851659336265

Tabla 14: Kappa de Cohen por clase del Grupo 4
Fuente: Elaboración Propia.

Clase	Anotador 13-14	Anotador 13-15	Anotador 14-15
Comunidad LGBTQ+	0,7350154105327642	0,7182522862951601	0,7596079585156853
Comunidades Migrantes	0,8910308541574252	0,7929545582970746	0,7737615777139785
Mujeres	0,6588250431754398	0,5021471291215268	0,648073617303861
Pueblos Originarios	0,7110140459994228	0,801366840660819	0,7153742663603837
Odio	0,5433127315718593	0,4957280503156696	0,6353198928353314

Tabla 15: Kappa de Cohen por clase del Grupo 5
Fuente: Elaboración Propia.

De las tablas mostradas es importante recalcar que los grupos 1 y 4 (tablas 11, 14 y 10) fueron los únicos grupos que incluyeron mujeres en su anotado. En efecto, los anotadores 3 y 10 eran mujeres, lo que hace interesante observar la discrepancia en las tablas 11 y 14 precisamente para la identificación de esta comunidad.

4.3. Resultados experimentales

En esta sección se describen y analizan los resultados obtenidos tras la realización de los experimentos descritos en la sección 3.3.1, relacionados con la transferencia de conocimiento entre *datasets*, medición de sesgos y resultado de funcionalidades en *Hatecheck*.

4.3.1. Experimentos de transferencia

A continuación se presentan los resultados de evaluación de los modelos en los experimentos de transferencia de conocimiento.

Entrenamiento	Evaluación	Accuracy	Precision	Recall	F1-score	AUC
es	en	0.568750	0.567634	0.577000	0.572279	0.601915
	cl ₁	0.635957	0.632613	0.648566	0.640490	0.760301
	cl ₂	0.509457	0.509267	0.519683	0.514423	0.522938
en	es	0.542787	0.543169	0.538361	0.540754	0.551567
	cl ₁	0.682733	0.685525	0.675208	0.680328	0.720489
	cl ₂	0.539587	0.539982	0.534638	0.537297	0.538476
cl ₁	es	0.557377	0.557947	0.552459	0.555189	0.591628
	en	0.570042	0.570660	0.565667	0.568152	0.628883
	cl ₂	0.528150	0.528444	0.522982	0.525699	0.531031
cl ₂	es	0.591148	0.590495	0.594754	0.592617	0.638031
	en	0.584375	0.583809	0.587750	0.585773	0.628007
	cl ₁	0.761694	0.757170	0.770490	0.763772	0.805874

Tabla 16: Resultados transferencia para ACL19. Mejores resultados para transferencia al: Inglés, Español y Chileno resaltados en verde, azul y rosado respectivamente.

Fuente: Elaboración propia

Entrenamiento	Evaluación	Accuracy	Precision	Recall	F1-Score	AUC
es	en	0.648833	0.597050	0.505651	0.547563	0.629137
	cl ₁	0.695749	0.096752	0.447077	0.159078	0.579967
	cl ₂	0.610952	0.583243	0.516627	0.547917	0.603378
en	es	0.640328	0.562670	0.576083	0.569297	0.630771
	cl ₁	0.783608	0.095727	0.279621	0.142627	0.548951
	cl ₂	0.579063	0.553631	0.400482	0.464765	0.564723
cl ₁	es	0.588852	0.505325	0.169646	0.254015	0.526492
	en	0.590167	0.533016	0.200079	0.290946	0.536507
	cl ₂	0.627007	0.691996	0.329157	0.446114	0.603090
cl ₂	es	0.589836	0.504065	0.369487	0.426410	0.557057
	en	0.585917	0.512785	0.294269	0.373945	0.545798
	cl ₁	0.540675	0.091502	0.687204	0.161500	0.608899

Tabla 17: Resultados transferencia para ECML20. Mejores resultados para transferencia al: Inglés, Español y Chileno resaltados en verde, azul y rosado respectivamente.

Fuente: Elaboración propia

Entrenamiento	Evaluación	Accuracy	Precision	Recall	F1-Score	AUC
es	en	0.688500	0.673954	0.501289	0.574937	0.662747
	cl ₁	0.812691	0.122895	0.311216	0.176208	0.579203
	cl ₂	0.646360	0.636790	0.523855	0.574828	0.636523
en	es	0.714918	0.720772	0.504569	0.593597	0.683627
	cl ₁	0.849502	0.119497	0.210111	0.152348	0.551800
	cl ₂	0.668353	0.759378	0.400000	0.523990	0.646804
cl ₁	es	0.588525	0.577778	0.010330	0.020297	0.502513
	en	0.580083	0.527778	0.007535	0.014858	0.501324
	cl ₂	0.554212	0.807692	0.030361	0.058523	0.512147
cl ₂	es	0.674426	0.628821	0.514899	0.566186	0.650695
	en	0.699667	0.686544	0.525084	0.595056	0.675651
	cl ₁	0.646227	0.109923	0.633491	0.187339	0.640297

Tabla 18: Resultados transferencia para EMNLP20. Mejores resultados para transferencia al: Inglés, Español y Chileno resaltados en verde, azul y rosado respectivamente.

Fuente: Elaboración propia

Entrenamiento	Evaluación	Best AUC	AVG AUC	STD AUC
es	en	0.66	0.65	0.02
	cl ₁	0.76	0.64	0.10
	cl ₂	0.64	0.59	0.06
en	es	0.68	0.62	0.07
	cl ₁	0.72	0.61	0.10
	cl ₂	0.65	0.56	0.08
cl ₁	es	0.59	0.54	0.05
	en	0.62	0.55	0.06
	cl ₂	0.60	0.55	0.05
cl ₂	es	0.65	0.62	0.05
	en	0.68	0.62	0.07
	cl ₁	0.81	0.69	0.11

Tabla 19: Resultados de transferencia consolidados para AUC. Mejores resultados consolidados y promedio para transferencia al: Inglés, Español y Chileno resaltados en verde, azul y rosado respectivamente.

Fuente: Elaboración propia

En las tablas 19 y 20 se observan los resultados de transferencia consolidados donde se dejó el mejor resultado para la métrica AUC. En cuanto a los resultados de los modelos para esta métrica, el modelo con los mejores valores fue EMNLP20 con 6 (es → en, es → cl₂, en → es, en → cl₂, cl₂ → es, cl₂ → en) de los mejores resultados, seguido por ACL19 con 5 (es → cl₁, en → cl₁, cl₁ → es, cl₁ → en, cl₂ → cl₁) de los mejores resultados y finalmente, ECML20 con 1 (cl₁ → cl₂) de los mejores resultados. Para la métrica de valor-F1 solo 3 de los mejores resultados pertenecen a EMNLP20 mientras que los 9 restantes pertenecen a ACL19. En base en los resultados expuestos en ambas tablas, se nota cómo la transferencia del conjunto de

Entrenamiento	Evaluación	Best F1-Score	AVG F1-Score	STD F1-Score
es	en	0.60	0.58	0.03
	cl ₁	0.64	0.33	0.27
	cl ₂	0.57	0.55	0.03
en	es	0.59	0.57	0.03
	cl ₁	0.68	0.33	0.31
	cl ₂	0.52	0.49	0.03
cl ₁	es	0.56	0.28	0.27
	en	0.59	0.30	0.29
	cl ₂	0.53	0.34	0.25
cl ₂	es	0.59	0.53	0.09
	en	0.61	0.53	0.13
	cl ₁	0.76	0.37	0.34

Tabla 20: Resultados de transferencia consolidados y promedio para valor F1. Mejores resultados para transferencia al: Inglés, Español y Chileno resaltados en verde, azul y rosado respectivamente.

Fuente: Elaboración propia

datos del presente trabajo (cl₂) hacia el español de Basile *et al.* es la mejor utilizando valor-F1 y es la segunda mejor utilizando AUC en cuanto a transferencia de conocimientos, con tan solo un 3 % de diferencia con el mejor valor. En el caso de la transferencia al inglés de Basile *et al.*, los datos generados (cl₂) logran el mejor nivel de transferencia para ambas métricas, superando el estado del arte, notándose también que como es de esperarse, la transferencia hacia el chileno de Arango Monnar *et al.* es la más alta de todos los *datasets* utilizando cl₂, logrando una vez más superar al estado del arte presentado en Arango Monnar *et al.*.

Contrastando una vez más los resultados expuestos en las tablas 19 y 20, pero ahora solo enfocándonos en cl₁ y cl₂. Se observa claramente que para ambas métricas cl₂ siempre genera mejores transferencias que cl₁ a otros recursos si se considera el mejor algoritmo. Esto queda en evidencia al observar que cl₂ logra mejoras del 6 % en transferencias para “es” y “en” en la métrica AUC con respecto a cl₁. Mientras que para la métrica de valor F1, cl₂ logra mejoras del 3 % y 2 % en transferencias hacia “es” y “en” respectivamente con respecto a cl₁.

Lo anterior resulta altamente relevante, pues si bien el *dataset* generado contiene alrededor de un 50 % de datos menos, este posee un balance de clases mucho mayor en comparación a Arango Monnar *et al.* en la categoría de *Hate Speech* o Discurso de Odio, donde Arango Monnar *et al.* reportan aproximadamente un 6 % de datos pertenecientes a esta clase. Sin embargo, al evaluar la transferencia hacia cl₂, en la que se esperaba que cl₁ presentara el mayor nivel de transferencia, tal y como ocurre en el caso contrario, se observó que esto no se cumple, pues cl₁ obtuvo el peor desempeño entre todos los conjuntos para la métrica AUC. Esta anomalía, como se mencionó con anterioridad, posiblemente se encuentre ligada al alto desbalance de clases para la categoría de Discurso de Odio, en la que cl₁ presenta solo un 6 % de positivos, mientras que cl₂ presenta un 45,5 %.

Al analizar las tablas 16,17 y 18 se observa que la anomalía de transferencia mencionada ($cl_2 \rightarrow cl_1$) es evidenciable para todos los modelos al utilizar la métrica AUC.

Evaluando el mejor caso de transferencia de $cl_1 \rightarrow cl_2$ (ver tabla 19), se determina que si bien esto puede resultar indicativo de transferencia exitosa a cl_2 , el que el resultado sea tan bajo y que la diferencia entre el mejor resultado de cl_1 a cl_2 y cl_2 a cl_1 sea de $\approx 0,2$ para AUC y $\approx 0,2$ para F1-Score (con nuestros datos logrando los mejores resultados de transferencia), parece indicar un problema relacionado con la distribución de clases presente en cl_1 .

Observamos también el fenómeno donde la transferencia de conocimiento en ambas tablas (19, 20), es siempre menor cuando cl_2 es el conjunto de prueba. En este trabajo se plantea que este fenómeno se encuentra relacionado con las formas de *hate speech* más sutiles y complejas que se obtuvieron gracias a la recolección manual de *tweets*. La presencia de estas formas de mayor complejidad añade riqueza al recurso presentado, sin embargo, al mismo tiempo estas formas aumentan la dificultad de otros recursos para generalizar correctamente en estos casos, lo cual resulta indicativo de por qué nuestro recurso es capaz de generalizar correctamente a corpus con casos más simples.

4.3.2. Experimentos de Sesgo

A continuación se presentan los resultados de realizar la evaluación de métricas de sesgos propuestas por Borkan *et al.*

Entrenamiento	Comunidad	Subgroup AUC	BPSN AUC	BNSP AUC
es	black people	0.465005	0.483084	0.452336
	gay people	0.486269	0.474703	0.478238
	trans people	0.501103	0.486551	0.479263
	indigenous people	0.507635	0.509501	0.462781
	jews	0.476461	0.497943	0.447882
	disabled people	0.488704	0.493554	0.462261
	women	0.514441	0.498747	0.478792
en	black people	0.504094	0.510443	0.491063
	gay people	0.487122	0.495147	0.491469
	trans people	0.479217	0.491719	0.488166
	indigenous people	0.489078	0.485163	0.50263
	jews	0.499287	0.505635	0.491558
	disabled people	0.495028	0.498575	0.494771
	women	0.500901	0.468794	0.528262
cl ₁	black people	0.5	0.500218	0.497522
	gay people	0.488974	0.487705	0.50026
	trans people	0.495902	0.496125	0.497982
	indigenous people	0.5	0.500223	0.497477
	jews	0.5	0.500218	0.497522
	disabled people	0.5	0.500223	0.497477
	women	0.495902	0.496124	0.497982
cl ₂	black people	0.512996	0.513929	0.506902
	gay people	0.506366	0.458115	0.554599
	trans people	0.512212	0.490825	0.528225
	indigenous people	0.502538	0.521448	0.490934
	jews	0.504451	0.520697	0.493054
	disabled people	0.505076	0.521001	0.493472
	women	0.501309	0.521556	0.489704
cl ₁ + cl ₂	black people	0.5	0.51329	0.48662
	gay people	0.465486	0.389568	0.576142
	trans people	0.501269	0.513405	0.487646
	indigenous people	0.5	0.513628	0.486377
	jews	0.5	0.51329	0.48662
	disabled people	0.5	0.513628	0.486377
	women	0.5	0.513556	0.486377

Tabla 21: ACL19: Resultados de métricas de sesgo por comunidad para *Hatecheck*. Mejores y peores resultados por métrica resaltados en verde y rosado respectivamente para cada *dataset*. En los casos donde el peor o mejor resultado esta repetido entre comunidades éste no es resaltado.

Fuente: Elaboración propia

Entrenamiento	Comunidad	Subgroup AUC	BPSN AUC	BNSP AUC
es	black people	0.567679	0.406687	0.769325
	gay people	0.606516	0.714232	0.50768
	trans people	0.594907	0.692176	0.519085
	indigenous people	0.632333	0.56991	0.666981
	jews	0.626812	0.624064	0.611026
	disabled people	0.611197	0.727867	0.498552
	women	0.586387	0.467778	0.726902
en	black people	0.581289	0.488667	0.650197
	gay people	0.567092	0.571672	0.558118
	trans people	0.53106	0.611754	0.489866
	indigenous people	0.551469	0.588882	0.528686
	jews	0.557621	0.607929	0.515162
	disabled people	0.538966	0.615183	0.493168
	women	0.598125	0.432142	0.718032
cl ₁	black people	0.548662	0.576179	0.530579
	gay people	0.561122	0.564398	0.552536
	trans people	0.535928	0.592933	0.504328
	indigenous people	0.57065	0.384385	0.732372
	jews	0.570261	0.539926	0.583998
	disabled people	0.519036	0.610366	0.473627
	women	0.539546	0.577632	0.521755
cl ₂	black people	0.546764	0.759884	0.425062
	gay people	0.641404	0.600902	0.658587
	trans people	0.641487	0.644267	0.617244
	indigenous people	0.562994	0.388172	0.796514
	jews	0.647412	0.695481	0.573438
	disabled people	0.62992	0.496887	0.748367
	women	0.607265	0.726769	0.509376
cl ₁ + cl ₂	black people	0.563085	0.711518	0.464177
	gay people	0.603354	0.497802	0.703584
	trans people	0.585649	0.606957	0.584616
	indigenous people	0.544354	0.329854	0.814891
	jews	0.600234	0.684008	0.522539
	disabled people	0.594137	0.619922	0.579295
	women	0.571689	0.642772	0.538435

Tabla 22: ECML20: Resultados de métricas de sesgo por comunidad para *Hatecheck*. Mejores y peores resultados por métrica resaltados en verde y rosado respectivamente para cada *dataset*. En los casos donde el peor o mejor resultado esta repetido entre comunidades éste no es resaltado.

Fuente: Elaboración propia

Entrenamiento	Comunidad	Subgroup AUC	BPSN AUC	BNSP AUC
es	black people	0.511812	0.324024	0.885545
	gay people	0.659711	0.834513	0.523291
	trans people	0.688025	0.815068	0.565414
	indigenous people	0.727885	0.711653	0.697332
	jews	0.670595	0.576391	0.779166
	disabled people	0.67184	0.760079	0.604463
	women	0.644709	0.514918	0.816683
en	black people	0.737503	0.637355	0.73087
	gay people	0.604747	0.693769	0.564004
	trans people	0.618124	0.681775	0.586586
	indigenous people	0.599963	0.699432	0.554616
	jews	0.660294	0.670747	0.632449
	disabled people	0.573999	0.694363	0.537858
	women	0.770106	0.468659	0.918081
cl ₁	black people	0.501484	0.500000	0.501484
	gay people	0.500000	0.500223	0.500000
	trans people	0.500000	0.500223	0.500000
	indigenous people	0.500000	0.500223	0.500000
	jews	0.500000	0.500218	0.500000
	disabled people	0.500000	0.500223	0.500000
	women	0.500000	0.500222	0.500000
cl ₂	black people	0.684675	0.770251	0.609425
	gay people	0.501539	0.414169	0.797846
	trans people	0.653345	0.652537	0.696492
	indigenous people	0.586336	0.500458	0.785981
	jews	0.624943	0.497826	0.820157
	disabled people	0.685882	0.849243	0.535707
	women	0.717878	0.766233	0.641349
cl ₁ + cl ₂	black people	0.654831	0.679577	0.584326
	gay people	0.544853	0.327975	0.827934
	trans people	0.603291	0.423724	0.785108
	indigenous people	0.599692	0.385799	0.818332
	jews	0.575311	0.746391	0.451774
	disabled people	0.511421	0.76765	0.380241
	women	0.544782	0.731854	0.441218

Tabla 23: EMNLP20: Resultados de métricas de sesgo por comunidad para *Hatecheck*. Mejores y peores resultados por métrica resaltados en verde y rosado respectivamente para cada *dataset*. En los casos donde el peor o mejor resultado esta repetido entre comunidades éste no es resaltado.

Fuente: Elaboración propia

Entrenamiento	GMB Subgroup AUC	GMB BPSN AUC	GMB BNSP AUC
es	0.48975	0.491316	0.464981
en	0.493121	0.492604	0.497356
cl ₁	0.497164	0.497148	0.498027
cl ₂	0.506318	0.503379	0.505383
cl ₁ + cl ₂	0.494242	0.478326	0.495006

Tabla 24: ACL19: Resultados de métricas de sesgo acumuladas para *Hatecheck*. Mejores y peores resultados por métrica resaltados en verde y rosado respectivamente.

Fuente: Elaboración propia

Entrenamiento	GMB Subgroup AUC	GMB BPSN AUC	GMB BNSP AUC
es	0.601449	0.523345	0.569736
en	0.558345	0.529736	0.539042
cl ₁	0.547545	0.504037	0.534048
cl ₂	0.603355	0.52047	0.546218
cl ₁ + cl ₂	0.578193	0.461843	0.555172

Tabla 25: ECML20: Resultados de métricas de sesgo acumuladas para *Hatecheck*. Mejores y peores resultados por métrica resaltados en verde y rosado respectivamente.

Fuente: Elaboración propia

Entrenamiento	GMB Subgroup AUC	GMB BPSN AUC	GMB BNSP AUC
es	0.628262	0.46051	0.632194
en	0.633511	0.60511	0.600825
cl ₁	0.50021	0.50019	0.50021
cl ₂	0.609366	0.531649	0.652904
cl ₁ + cl ₂	0.566642	0.430491	0.484545

Tabla 26: EMNLP20: Resultados de métricas de sesgo acumuladas para *Hatecheck*. Mejores y peores resultados por métrica resaltados en verde y rosado respectivamente.

Fuente: Elaboración propia

A partir de los resultados consolidados presentados en la tabla 27, se pueden observar los resultados de las métricas de sesgos asociadas a cada una de las comunidades del *dataset* de *Hatecheck*, tomando en cuenta que, tal y como relatan en Röttger *et al.*, ellos utilizaron español de España y de Argentina. Resulta importante indicarlo previo al análisis de los resultados, puesto que es indicativo de que nuestros datos y los de Arango Monnar *et al.* podrían encontrarse en desventaja con respecto a Basile *et al.*

Para evaluar los desempeños de forma global en relación con las comunidades se recurrió a la métrica utilizada por *Jigsaw* en el desafío *Kaggle* titulado “*Jigsaw Unintended Bias in Toxicity Classification*”¹¹, donde calculan la media de potencias de las comunidades con tal

¹¹<https://www.kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification/overview/faq>

Datos	Comunidad	Best Subgroup AUC	AVG Subgroup AUC	Best BPSN AUC	AVG BPSN AUC	Best BNSP AUC	AVG BNSP AUC
es	black people	0.57	0.51	0.48	0.40	0.89	0.70
	gay people	0.66	0.58	0.83	0.67	0.52	0.50
	trans people	0.69	0.59	0.82	0.66	0.57	0.52
	indigenous people	0.73	0.62	0.71	0.60	0.70	0.61
	jews	0.67	0.59	0.62	0.57	0.78	0.61
	disabled people	0.67	0.59	0.76	0.66	0.60	0.52
	women	0.64	0.58	0.51	0.49	0.82	0.67
en	black people	0.74	0.61	0.64	0.55	0.73	0.62
	gay people	0.60	0.55	0.69	0.59	0.56	0.54
	trans people	0.62	0.54	0.68	0.60	0.59	0.52
	indigenous people	0.60	0.55	0.70	0.59	0.55	0.53
	jews	0.66	0.57	0.67	0.59	0.63	0.55
	disabled people	0.57	0.54	0.69	0.60	0.54	0.51
	women	0.77	0.62	0.47	0.46	0.92	0.72
cl ₁	black people	0.55	0.52	0.58	0.53	0.53	0.51
	gay people	0.56	0.52	0.56	0.52	0.55	0.52
	trans people	0.54	0.51	0.59	0.53	0.50	0.50
	indigenous people	0.57	0.52	0.50	0.46	0.73	0.58
	jews	0.57	0.52	0.54	0.51	0.58	0.53
	disabled people	0.52	0.51	0.61	0.54	0.50	0.49
	women	0.54	0.51	0.58	0.52	0.52	0.51
cl ₂	black people	0.68	0.58	0.77	0.68	0.61	0.51
	gay people	0.64	0.55	0.60	0.49	0.80	0.67
	trans people	0.65	0.60	0.65	0.60	0.70	0.61
	indigenous people	0.59	0.55	0.52	0.47	0.80	0.69
	jews	0.65	0.59	0.70	0.57	0.82	0.63
	disabled people	0.69	0.61	0.85	0.62	0.75	0.59
	women	0.72	0.61	0.77	0.67	0.64	0.55
cl ₁ + cl ₂	black people	0.65	0.57	0.71	0.63	0.58	0.51
	gay people	0.60	0.54	0.50	0.41	0.83	0.70
	trans people	0.60	0.56	0.61	0.51	0.79	0.62
	indigenous people	0.60	0.55	0.51	0.41	0.82	0.71
	jews	0.60	0.56	0.75	0.65	0.52	0.49
	disabled people	0.59	0.54	0.77	0.63	0.58	0.48
	women	0.57	0.54	0.73	0.63	0.54	0.49

Tabla 27: Resultados consolidados de métricas de sesgo por comunidad para *Hatecheck*. Considerando el mejor algoritmo multi-lingua para cada corpus. Mejores y peores resultados por métrica resaltados en verde y rosado respectivamente para cada comunidad.

Fuente: Elaboración propia

Entrenamiento	Best GMB Subgroup AUC	AVG GMB Subgroup AUC	Best GMB BPSN AUC	AVG GMB BPSN AUC	Best GMB BNSP AUC	AVG GMB BNSP AUC
es	0.63	0.57	0.52	0.49	0.63	0.56
en	0.63	0.56	0.61	0.54	0.60	0.55
cl ₁	0.55	0.51	0.50	0.50	0.53	0.51
cl ₂	0.61	0.57	0.53	0.52	0.65	0.57
cl ₁ + cl ₂	0.58	0.55	0.48	0.46	0.56	0.51

Tabla 28: Resultados consolidados de métricas de sesgo acumuladas para *Hatecheck*. Considerando el mejor algoritmo multi-lingua para cada corpus. Mejores y peores resultados por métrica resaltados en verde y rosado respectivamente.

Fuente: Elaboración propia

de medir el sesgo asociado a éstas. Observando la tabla 28, es posible evidenciar que nuestro *dataset* logra valores competitivos en 2 de las 3 métricas, logrando el mejor resultado en GMB BPSN AUC. Los resultados demuestran que sin ninguna clase de preprocesamiento de los datos o un entrenamiento específico para distinguir los grupos considerados en *Hatecheck*, el recurso propuesto permite obtener resultados competitivos en términos de GMB Subgroup AUC, alcanzando al mismo nivel que Basile *et al.* si consideramos el GMB promedio. Por otro lado, el corpus propuesto permite lograr la mejor capacidad de todos los datos para distinguir falsos negativos pertenecientes a los subgrupos y, si bien, éste no permite obtener resultados competitivos para BPSN AUC, nuestro recurso se mantiene segundo en desempeño.

Contrastando los resultados obtenidos por cl₂ con los obtenidos por cl₁ (ver tabla 28), este último obtiene el peor resultado en todas las métricas evaluadas. En comparación, cl₂ presenta mejoras del 6 % para Best y AVG GMB Subgroup AUC, 3 % para Best GMB BPSN AUC, 2 % para AVG GMB BPSN AUC, 12 % para Best GMB BNSP AUC y 6 % para AVG GMB BNSP AUC. Mostrando que el corpus generado es capaz de una mejor detección de sesgos que cl₁.

4.3.3. Experimentos *Hatecheck*

En esta sección se exponen los resultados de realizar la evaluación de los modelos ACL19, ECML20 y EMNLP20 entrenados en los *dataset* en inglés y español de Basile *et al.*, el chileno de Arango Monnar *et al.* denotado como “cl₁” y el nuestro denotado como “cl₂”, sobre el conjunto de evaluación multilingüe de *Hatecheck*.

	es	en	cl ₁	cl ₂	cl ₁ + cl ₂	Δcl^{12}
derog_neg_emote_h	0.792857	0.000000	0.000000	0.021429	0.007143	0.021429
derog_neg_attrib_h	0.392857	0.000000	0.000000	0.035714	0.014286	0.035714
derog_dehum_h	0.150000	0.000000	0.000000	0.085714	0.014286	0.085714
derog_impl_h	0.800000	0.050000	0.000000	0.114286	0.100000	0.114286
threat_dir_h	0.928571	0.035714	0.000000	0.007143	0.064286	0.007143
threat_norm_h	0.972414	0.034483	0.000000	0.000000	0.013793	0.000000
slur_h	0.900000	0.040000	0.000000	0.000000	0.000000	0.000000
profanity_h	0.564286	0.000000	0.000000	0.085714	0.035714	0.085714
profanity_nh	0.050000	0.980000	1.000000	0.970000	1.000000	-0.030000
ref_subs_clause_h	0.448276	0.020690	0.000000	0.034483	0.013793	0.034483
ref_subs_sent_h	0.462069	0.034483	0.000000	0.034483	0.013793	0.034483
negate_pos_h	0.531034	0.075862	0.000000	0.124138	0.055172	0.124138
negate_neg_nh	0.335714	0.964286	1.000000	0.978571	0.950000	-0.021429
phrase_question_h	0.371429	0.007143	0.007143	0.000000	0.021429	-0.007143
phrase_opinion_h	0.510345	0.000000	0.000000	0.020690	0.013793	0.020690
ident_neutral_nh	0.114286	1.000000	1.000000	0.942857	0.935714	-0.057143
ident_pos_nh	0.338095	0.985714	1.000000	0.985714	0.971429	-0.014286
counter_quote_nh	0.646341	0.896341	0.969512	0.963415	0.975610	-0.006098
counter_ref_nh	0.329268	0.939024	1.000000	1.000000	0.993902	0.000000
target_obj_nh	0.323077	1.000000	1.000000	1.000000	1.000000	0.000000
target_indiv_nh	0.030769	1.000000	1.000000	1.000000	1.000000	0.000000
target_group_nh	0.415385	1.000000	1.000000	1.000000	1.000000	0.000000
spell_char_swap_h	0.592857	0.000000	0.000000	0.042857	0.021429	0.042857
spell_char_del_h	0.750000	0.064286	0.000000	0.014286	0.028571	0.014286
spell_space_del_h	0.768293	0.048780	0.000000	0.018293	0.006098	0.018293
spell_space_add_h	0.560976	0.079268	0.000000	0.024390	0.000000	0.024390
spell_leet_h	0.710059	0.017751	0.000000	0.017751	0.005917	0.017751
AVG	0.51	0.34	0.33	0.35	0.34	
STD	0.26	0.46	0.48	0.45	0.46	

Tabla 29: ACL19: Resultados de Evaluación en *Hatecheck* utilizando la exactitud como métrica. Mejores y peores resultados por fila resaltados en verde y rosado respectivamente.

Fuente: Elaboración propia

¹² $\Delta cl = cl_2 - cl_1$

	es	en	cl ₁	cl ₂	cl ₁ + cl ₂	Δcl^{12}
derog_neg_emote_h	0.307143	0.014286	0.107143	0.342857	0.214286	0.235714
derog_neg_attrib_h	0.650000	0.200000	0.450000	0.628571	0.592857	0.178571
derog_dehum_h	0.635714	0.264286	0.392857	0.592857	0.521429	0.200000
derog_impl_h	0.485714	0.407143	0.014286	0.242857	0.171429	0.228571
threat_dir_h	0.450000	0.300000	0.107143	0.392857	0.292857	0.285714
threat_norm_h	0.634483	0.668966	0.131034	0.779310	0.496552	0.648276
slur_h	0.486667	0.420000	0.060000	0.333333	0.393333	0.273333
profanity_h	0.492857	0.485714	0.200000	0.550000	0.500000	0.350000
profanity_nh	0.970000	0.870000	1.000000	0.980000	0.980000	-0.020000
ref_subs_clause_h	0.441379	0.296552	0.331034	0.572414	0.634483	0.241379
ref_subs_sent_h	0.703448	0.324138	0.406897	0.724138	0.627586	0.317241
negate_pos_h	0.331034	0.082759	0.055172	0.303448	0.213793	0.248276
negate_neg_nh	0.564286	0.607143	0.835714	0.571429	0.685714	-0.264286
phrase_question_h	0.485714	0.264286	0.221429	0.542857	0.471429	0.321429
phrase_opinion_h	0.427586	0.227586	0.213793	0.579310	0.462069	0.365517
ident_neutral_nh	0.750000	0.971429	0.971429	0.871429	0.892857	-0.100000
ident_pos_nh	0.742857	0.952381	0.909524	0.790476	0.838095	-0.119048
counter_quote_nh	0.859756	0.865854	0.859756	0.865854	0.743902	0.006098
counter_ref_nh	0.743902	0.707317	0.945122	0.579268	0.609756	-0.365854
target_obj_nh	0.830769	0.830769	0.923077	0.815385	0.907692	-0.107692
target_indiv_nh	0.784615	0.938462	1.000000	0.876923	0.892308	-0.123077
target_group_nh	0.461538	0.769231	0.892308	0.507692	0.892308	-0.384615
spell_char_swap_h	0.500000	0.285714	0.228571	0.521429	0.378571	0.292857
spell_char_del_h	0.407143	0.314286	0.135714	0.378571	0.321429	0.242857
spell_space_del_h	0.445122	0.317073	0.213415	0.536585	0.396341	0.323171
spell_space_add_h	0.207317	0.054878	0.091463	0.237805	0.286585	0.146341
spell_leet_h	0.420118	0.301775	0.147929	0.485207	0.325444	0.337278
AVG	0.56	0.47	0.44	0.58	0.55	
STD	0.19	0.30	0.37	0.21	0.24	

Tabla 30: ECML20: Resultados de Evaluación en *Hatecheck* utilizando la exactitud como métrica. Mejores y peores resultados por fila resaltados en verde y rosado respectivamente.

Fuente: Elaboración propia

	es	en	cl ₁	cl ₂	cl ₁ + cl ₂	Δcl^{12}
black people	0.456845	0.208892	0.190826	0.227895	0.190826	0.037069
gay people	0.480396	0.201418	0.190210	0.292106	0.297154	0.101896
trans people	0.493080	0.191864	0.189953	0.259971	0.194054	0.070018
indigenous people	0.494089	0.220121	0.191223	0.196874	0.191223	0.005652
jews	0.464682	0.207335	0.190826	0.200705	0.190826	0.009880
disabled people	0.478661	0.210913	0.191223	0.202479	0.191223	0.011257
women	0.505016	0.267253	0.193600	0.197811	0.194888	0.004211

Tabla 32: ACL19: Resultados de Evaluación en *Hatecheck* para comunidades utilizando valor F1 Macro como métrica. Mejores y peores resultados por fila resaltados en verde y rosado respectivamente.

Fuente: Elaboración propia

	es	en	cl ₁	cl ₂	cl ₁ + cl ₂	Δcl^{12}
derog_neg_emote_h	0.585714	0.235714	0.000000	0.685714	0.478571	0.685714
derog_neg_attrib_h	0.757143	0.335714	0.007143	0.892857	0.600000	0.885714
derog_dehum_h	0.807143	0.492857	0.000000	0.828571	0.671429	0.828571
derog_impl_h	0.614286	0.314286	0.000000	0.621429	0.407143	0.621429
threat_dir_h	0.757143	0.557143	0.000000	0.707143	0.450000	0.707143
threat_norm_h	0.793103	0.868966	0.000000	0.924138	0.517241	0.924138
slur_h	0.533333	0.306667	0.000000	0.526667	0.360000	0.526667
profanity_h	0.728571	0.428571	0.000000	0.892857	0.564286	0.892857
profanity_nh	0.950000	0.930000	1.000000	1.000000	1.000000	0.000000
ref_subs_clause_h	0.634483	0.344828	0.000000	0.813793	0.531034	0.813793
ref_subs_sent_h	0.641379	0.303448	0.000000	0.889655	0.600000	0.889655
negate_pos_h	0.565517	0.310345	0.000000	0.606897	0.406897	0.606897
negate_neg_nh	0.642857	0.907143	1.000000	0.457143	0.692857	-0.542857
phrase_question_h	0.542857	0.335714	0.000000	0.771429	0.450000	0.771429
phrase_opinion_h	0.613793	0.413793	0.000000	0.827586	0.517241	0.827586
ident_neutral_nh	0.750000	0.978571	1.000000	0.757143	0.885714	-0.242857
ident_pos_nh	0.557143	0.866667	1.000000	0.604762	0.695238	-0.395238
counter_quote_nh	0.646341	0.908537	1.000000	0.414634	0.597561	-0.585366
counter_ref_nh	0.713415	0.963415	1.000000	0.426829	0.615854	-0.573171
target_obj_nh	0.907692	0.907692	1.000000	0.984615	1.000000	-0.015385
target_indiv_nh	0.876923	0.815385	1.000000	0.876923	0.953846	-0.123077
target_group_nh	0.784615	0.861538	1.000000	0.907692	0.923077	-0.092308
spell_char_swap_h	0.721429	0.264286	0.000000	0.735714	0.435714	0.735714
spell_char_del_h	0.607143	0.314286	0.000000	0.714286	0.357143	0.714286
spell_space_del_h	0.658537	0.420732	0.000000	0.750000	0.439024	0.750000
spell_space_add_h	0.585366	0.207317	0.000000	0.487805	0.243902	0.487805
spell_leet_h	0.704142	0.355030	0.000000	0.692308	0.295858	0.692308
AVG	0.69	0.55	0.33	0.73	0.58	
STD	0.11	0.28	0.48	0.17	0.21	

Tabla 31: EMNLP20: Resultados de Evaluación en *Hatecheck* utilizando la exactitud como métrica. Mejores y peores resultados por fila resaltados en verde y rosado respectivamente.

Fuente: Elaboración propia

En la tabla 35 se observan los resultados consolidados de los modelos para cada *dataset* agrupados por funcionalidad, además de la columna de “XLMT 2021” [55] a modo de comparación con el *benchmark* obtenidos por Röttger *et al.* 2022. En la tabla mencionada anteriormente es posible evidenciar que nuestro *dataset* logra el mejor resultado en 10 de 27 funcionalidades al comparar todos los recursos, incluyendo al *benchmark*. Realizando la misma comparativa pero solo entre cl₂ y el *benchmark* (XLMT 2021), el corpus generado en este trabajo logra superar el estado del arte (XLMT 2021) en 16 de las 27 funcionalidades, obteniendo su peor resultado en la funcionalidad 26 con un 0,49, muy inferior al 0,71 del estado del arte. A modo de resumen, nuestros datos superan el estado del arte en 16 funcionalidades, obtienen resultados cercanos ($\Delta \leq 7\%$), pero inferiores al estado del arte en 5 funcionalidades (F3, F4, F10, F12, F25) y resultados inferiores en las 6 funcionalidades

	es	en	cl ₁	cl ₂	cl ₁ + cl ₂	Δcl^{12}
black people	0.773700	0.585742	0.250639	0.201592	0.252577	-0.049048
gay people	0.437984	0.424474	0.310924	0.675969	0.686567	0.365045
trans people	0.454545	0.258065	0.174713	0.622735	0.515098	0.448022
indigenous people	0.688923	0.357430	0.628125	0.787724	0.776923	0.159599
jews	0.615385	0.328502	0.385151	0.559006	0.404651	0.173855
disabled people	0.422091	0.269807	0.073350	0.768176	0.509874	0.694826
women	0.734286	0.676971	0.224256	0.430584	0.426036	0.206327

Tabla 33: ECML20: Resultados de Evaluación en *Hatecheck* para comunidades utilizando valor F1 Macro como métrica. Mejores y peores resultados por fila resaltados en verde y rosado respectivamente.

Fuente: Elaboración propia

	es	en	cl ₁	cl ₂	cl ₁ + cl ₂	Δcl^{12}
black people	0.863459	0.700565	0.005917	0.711986	0.57377	0.706069
gay people	0.524954	0.367010	0.000000	0.821596	0.804938	0.821596
trans people	0.604167	0.420635	0.000000	0.790257	0.788991	0.790257
indigenous people	0.780347	0.343816	0.000000	0.840964	0.816121	0.840964
jews	0.834337	0.523810	0.000000	0.881633	0.27551	0.881633
disabled people	0.651466	0.292473	0.000000	0.620102	0.044665	0.620102
women	0.850633	0.890339	0.000000	0.760606	0.240363	0.760606

Tabla 34: EMNLP20: Resultados de Evaluación en *Hatecheck* para comunidades utilizando valor F1 Macro como métrica. Mejores y peores resultados por fila resaltados en verde y rosado respectivamente.

Fuente: Elaboración propia

restantes.

En promedio, se observa que el desempeño de exactitud de nuestros datos es superior al *benchmark* (columna “XLMT 2021”), además de ser el mejor entre los conjuntos de datos utilizados, seguido en forma descendente por español (es), inglés (en) y chileno (cl₁). Se hace notar que solo el *dataset* en español (es) y el nuestro (cl₂) logran resultados consistentemente superiores al *benchmark*. Evaluando los resultados mostrados por las tablas 29,30,31, se detecta que, en promedio, el mejor desempeño para nuestros datos ocurre en EMNLP20 con un promedio de 0,73, el cual es 0,01 menor que el logrado por “XLMT 2021” [55]. Al ver estos resultados es importante recordar que se esperaba que la transferencia a *Hatecheck* fuera menor, debido a la diferencia de los dialectos utilizados en su elaboración. Sin embargo, queda en evidencia que la utilización de modelos enfocados en transferencias multilingües logra generar la abstracción necesaria para obtener buenos resultados en este *dataset* de evaluación. Resulta importante notar la baja de rendimiento en general para cl₂ al combinarlo con cl₁. Una posible explicación de esto puede ser tal y como fue mencionado con anterioridad, esto se deba a la baja cantidad de positivos que posee cl₁, al igual que la diferencia de criterios para recolectar Discursos de Odio utilizado por los autores Arango Monnar *et al.*, que estuvo basada en el uso de *keywords*.

Functionality	Best es	Best en	Best cl ₁	Best cl ₂	Best cl ₁ +cl ₂	XLMT 2021
F1 derog neg emote h	0.79	0.24	0.11	0.69	0.48	0.81
F2 derog neg attrib h	0.76	0.34	0.45	0.89	0.60	0.79
F3 derog dehum h	0.81	0.49	0.39	0.83	0.67	0.84
F4 derog impl h	0.80	0.41	0.01	0.62	0.41	0.64
F5 threat dir h	0.93	0.56	0.11	0.71	0.45	0.89
F6 threat norm h	0.97	0.87	0.13	0.92	0.52	0.91
F7 slur h	0.90	0.42	0.06	0.53	0.39	0.43
F8 profanity h	0.73	0.49	0.20	0.89	0.56	0.76
F9 profanity nh	0.97	0.98	1.00	1.00	1.00	0.99
F10 ref subs clause h	0.63	0.34	0.33	0.81	0.63	0.84
F11 ref subs sent h	0.70	0.32	0.41	0.89	0.63	0.79
F12 negate pos h	0.57	0.31	0.06	0.61	0.41	0.68
F13 negate neg nh	0.64	0.96	1.00	0.98	0.95	0.36
F14 phrase question h	0.54	0.34	0.22	0.77	0.47	0.73
F15 phrase opinion h	0.61	0.41	0.21	0.83	0.52	0.79
F16 ident neutral nh	0.75	1.00	1.00	0.94	0.94	0.69
F17 ident pos nh	0.74	0.99	1.00	0.99	0.97	0.71
F18 counter quote nh	0.86	0.91	1.00	0.96	0.98	0.41
F19 counter ref nh	0.74	0.96	1.00	1.00	0.99	0.65
F20 target obj nh	0.91	1.00	1.00	1.00	1.00	0.86
F21 target indiv nh	0.88	1.00	1.00	1.00	1.00	0.92
F22 target group nh	0.78	1.00	1.00	1.00	1.00	0.71
F23 spell char swap h	0.72	0.29	0.23	0.74	0.44	0.84
F24 spell char del h	0.75	0.31	0.14	0.71	0.36	0.69
F25 spell space del h	0.77	0.42	0.21	0.75	0.44	0.77
F26 spell space add h	0.59	0.21	0.09	0.49	0.29	0.71
F27 spell leet h	0.71	0.36	0.15	0.69	0.33	0.81
AVG	0.76	0.59	0.46	0.82	0.64	0.74
STD	0.12	0.30	0.40	0.16	0.26	0.15

Tabla 35: Resultados Consolidados de Evaluación en *Hatecheck*. Considerando el mejor algoritmo multi-lingua para cada corpus. Los datos resaltados en verde corresponden al valor más alto por fila que supera el *benchmark* (XLMT 2021), mientras que los resaltados en rosado a los que empeoran el *benchmark*.

Fuente: Elaboración propia

Observando los resultados expuestos en la tabla 35 y contrastando los recursos cl₁ y cl₂. Se observa que tomando el promedio de desempeño en las funcionalidades nuestro recurso presenta una mejora del 36 % con respecto a cl₁ dejando en evidencia la mejora funcional que presenta el nuevo corpus por sobre éste último.

Reduciendo los resultados por funcionalidad a sus respectivas categorías (ver tabla 6), se observa que nuestro recurso obtiene resultados superiores al *benchmark* para las siguientes categorías relevantes (mencionadas en la sección 4.3.3): *Profanity usage*, *Negation*, *Phrasing*, *Counter Speech*, *Abuse against non-protected targets*. Estos resultados son de vital

Target Community	Best es	Best en	Best cl ₁	Best cl ₂	Best cl ₁ +cl ₂
black people	0.86	0.70	0.33	0.71	0.57
gay people	0.52	0.42	0.36	0.82	0.80
trans people	0.60	0.42	0.29	0.79	0.79
indigenous people	0.78	0.38	0.51	0.84	0.82
jews	0.83	0.52	0.40	0.88	0.42
disabled people	0.65	0.33	0.23	0.62	0.47
women	0.85	0.89	0.32	0.76	0.42

Tabla 36: Resultados Consolidados de Evaluación en *Hatecheck*. Considerando el mejor algoritmo multi-lingua para cada corpus. Mejores y peores resultados por fila resaltados en verde y rosado respectivamente. En los casos donde el peor o mejor resultado esta repetido entre *datasets* éste no es resaltado.

Fuente: Elaboración propia

Modelo	Datos	Accuracy	Precision	Recall	F1-Score	AUC
EMNLP20	es	0.675033	0.845628	0.657675	0.739902	0.686879
	en	0.535915	0.909341	0.37728	0.533298	0.644165
	cl ₁	0.297463	1	0.00038	0.00076	0.50019
	cl ₂	0.710814	0.830277	0.739742	0.782399	0.691075
	cl ₁ + cl ₂	0.550067	0.822768	0.458587	0.588924	0.612492
ACL19	es	0.532443	0.682856	0.625	0.652648	0.469283
	en	0.30761	0.672566	0.028875	0.055373	0.497816
	cl ₁	0.296128	0.166667	0.00038	0.000758	0.497944
	cl ₂	0.317223	0.809917	0.037234	0.071195	0.508285
	cl ₁ + cl ₂	0.306275	0.693182	0.023176	0.044853	0.499459
EMCL20	es	0.553271	0.816502	0.469985	0.596576	0.610105
	en	0.452069	0.807203	0.289514	0.426174	0.562996
	cl ₁	0.409079	0.84975	0.193389	0.315073	0.556263
	cl ₂	0.566355	0.826848	0.484422	0.610925	0.622265
	cl ₁ + cl ₂	0.522296	0.828016	0.404255	0.543273	0.602846

Tabla 37: Resultados Métricas de Evaluación en *Hatecheck* por conjunto de datos. Mejores y peores resultados por columna resaltados en verde y rosado respectivamente.

Fuente: Elaboración propia

importancia, pues avalan que el recurso posee las capacidades funcionales que se pretendía entregarle al generar la recolección manual, poniendo en evidencia que éste no se encuentra sesgado hacia el uso de profanidades en sus clases de odio, ya que es capaz de detectar correctamente frases con odio negado y viceversa, entre otras cualidades que son evaluadas por *Hatecheck*. Esto demuestra el valor que posee este recurso en la detección de formas de discursos de Odio no triviales.

En la tabla 36 se pueden observar los mejores resultados para la evaluación de sesgos por comunidad, adoptando las métricas propuestas por Röttger *et al.* y evaluando vía F1. Toman-

do como base los resultados expuestos por la tabla 36, es posible evidenciar cómo cl_2 y “es” son los conjuntos de datos con mejores resultados, capaces de reconocer bastante bien todas las comunidades, sin sesgos particulares hacia una de ellas. Viéndose esto reflejado en que ambos conjuntos promedian sobre 70 % al calcular la media simple entre las comunidades. Estos resultados se condicen con los expuestos en la sección 4.3.2: los conjuntos cl_2 y “es” obtienen los mejores resultados para 2 de las 3 métricas de sesgos.

Utilizando la tabla 36, una vez más para comparar el desempeño de cl_1 y cl_2 . Se observa que nuestro recurso presente mejoras evidentes para todas las comunidades identificadas logrando mejoras del: 38 % en *black people*, 46 % en *gay people*, 50 % en *trans people*, 33 % en *indigenous people*, 48 % en *jews*, 39 % en *disabled people* y 44 % en *women*. Evidenciando las mejoras que presenta cl_2 por sobre cl_1 .

CAPÍTULO 5

CONCLUSIONES

En este capítulo se presentan tanto las conclusiones del trabajo realizado, como también el trabajo futuro. Las conclusiones se encuentran organizadas en función de los objetivos propuestos.

5.1. Definición de Herramientas, Metodología y Criterios de Recolección

Con respecto al primer y segundo objetivo específico, se realizó un estudio del estado del arte en la materia, mediante el cual se determinó una baja consistencia tanto en herramientas de muestreo como en metodologías para realizarlo. No obstante, se descubrió que la tendencia en el área se encontraba orientada a la utilización de la API de *Twitter* para el muestreo y diccionarios de palabras clave, razón por la cual se implementó inicialmente esta a la solución propuesta.

Al implementar esta metodología, se descubrió que la estrategia de recolección automática generaba pocos casos positivos, problemática determinada por el vocabulario informal del chileno utilizado en *Twitter*. Por esto, se optó por establecer una estrategia propia de recolección híbrida inspirada en Basile *et al.*. Esta estrategia consistió en la búsqueda automática y manual para enriquecer los datos obtenidos, añadiendo criterios de comunidades, palabras claves, personajes públicos y eventos de interés nacional. A modo de resumen, se logró confeccionar una metodología de muestreo en la que se detallan claramente los tiempos de búsqueda, comunidades objetivos, personajes públicos y eventos de interés nacional (semillas de búsqueda). Mediante la aplicación de esta metodología manual se lograron recolectar 2288 *tweets*, en conjunto con la metodología automática recolectando 2284 *tweets* relevantes con sus respectivos *threads* o contextos.

5.2. Metodologías y Criterios de anotación

En relación con el tercer objetivo específico, el actual estado del arte generó problemas debido a la falta de consenso en cuanto a qué constituye Discursos de Odio. Por esta razón se determinó acudir a entidades internacionales que se encuentran abordando el problema activamente y se optó por utilizar sus definiciones del fenómeno, para así no caer en la disyuntiva de generar otra nueva definición en la literatura. Los criterios de anotado pueden ser vistos en los lineamientos entregados a los anotadores presentes en el apéndice 5.6. Esta definición de criterios y el uso de ellos en la confección de los lineamientos ligados a la metodología de anotado resultaron fundamentales para el proceso, especialmente para ayudar a distinguir casos complejos y para no inducir sesgos de los investigadores en el

anotado, generando reproducibilidad y robustez en el conjunto de datos anotado.

Es importante destacar qué este nuevo recurso entregado a la comunidad científica es el primer *dataset* en dialecto chileno enfocado en identificar explícitamente la comunidad objetivo de los potenciales casos de odio asociados a *tweets*. Al igual que dejar disponible el contexto asociado a los *tweets* clasificados.

5.3. Resultados de Anotación

Siguiendo al cuarto objetivo y con el fin de evaluar la distribución de los datos anotados, se generaron tablas y gráficos (ver figuras 17, 18, 19, 20) que indicaron una distribución del 45,5 % de datos de Odio marcados como positivos por los anotadores, utilizando como criterio para los positivos que ganara la categoría por mayoría simple, es decir votos ≥ 2 . Así, se anotaron 4572 *tweets*, donde las cantidades distribuidas por categoría pueden ser observadas en la tabla 9. En relación con estos últimos resultados, se destaca el hecho de obtener un alto balance de clases, en especial para la categoría de Odio, y un balance mediano para las comunidades. Como métrica del nivel de consenso por categoría se utilizó el alfa de Krippendorff, métrica de consenso ampliamente utilizada en la lingüística computacional a la hora de clasificar datos y medir niveles de consenso para categorías. Se obtuvieron así los resultados expuestos en la tabla 10, donde los grupos muy altos niveles de consenso para la clasificación de Odio y niveles de acuerdo menos decisivos pero consistentes con los umbrales utilizados en la literatura [31], para las distintas comunidades. Finalmente, estos resultados con los votos por categoría disponibles, además de los contextos o *threads* asociados a cada *tweet*, fueron publicados en *ieee-dataport* conforme a los lineamientos dictaminados por *Twitter* bajo los principios de *FAIR* (*Findability, Accessibility, Interoperability, and Reusability*) para el manejo de datos científicos, y es posible encontrarlos para su uso en *hate-speech-chilean-twitter*. La decisión de dejar los votos por categoría en los datos fue tomada con el fin de poder entregar mayor flexibilidad a éstos y, a su vez, permitir a otros investigadores mayor libertad en la manipulación y nuevas formulaciones de este nuevo recurso.

5.4. Modelos *Baselines* y *Benchmarks*

En lo que compete al quinto objetivo, se utilizaron los modelos descritos en la sección 2.3.3, donde y tal como se describe en el diseño experimental (ver sección 3.3.1) se formularon 3 diseños experimentales con los objetivos determinar las capacidades de transferencia a nuevos dominios de nuestros datos, las mejoras funcionales logradas por el recurso y finalmente las capacidades de mitigación de sesgos del recurso con respecto a recursos similares. En los experimentos de transferencia, se evidencia que nuestro *dataset* muestra mejoras significativas transfiriendo tanto al inglés como al español castellano, demostrando que la transferencia entre idiomas y dialectos mediante modelos multi-lingua puede resultar exitosa, sugiriendo, por lo tanto, que la transferibilidad podría estar ligada a la riqueza que posee un

dataset en términos de formas de expresión de odio y diversidad de comunidades involucradas. Estos resultados confirman que la transferibilidad entre dialectos es mucho más alta que entre idiomas diferentes, mostrando una clara preferencia a la utilización de recursos entrenados sobre un mismo dialecto para la realización de transferencia de conocimiento. De lo anterior, se observa que la transferencia desde nuestro *dataset* a otro *dataset* en dialecto chileno resultó mucho más satisfactoria que la transferencia desde ese recurso a nuestro *dataset*.

Tres situaciones que pueden explicar este resultado son las siguientes. Primero, los autores reportan entrenar los modelos para la detección de lenguaje ofensivo, sin aclarar si con esto se refieren a la detección de Discursos de Odio que se interseccionan con el conjunto del lenguaje ofensivo, o la detección de lenguaje ofensivo como conjunto que puede contener a los Discursos de Odio. Si bien la diferencia es sutil, las implicaciones en los modelos son severas, puesto que en el segundo caso se hablaría de una falla metodológica, ya que la comunidad distingue claramente entre Discursos de Odio y lenguaje ofensivo como categorías relacionadas pero diferentes.

Otra posible problemática que explica la baja transferibilidad de “cl₁” hacia otros conjuntos de datos, pueda ser causada por, como se planteó en la sección 4.3.1, el alto desbalance de clases para la categoría de *Hate Speech* de Arango Monnar *et al.* donde reportan solo un $\approx 6\%$ de positivos para esta categoría, siendo esta la causa principal de la incapacidad del recurso para generalizar correctamente. Finalmente, el uso de un solo anotador por dato le quita robustez a las etiquetas. Si bien la utilización de una etapa de entrenamiento nos parece fundamental al momento de introducir a los anotadores al problema y asegurar un buen desempeño en los datos finales, al anotar datos por un solo anotador se corre el riesgo de incurrir en fallas y sesgos personales en datos que pretende generalizar. Por otro lado, esta práctica reduce las capacidades de análisis de errores o evaluar la real significancia estadística de las anotaciones.

En relación con la evaluación de las características funcionales del recurso en el *dataset* de Röttger *et al.*, se obtuvieron resultados que fácilmente superaron al estado del arte en 16 de las 27 categorías, entre las cuales se encuentran las 4 categorías de especial relevancia para este recurso: *Profanity usage*, *Negation*, *Phrasing*, *Counter Speech*, *Abuse against non-protected targets*. Esto es de suma importancia al tomar en cuenta que el *dataset* de evaluación no posee un único dialecto del español y que ambos dialectos utilizados son español de España y Argentina. Tal y como se mencionó en la sección 4.3.3, nuestro *dataset* y el de Basile *et al.* logran los mejores resultados, logrando superar al *benchmark* en repetidas ocasiones. Esto resulta indicativo de las capacidades funcionales del nuevo recurso, pues es capaz de abstraer el fenómeno de discursos de odio, no solo en el dialecto chileno, sino en otros también. Lamentablemente, los resultados de Arango Monnar *et al.* reflejan una situación de *hit or miss* donde los modelos entrenados sobre el recurso obtienen excelentes resultados en ciertas funcionalidades y paupérrimos en otras, lo cual nuevamente parece ser indicativo de las deficiencias intrínsecas que posee este recurso.

Se determinó que al emular lo realizado por Röttger *et al.* para el análisis de sesgos hacia

comunidades, nuestro *dataset* y el de Basile *et al.* obtienen nuevamente los mejores resultados, estas conclusiones muestran seguir la tendencia expuesta en la sección 4.3.2 donde se utilizaron métricas especializadas en la medición de sesgos, que lograron evidenciar el mismo comportamiento.

Al utilizar la unión de nuestro *dataset* y el de Arango Monnar *et al.*, no se evidencian mejoras de desempeño en ninguno de los experimentos realizados, por lo cual si se quiere persistir en esta línea, se deberá recurrir a métodos más sofisticados para unirlos, con tal de generar una mejora importante en el rendimiento.

Finalmente, se concluye que se logró el objetivo exitosamente, pudiendo evaluar 3 modelos del estado del arte sobre nuestros datos, logrando generar resultados y evaluaciones, capaces de evidenciar claramente las capacidades de transferencias de dominio, capacidades funcionales y sesgos asociados a este nuevo recurso. Ayudando así a la creación de herramientas que nos permitan luchar contra la discriminación y el odio en internet.

5.5. Publicación y Divulgación

En lo que refiere al sexto objetivo, actualmente se está redactando en paralelo una publicación con los resultados obtenidos en este trabajo. La idea tras esta publicación es lograr publicar nuestros resultados tanto en la recopilación de datos como en los modelos que lo gran generalizar mejor el fenómeno de Discursos de Odio para el dialecto chileno.

5.6. Trabajo Futuro

Como trabajo futuro se proponen algunas de las posibles siguientes líneas:

- Experimentar con más modelos multilingües y generar un modelo original capaz de obtener resultados competitivos en el área.
- Experimentar con transferencias a otras lenguas y verificar si existen tendencias claras para el fenómeno de *Hate Speech* dentro de lenguajes pertenecientes a las mismas familias y/o para distintos dialectos de un mismo lenguaje.
- Emular lo presentado por Mathew *et al.* con tal de mejorar la explicabilidad de los datos presentados o tomar nuestras metodologías de inspiración para generar nuevos datos utilizando lo anterior para mejorar la explicabilidad.
- Profundización del recurso mediante el uso de humanos para anotar trozos del texto más correlacionadas con odio o para la validación de técnicas de explicabilidad en NLP sobre éste.

- Expansión de este recurso para su utilización en el anotado mediante métodos de aprendizaje semisupervisados.

ANEXOS

ANEXO A: Palabras claves, cuentas y *hashtags*

Comunidad LGBTQ+	Pueblos Originarios	Mujeres	Comunidades Migrantes
hueco culiao	huele a indígena	te la meto	extranjeros
weco culiao	mapuche culiao	choro	migrante
tortillera	mapuche	noesviolacion	inmigrante
camionera	aymara	no es violacion	negro culiao
gey	changos	todas putas	negra culia
gay	chonos	eres una imbecil	veneco culiao
mariposon	diaguita	sirvienta	cholano culiao
mariposona	quechua	tragatela	haitiano culiao
trans	indigenismo	acoso	peruano culiao
transgenero	indigenista	histerica	chile para los chilenos
travesti	loncon	violar	colombiano culiao
transvestidos	llaitul	violacion	venezolano
trasvestido	catrilanca	feminazi	peruano
trava		feminasi	cholo
amariconadas		feminazis estupidas	haitiano
amariconado		feminazi estúpida	negro
amariconao		feminasis estupidas	
maraca		feminasi estúpida	
maraco		perra	
maricon		perra estúpida	
maricona		perra loca	
lesbiana		perra tonta	
lela		puta	
menstruante		puta estúpida	
homo		puta tonta	
homosexual		puta loca	
homofobia		zorra	
lesbofobia		maraca	
transfobia		maraca culia	
la sodomizacion		maricona	
creerse mujer		vuelve a la cocina	
hombre menstruante		andate a lavar	
solo hay hombres y mujeres		andate a la cocina	
no es homofobia		andate a lavar	
maricon menstruante		anda laar	
menstruante culiao		devuelvete a la cocina	
menstruante		las mujeres son inferiores	
		frescaderaja	
		fresca de raja	
	Página 88 de 97	to q debe hacer una mujer	
		lo que debe hacer una mujer	

Tabla 38: Pabras claves por comunidad.

Cuentas	Hashtags
@PRChile	#Heterofobia
@RojoEdwards	#niunamenos
@joseantoniokast	#ToleranciaCero
@Jou_Kaiser	#IzkiSiches
@camila_vallejo	#CamilaVallejos
@ProfMarcoAvila	
@AleBenado	
@emischneiderv	
@MAM_Chile	
@inddhh	
@ciper	
@CNNChile	
@latercera	
@Emol	
@Movilh	
@corphumanas	
@Mamanilsabella	
@HortenciaXArica	

Tabla 39: Cuentas y *Hashtags* relevantes.
Fuente: Elaboración Propia.

ANEXO B: Lineamientos

Lineamientos

Visión General

Bienvenido a la tarea de anotación. En esta tarea presentaremos un *tweet* en conjunto con el hilo de conversación asociado y te pediremos clasificarlo indicando a que clase pertenece dado su contexto, estos datos nos permitirán generar un *dataset* de libre acceso el cual será liberado a la comunidad científica, con el cual construiremos un modelo de *baseline* el cual permitirá generar comparativas ahondando en el desarrollo de esta área nivel nacional.

Conceptos

Queremos que clasifiques cada *tweet* como **odio** o **no odio**, en el caso de ser de **odio** será obligatorio indicar la comunidad objetivo al cual esta dirigido el *tweet* pudiendo ser estas

comunidades migrantes, comunidad LGBTQ+, mujeres y pueblos originarios, en caso contrario y ser un *tweet* de *no odio* si se encuentra alguna de las comunidades mencionadas será necesario indicarla.

El concepto de presencia de **odio** con el que se trabajará se encuentra ligado al concepto de discurso de odio definido por la ONU como: “Cualquier forma de comunicación de palabra, por escrito o a través del comportamiento, que sea un ataque o utilice lenguaje peyorativo o discriminatorio en relación con una persona o un grupo sobre la base de quiénes son o, en otras palabras, en razón de su religión, origen étnico, nacionalidad, raza, color, ascendencia, género u otro factor de identidad” Si se desea ahondar en el tema recomendamos revisar el plan de acción de la ONU para más definiciones respecto a que es el discurso de odio.

En esta tarea utilizaremos la definición de discurso de odio acotada como:

- Discurso de Odio: Texto o mensaje que constituya un ataque o promueva la discriminación de una persona o grupo sólo sobre la base de pertenencia a uno de los siguientes grupos: **comunidades migrantes, comunidad LGBTQ+, mujeres y pueblos originarios**.

La dificultad principal de esta tarea yace principalmente en dos casos el primero es cuando es discurso de odio pero no utiliza lenguaje peyorativo de por sí, si no que la frase completa busca mostrar al grupo mencionado como menos capaces en cualquier ámbito físico, mental, monetario, amoroso, etc. o menos merecedores del acceso a derechos. El segundo caso corresponde al uso de lenguaje peyorativo pero que en su contexto no es considerado discriminatorio es decir utilizar estas palabras al hablar de un amigo bromeando o de manera sarcástica, este caso es de vital importancia puesto que en el contexto social y comunicativo del dialecto chileno este método comunicativo es altamente frecuente en situaciones informales o al relacionarse con pares.

Para esta tarea también introduciremos el concepto de **contexto**, contexto se encuentra definido por la RAE como “Entorno lingüístico de una palabra, frase o fragmento considerados, del que depende muchas veces su sentido”, para esta tarea es de vital importancia que al clasificar el *tweet* entregado usted utilice el contexto para encontrar el sentido real de la frase, es decir pueden existir frases con sentido aparentemente inocuos por sí solas como “si así debería ser” pero que revisando el contexto su significado puede cambiar notoriamente por ejemplo el *tweet* anterior puede darse como una respuesta al siguiente *tweet* “Solo la gente blanca debería controlar el país”, donde si bien el *tweet* anterior no utiliza lenguaje peyorativo contra un grupo específico en este caso busca disminuir los derechos de todos aquellos que no pertenezcan al grupo en cuestión y por lo tanto en su contexto el *tweet* si contribuye al discurso de odio.

En general estos casos de odio utilizan argumentos falaces y esto nos permite facilitar su detección en cierto sentido a continuación se muestran algunos tipos usuales y en los siguientes links pueden ver falacias más a fondo clasificaciones de falacias, tipos.

Clases

Para cada *tweet*, elija primero alguna de las siguientes clases:

- **Odio:** Si un *tweet* en el contexto del hilo en el que se encuentra incita al odio de acuerdo al concepto indicado con anterioridad. Por ejemplo:
 - “lo llama matrimonio, bien por los que lucharon por eso pero para mí va a seguir siendo un concepto que une a un hombre y a una mujer. Simple.” (este *tweet* implica que el matrimonio entre parejas homosexuales no es válido, implica que los miembros de esta comunidad carecen de acceso a los mismo recursos y derechos que otras personas que no pertenecen a esta comunidad)
 - “Es que no se como la vamos a nombrar, porque no es persona menstruante ni persona eyaculante, le podremos decir maricon a secas?”
Utiliza lenguaje peyorativo para referirse y burlarse de toda una comunidad, no siendo un caso de bromas entre amigos o auto ironía.
- **No Odio:** Si un *tweet* no presenta odio. Por ejemplo:
 - “cansona triplehijueputa, hija de las tres miles putas, malparida, perra, puta, cansona, me tenés mamada, sapaaaaa
de: mi
para: mi ansiedad”
En este *tweet* se utiliza lenguaje peyorativo, sin embargo lo utilizar para burlarse de si misma por lo cual no corresponde a lenguaje de odio.

Para cada *tweet*, una vez clasificado en alguna de las clases anteriores determine Si el *tweet* hace referencia a alguna de las siguientes comunidades (elección múltiple):

- **Comunidad LGBTQ+** Por ejemplo:
 - “Yo soy heterosexual y quiero que mis hijos sean heterosexuales algún problema?”
Odio, el argumentar que la identidad sexual de los hijos es definida por los padres es incorrecto y busca promover que aquellos con orientaciones distintas son menos válidos que quienes poseen orientaciones sexuales convencionales.
 - “Profe, mi mamá me dijo que usted es gay.”
No odio, no utiliza lenguaje peyorativo ni busca discriminar si no que mas bien parece estar realizando un cuestionamiento a alguien en base a a un comentario generado en torno al interlocutor mencionado.
- **Mujeres** Por ejemplo:

- “Existe vida tras ser feminazi. Hay gente que evoluciona cuando empieza a leer y pensar. Un aplauso.”

Odio, busca perpetuar la visión del feminismo como una postura ideológica perteneciente a aquellos de bajo nivel intelectual con lo cual cae en la discriminación.

- “Re sola estoy igual q vos gorda puta”

No Odio, si bien esta insultado a alguien el emisor del mensaje deja en claro que ambas se encuentran a un nivel de pares en los que respecta a su soledad.

■ **Pueblos Originarios** Por ejemplo:

- “Wnes imbeciles diciendo Que Emoción!!!! Mientras la mayoría de los Chilenos pagamos Contribuciones, no así los Mapuches, les siguen dando más beneficios, mientras el 90 % se los debemos financiar. Aweonaos.”

Odio, se utilizan argumentos falaces para apoyar la discriminación hacia pueblos originarios, señalando que ellos no pagan contribuciones y viven de dineros del estado.

■ **Comunidades Migrantes** Por ejemplo:

- “Todos los venezolanos son flojos, vienen a puro robar”

Odio, utiliza argumentos falaces como la generalización para señalar que toda la comunidad de migrantes venezolanos son flojos y criminales.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Alatawi, H. S., Alhothali, A. M., y Moria, K. M. (2021). Detecting white supremacist hate speech using domain specific word embedding with deep learning and bert. *IEEE Access*, 9:106363–106374.
- [2] Albadi, N., Kurdi, M., y Mishra, S. (2018). Are they our brothers? analysis and detection of religious hate speech in the arabic twittersphere. En *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 69–76. IEEE.
- [3] Alfina, I., Mulia, R., Fanany, M. I., y Ekanata, Y. (2017). Hate speech detection in the indonesian language: A dataset and preliminary study. En *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pp. 233–238. IEEE.
- [4] Alotaibi, A. y Hasanat, M. H. A. (2020). Racism detection in twitter using deep learning and text mining techniques for the arabic language. En *2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, pp. 161–164. IEEE.
- [5] Aluru, S. S., Mathew, B., Saha, P., y Mukherjee, A. (2020). Deep learning models for multilingual hate speech detection. *CoRR*, abs/2004.06465.
- [6] Arango Monnar, A., Perez, J., Poblete, B., Saldaña, M., y Proust, V. (2022). Resources for multilingual hate speech detection. En *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, pp. 122–130, Seattle, Washington (Hybrid). Association for Computational Linguistics.
- [7] Artetxe, M. y Schwenk, H. (2018). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *CoRR*, abs/1812.10464.
- [8] Artstein, R. y Poesio, M. (2008). Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4):555–596.
- [9] Bai, S., Kolter, J. Z., y Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.
- [10] Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F. M., Rosso, P., y Sanguinetti, M. (2019). SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. En *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.
- [11] Borkan, D., Dixon, L., Sorensen, J., Thain, N., y Vasserman, L. (2019). Nuanced metrics for measuring unintended bias with real data for text classification. *CoRR*, abs/1903.04561.
- [12] Cohen, J. (1968). Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

- [13] Collobert, R. y Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. En *Proceedings of the 25th international conference on Machine learning*, pp. 160–167.
- [14] Conneau, A., Lample, G., Ranzato, M., Denoyer, L., y Jégou, H. (2017). Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.
- [15] Convención (2021). Reglamento de Ética convención constitucional.
- [16] Davidson, T., Warmusley, D., Macy, M., y Weber, I. (2017). Automated hate speech detection and the problem of offensive language. En *Proceedings of the International AAAI Conference on Web and Social Media*, volumen 11.
- [17] de Gibert, O., Perez, N., García-Pablos, A., y Cuadros, M. (2018). Hate Speech Dataset from a White Supremacy Forum. En *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pp. 11–20, Brussels, Belgium. Association for Computational Linguistics.
- [18] Goldberg, Y. (2016). A primer on neural network models for natural language processing. *ArXiv*, abs/1510.00726.
- [19] Goodfellow, I., Bengio, Y., y Courville, A. (2016). *Deep learning*. MIT press.
- [20] Grefenstette, E., Blunsom, P., De Freitas, N., y Hermann, K. M. (2014). A deep architecture for semantic parsing. *arXiv preprint arXiv:1404.7296*.
- [21] Guterres, A. (2019). United nations strategy and plan of action on hate speech.
- [22] Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen netzen. diploma thesis. *Josef Hochreiter Institute for Computer Science. Technical University, Munich*.
- [23] Hochreiter, S. y Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- [24] Ibrohim, M. O. y Budi, I. (2019). Multi-label hate speech and abusive language detection in indonesian twitter. En *Proceedings of the Third Workshop on Abusive Language Online*, pp. 46–57.
- [25] James, G., Witten, D., Hastie, T., y Tibshirani, R. (2013). *An introduction to statistical learning*, volumen 112. Springer.
- [26] Jurafsky, D. (2000). *Speech & language processing*. Pearson Education India.
- [27] Kalchbrenner, N., Grefenstette, E., y Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.
- [28] Kehat, G. y Pustejovsky, J. (2016). Annotation methodologies for vision and language dataset creation. *arXiv preprint arXiv:1607.02769*.

- [29] Khan, M. M., Shahzad, K., y Malik, M. K. (2021). Hate speech detection in roman urdu. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 20(1):1–19.
- [30] Kiela, D., Firooz, H., Mohan, A., Goswami, V., Singh, A., Ringshia, P., y Testuggine, D. (2020). The hateful memes challenge: Detecting hate speech in multimodal memes. *CoRR*, abs/2005.04790.
- [31] Krippendorff, K. (2004). Measuring the reliability of qualitative text analysis data. *Quality and quantity*, 38:787–800.
- [32] Lawler, R. (2022). Meta reportedly plans to shut down crowdtangle, its tool that tracks popular social media posts.
- [33] Lipton, Z. C., Berkowitz, J., y Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.
- [34] Luu, S. T., Nguyen, H. P., Van Nguyen, K., y Nguyen, N. L.-T. (2020). Comparison between traditional machine learning models and neural network models for vietnamese hate speech detection. En *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, pp. 1–6. IEEE.
- [35] Marpaung, A., Rismala, R., y Nurrahmi, H. (2021). Hate speech detection in indonesian twitter texts using bidirectional gated recurrent unit. En *2021 13th International Conference on Knowledge and Smart Technology (KST)*, pp. 186–190. IEEE.
- [36] Mathew, B., Saha, P., Yimam, S. M., Biemann, C., Goyal, P., y Mukherjee, A. (2021). Hatexplain: A benchmark dataset for explainable hate speech detection. En *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 14867–14875.
- [37] Mathur, P., Sawhney, R., Ayyar, M., y Shah, R. (2018). Did you offend me? classification of offensive tweets in hinglish language. En *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pp. 138–148.
- [38] McHugh, M. L. (2012). Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.
- [39] McIntosh, P. y Plous, S. (2003). *Understanding prejudice and discrimination*. McGraw-Hill.
- [40] Milmo, D. (2022). Rohingya sue facebook for £150bn over myanmar genocide.
- [41] Mitchel, T. (1997). *Machine Learning, McGraw-Hill Education (ISE Editions)*. McGraw-Hill.
- [42] Mookdarsanit, L. y Mookdarsanit, P. (2021). Combating the hate speech in thai textual memes. *Indonesian Journal of Electrical Engineering and Computer Science*, 21(3):1493–1502.

- [43] Mozafari, M., Farahbakhsh, R., y Crespi, N. (2020). Hate speech detection and racial bias mitigation in social media based on bert model. *PloS one*, 15(8):e0237861.
- [44] Núñez, C. y France-Pressé, A. (2021). ¿no más migrantes!": Manifestantes queman carpas de venezolanos en iquique.
- [45] Oriola, O. y Kotzé, E. (2020). Evaluating machine learning techniques for detecting offensive and hate speech in south african tweets. *IEEE Access*, 8:21496–21509.
- [46] Pamungkas, E. W., Basile, V., y Patti, V. (2021). A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection. *Information Processing & Management*, 58(4):102544.
- [47] Perry, T. (2021). LightTag: Text annotation platform. En *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 20–27, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- [48] Pirie, M. (2015). *How to win every argument: The use and abuse of logic*. Bloomsbury Publishing.
- [49] Poletto, F., Stranisci, M., Sanguinetti, M., Patti, V., y Bosco, C. (2017). Hate speech annotation: Analysis of an italian twitter corpus. En *4th Italian Conference on Computational Linguistics, CLiC-it 2017*, volumen 2006, pp. 1–6. CEUR-WS.
- [50] Ranasinghe, T. y Zampieri, M. (2020). Multilingual offensive language identification with cross-lingual embeddings. *CoRR*, abs/2010.05324.
- [51] Reyes, F. (2021). Diputado electo johannes kaiser cierra su twitter tras revelarse historial misógino y antifeminista.
- [52] Romim, N., Ahmed, M., Talukder, H., e Islam, M. S. (2021). Hate speech detection in the bengali language: A dataset and its baseline evaluation. En *Proceedings of International Joint Conference on Advances in Computational Intelligence*, pp. 457–468. Springer.
- [53] Rumelhart, D. E., Hinton, G. E., y Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.
- [54] Russell, S. J. (2010). *Artificial intelligence a modern approach*. Pearson Education, Inc.
- [55] Röttger, P., Seelawi, H., Nozza, D., Talat, Z., y Vidgen, B. (2022). Multilingual hatecheck: Functional tests for multilingual hate speech detection models.
- [56] Sabat, B. O., Canton-Ferrer, C., y Giró-i-Nieto, X. (2019). Hate speech in pixels: Detection of offensive memes towards automatic moderation. *CoRR*, abs/1910.02334.
- [57] Samarasinghe, S., Meegama, R., y Punchimudiyanse, M. (2020). Machine learning approach for the detection of hate speech in sinhala unicode text. En *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, pp. 65–70. IEEE.

- [58] Sazany, E. y Budi, I. (2018). Deep learning-based implementation of hate speech identification on texts in Indonesian: Preliminary study. En *2018 International Conference on Applied Information Technology and Innovation (ICAITI)*, pp. 114–117. IEEE.
- [59] Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.
- [60] Sepúlveda, J. (2021). Denuncian posible agresión homofóbica en puente alto: víctima está en coma inducido.
- [61] Sharma, C., Bhageria, D., Scott, W., PYKL, S., Das, A., Chakraborty, T., Pulabaigari, V., y Gamback, B. (2020). Semeval-2020 task 8: Memotion analysis—the visuo-lingual metaphor! *arXiv preprint arXiv:2008.03781*.
- [62] Shen, Y., He, X., Gao, J., Deng, L., y Mesnil, G. (2014). Learning semantic representations using convolutional neural networks for web search. En *Proceedings of the 23rd international conference on world wide web*, pp. 373–374.
- [63] Simonyan, K. y Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [64] Sin-wai, C. (2014). *Routledge Encyclopedia of Translation Technology*. Taylor & Francis.
- [65] Suryawanshi, S., Chakravarthi, B. R., Arcan, M., y Buitelaar, P. (2020a). Multimodal meme dataset (multioff) for identifying offensive content in image and text. En *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pp. 32–41.
- [66] Suryawanshi, S., Chakravarthi, B. R., Verma, P., Arcan, M., McCrae, J. P., y Buitelaar, P. (2020b). A dataset for troll classification of tamilmemes. En *Proceedings of the WILDRE5–5th Workshop on Indian Language Data: Resources and Evaluation*, pp. 7–13.
- [67] Van Rijsbergen, C. (1979). *Information Retrieval*. Butterworths.
- [68] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., y Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- [69] Waseem, Z. y Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. En *Proceedings of the NAACL student research workshop*, pp. 88–93.
- [70] Yin, W., Kann, K., Yu, M., y Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. *CoRR*, abs/1702.01923.
- [71] Yu, F. y Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.
- [72] Zhang, Y. y Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.