Análisis de Sentimiento en Tweets de Fútbol Argentino

Facultad de Matemática, Astronomía, Física y Computación

Universidad Nacional de Córdoba

Mario Ferreyra

Director: Franco M. Luque

Resumen

En la actualidad la cantidad de datos que se genera en las redes sociales es gigantesca. Aquí es donde los sistemas de Análisis de Sentimiento resultan de gran utilidad, ya que su principal objetivo es identificar opiniones positivas o negativas en los textos de los usuarios sobre algún producto o marca. Para la construcción de sistemas de Análisis de Sentimiento se utilizan conjuntos de datos anotados con polaridad. Sin embargo, los recursos disponibles para el idioma español son limitados, particularmente para el castellano de Argentina a donde prácticamente no existen.

En este trabajo construimos un corpus de tweets en español de Argentina orientado al tópico del Fútbol Argentino. Para ello se recolectó una gran cantidad de tweets, que luego pasó por etapas de filtrado y anotación realizada por voluntarios, aplicando criterios claros y explícitos definidos por nosotros.

Luego, diseñamos e implementamos distintos sistemas de clasificación de sentimiento, usando técnicas estándar de preprocesamiento, recursos lingüísticos y distintas representaciones de los tweet. Realizamos experimentos utilizando para entrenar y evaluar el corpus de nuestra creación, así como también otros corpus en español previamente existentes. Finalmente hicimos un análisis de los modelos y de los resultados de la evaluación.

Palabras claves: Procesamiento del Lenguaje Natural, Análisis de Sentimiento, Twitter, Corpus, Fútbol Argentino, Clasificadores Lineales, Recursos Lingüísticos, Bolsa de Palabras, Word Embeddings, Aumentación de Datos, Clasificadores en Cascada.

Abstract

Currently the amount of data generated on social networks is gigantic. This is where Sentiment Analysis systems are very useful, since their main objective is to identify positive or negative opinions in users' texts about a product or brand. For the construction of Sentiment Analysis systems, datasets annotated with polarity are used. However, the resources available for the Spanish language are limited, particularly for the Castilian of Argentina where they practically do not exist.

In this work we build a corpus of tweets in Spanish from Argentina oriented to the topic of Argentine Soccer. For this, a large number of tweets were collected, which then went through filtering and annotation stages carried out by volunteers, applying clear and explicit criteria defined by us.

Then, we designed and implemented different sentiment classification systems, using standard preprocessing techniques, language resources, and different representations of tweets. We carry out experiments using the corpus of our creation, as well as previously existing Spanish corpora, for training and evaluation. Finally we did an analysis of the models and the evaluation results.

Keywords: Natural Language Processing, Sentiment Analysis, Twitter, Corpus, Fútbol Argentino, Linear Classifiers, Linguistic Resources, Bag of Words, Word Embeddings, Data Augmentation, Cascading Classifiers.

Agradecimientos

En primer lugar mi agradecimientos es hacia mis padres, Mario e Isabel, quienes me apoyaron e incentivaron a estudiar, me facilitaron el no tener que trabajar, durante mis años de estudio en la facultad, haciendo que nunca me falte nada. A mis hermanos Mauro y Marco, que junto a mis padres son pilares fundamentales y participes en lo que hoy soy. Me bancaron cuando volvía cansado y de mal humor.
Este pequeño párrafo no alcanza para agradecer todo lo que hicieron y hacen por mi.

A mi director Franco Luque por aceptarme como tesista y proponerme un tema que me encanta y mantenerme motivado para la realización de este trabajo, también por su paciencia y tiempo para evacuar mis dudas. Sin su supervisión este trabajo no hubiera sido posible.

A mis amigos por el aguante de siempre y la motivación.

A FaMAF por permitirme estudiar lo que me gusta y a los profesores que participaron en toda mi formación académica, por su dedicación y entrega incluso fuera del horario de clases.

No quiero dejar de nombrar a mi Abuela, tias, tio y primos que siempre estuvieron ahí, alentandome y festejando cada uno de mis logros.

Por último, este trabajo va dedicado especialmente a mi abuelo José, que ya no está físicamente, pero vive en mi corazón, el cual nos dejó a todos la mejor herencia.
Tata: Quisiera que hoy te escaparas un ratito del cielo y me dieras un fuerte abrazo, este título es para vos !!!.

Índice

Capítulo 1

Introducción

1.1 Contexto / Motivación

Con las redes sociales, los usuarios tienen hoy en día todo tipo de facilidades para mostrar su opinión sobre cualquier tema que deseen. Tener constancia sobre las opiniones referentes a una marca o producto y medir su impacto es actualmente de vital importancia para todas las empresas, ya que es su imagen lo que está en juego.

A toda la información que se recopila de esta forma se le denomina Minería de Opinión (Opinion Mining) y gracias a ella, las empresas tienen una inmediata respuesta de qué es lo que opinan los internautas sobre sus productos o marcas, y con esto poder obtener ventajas competitivas en diferentes ámbitos.

Twitter, creado en marzo de 2006 en California (Estados Unidos) es un servicio de microblogging que se estima que tiene más de 500 millones de usuarios, generando 65 millones de tweets al día y que maneja más de 800 mil peticiones de búsqueda diarias. Esta red social permite enviar mensajes de texto plano de corta longitud, con un máximo de 280 caracteres (originalmente 140), llamados tuits o tweets.

El principal uso de los tweets es expresar opiniones y puntos de vista de los usuarios. Por ello, estos textos resultan ser de gran interés para realizar un análisis de tendencias, calificación de productos, etc.

Tales tweets han sido usados en distintos trabajos de investigación, como por ejemplo:

- Predicción de Tendencias
- Clasificación de usuarios según edad, sexo, orientación política.
- Clasificación de textos
- Análisis de Sentimiento

1.2 Problema / Desafío

En muchas casos, cuando hablamos de reputación online, aparece el concepto de **Análisis de Sentimiento**. El cual se refiere a los diferentes métodos de lingüística computacional que ayudan a identificar y extraer información subjetiva del contenido existente en el mundo digital como las redes sociales, foros, etc.

Gracias al análisis del sentimiento, podemos ser capaces de extraer un valor tangible y directo, como puede ser determinar si un texto extraído de internet contiene, en su caso más simple, connotaciones positivas o negativas.

La tarea de clasificar automáticamente un texto escrito en lenguaje natural, en un sentimiento positivo o negativo, es a veces tan complicada que incluso es difícil poner de acuerdo a diferentes anotadores humanos sobre qué polaridad asignar a un texto dado. Ya que la interpretación personal de un individuo es diferente a la de los demás y se ve afectada por distintos factores, como por ejemplo, las experiencias propias de cada persona.

En un texto corto donde abundan los errores de ortografía, esta tarea se torna aún más difícil ya que son una desventaja, por lo que se requiere un trabajo previo (como la normalización del texto) para poder realizar un análisis. Como es el caso de los mensajes en redes sociales como Twitter o Facebook.

Mediante el Análisis de Sentimiento, queremos lograr entender cuál es la intención exacta de una frase. Saber si se refiere a una marca, a un producto en concreto o a cualquier otro aspecto.

Gracias al Análisis de Sentimiento, se consigue desarrollar mejores estrategias empresariales, facilitar la gestión de la reputación online y ayudar a la toma de decisiones para llevar a cabo en el plan estratégico de marketing online.

En este trabajo, estudiamos el problema de Análisis de Sentimiento en Twitter, particularmente tweets escritos en español. Para ello trabajamos con varios corpus de distintas procedencias, además de crear uno propio con 1994 tweets de Argentina pertenecientes al tópico futbolístico; para obtener los mismos se siguió una serie de pasos, la recolección de los tweets, el filtrado de los mismos según el tópico y el etiquetado de polaridad, todo esto siguiendo criterios definidos por nosotros.

Luego desarrollamos sistemas con distintos enfoques para atacar el problema, evaluandolos con diferentes métricas que nos dan una noción de la performance de los mismos.

Posteriormente realizamos un análisis de errores obteniendo conclusiones de los sistemas, para finalmente proponer mejoras que se puedan aplicar, con el fin de obtener mejores resultados.

1.3 Esquema de Trabajo

En el **Capítulo 2** introduciremos los conceptos teóricos que utilizaremos más adelante, como son el Procesamiento del Lenguaje Natural, el Machine Learning y los Clasificadores Lineales. También hablaremos de las distintas métricas usadas para evaluar los modelos. Por últimos describiremos algunos trabajos previos enfocados en la resolución de la problemática que planteamos.

En el **Capítulo 3** describiremos los corpus utilizamos para el desarrollo de este trabajo, además de mostrar algunas estadísticas de los mismos. Contaremos el proceso y las circunstancias para la creación de un corpus propio en español hablado en Argentina, particularmente sobre el tópico de fútbol.

En el **Capítulo 4** detallaremos los distintos enfoques de los sistemas de clasificación que se fueron implementando.

En el **Capítulo 5** mostraremos los resultados que se obtuvieron sobre los distintos corpus aplicando los sistemas implementados, además de hacer una inspección de algunos de los sistemas y un análisis de error de los mismos.

En el **Capítulo 6** daremos una conclusión a los resultados de este trabajo y nombraremos algunas posibles mejoras u otros enfoques que se pueden hacer para seguir mejorando los sistemas.

Finalmente en el **Capítulo 7** anexamos la documentación que fue usada para la obtención de nuestro corpus propio, en donde se detallan las criterios que se siguieron en proceso de creación del corpus.

Capítulo 2

Preliminares

En este capítulo describiremos brevemente algunos conceptos importantes que han sido usados para resolver el problema que se plantea en este trabajo.

2.1 Natural Language Processing

El *Natural Language Processing* (**NLP**) o *Procesamiento del Lenguaje Natural* (**PLN**) es un campo de la Ciencias de la Computación, Inteligencia Artificial y Lingüística que estudia las interacciones entre las computadoras y el lenguaje humano.

El área de PLN se ocupa de la formulación e investigación de mecanismos eficaces computacionalmente para la comunicación entre personas y máquinas por medio de lenguajes naturales. Los modelos que se aplican se enfocan no solo a la comprensión del lenguaje de por sí, sino a aspectos generales cognitivos humanos, en donde el lenguaje natural es solo el medio para estudiar estos fenómenos [contributors, 2020b].

Algunas de las dificultades que nos podemos encontrar en el PLN son:

- Ambigüedad: existe varios tipos de ambigüedad, alguna de estas son:
  - Léxica: palabra con múltiples significados, la elección de este se debe hacer en base al contexto o conocimiento básico.
  - Pragmática: oraciones que no significan lo que dicen, por ejemplo la ironía.
- Detección de separación entre las palabras: esta complicación viene dada porque en la lengua hablada no se suele hacer pausas entre palabra y palabra.

A la hora de trabajar en el área de PLN, se necesita un conjunto de textos denominado *Corpus*, y donde cada palabra en el *Corpus* se denomina *token*.

2.2 Machine Learning

El *Machine Learning* o *Aprendizaje Automático* es el subcampo de las ciencias de la computación y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que las computadoras aprendan. De forma más concreta, se trata de crear programas capaces de generalizar comportamientos a partir de una información suministrada en forma de ejemplos. Por lo que lo podemos pensar como un proceso de inducción del conocimiento, es decir, un método que permite obtener por generalización un enunciado general a partir de enunciados que describen casos particulares [Bishop, 2006].

El Machine Learning es la base de innumerables aplicaciones importantes, como por ejemplo:

- Motores de búsqueda
- Detección de spam en correos electrónicos
- Diagnósticos médicos
- Reconocimiento de voz y lenguaje escrito
- Detección de fraude en el uso de tarjetas de crédito
- Sistemas de recomendación
- Robótica
- etc.

Figura 2.1: Esquema básico del Machine Learning [Diplomatura en Ciencia de Datos, 2018b].

El Machine Learning se desglosa en diferentes tipos de algoritmos, los cuales se agrupan según su salida.

A continuación, en las siguientes subsecciones nombraremos y describiremos algunos de ellos.

2.2.1 Supervised Learning

El *Supervised Learning* o *Aprendizaje Supervisado* es una técnica para deducir una función a partir de datos de entrenamiento (datos etiquetados) [Russell and Norvig, 2009], esta función debe ser capaz de predecir el valor correspondiente a cualquier objeto de entrada válida después de haber visto una serie de ejemplos, los datos de entrenamiento [Mohri et al., 2012]. Para ello, tiene que generalizar a partir de los datos presentados a las situaciones no vistas previamente.

Dichos datos de entrenamiento consisten de pares de objetos (normalmente vectores) tal que el primer componente del par son los datos de entrada y el otro componente el resultado deseado.

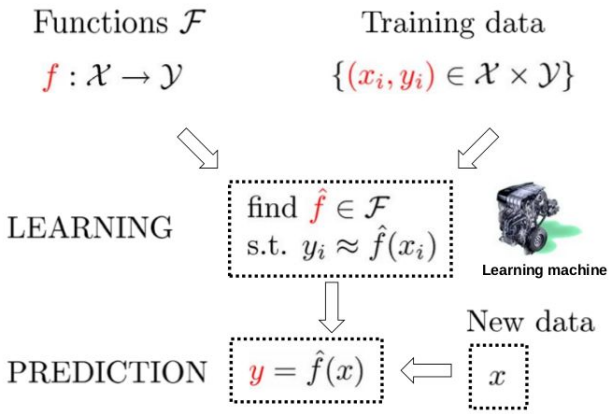


Figura 2.2: Vision General [Diplomatura en Ciencia de Datos, 2018b].

Según la salida de la función que se intenta deducir, el problema de Aprendizaje Supervisado se puede subdividir en dos:

Regresión

- Dados $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$
- Aprender una $f(x)$ que permita predecir y a partir de x
  - Si $y \in \mathbb{R}^n \to$ regresión

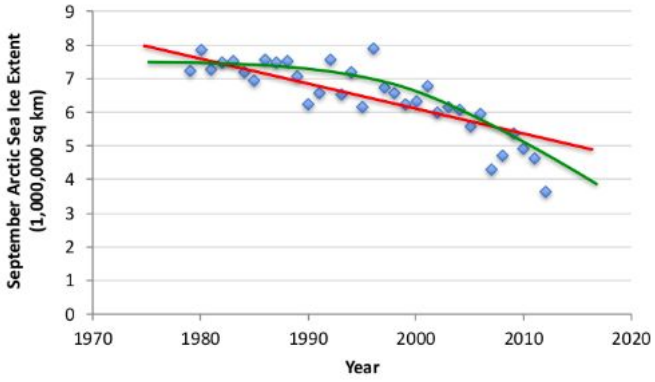


Figura 2.3: Ejemplo de Regresión [Diplomatura en Ciencia de Datos, 2018b].

Clasificación

- Dados $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$
- Aprender una $f(x)$ que permita predecir y a partir de x
  - Si y es categórica $\rightarrow$ clasificación

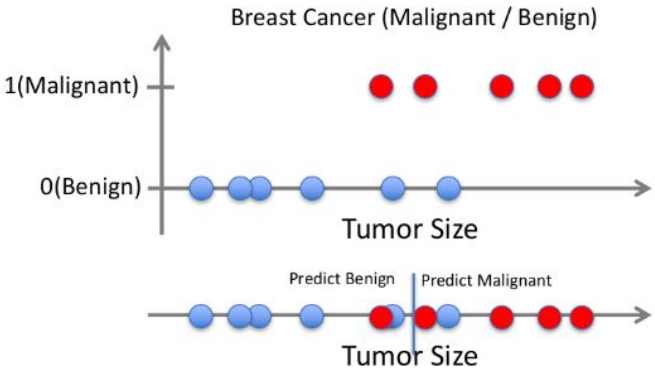


Figura 2.4: Ejemplo de Clasificación [Diplomatura en Ciencia de Datos, 2018b].

2.2.2 Unsupervised Learning

El *Unsupervised Learning* o *Aprendizaje no Supervisado* se distingue del Aprendizaje Supervisado por el hecho de que no hay un conocimiento a priori, es decir, no se cuenta con resultados etiquetados manualmente. Este tipo de Machine Learning nos permite trabajar con problemas sobre los cuales no hay información de cómo se debería ver la solución, en donde se genera un modelo a partir de las relaciones, descubriendo así una estructura presente entre los datos [Hinton and Sejnowski, 1999].

El objetivo del *Aprendizaje no Supervisado* podría ser descubrir patrones significativos presentes en los datos para que un experto pueda interpretarlos. Este tipo de uso se llama Análisis Exploratorio de Datos.

Algunas de las tecnologías relacionadas con este tipo de aprendizaje son:

- Clustering
- Vecinos más cercanos (recomendación)
- Reglas de asociación
- Detección de Anomalías

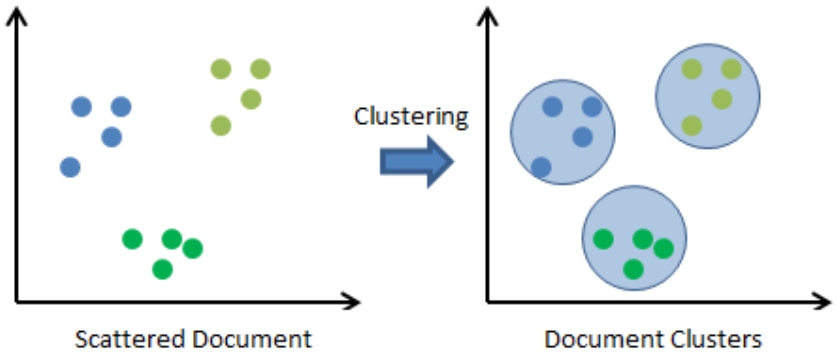


Figura 2.5: Ejemplo de Clustering [StackExchange, 2017].

Algunas de sus aplicaciones clásicas son:

- Segmentación de mercado (clientes)
- Caracterización de comportamiento de usuarios
- Detección de fallos en líneas de producción
- Detección de temas en documentos
- Detección de comunidades

2.2.3 Reinforcement Learning

El *Reinforcement Learning* o *Aprendizaje por Refuerzo* consiste en aprender que hacer - mapear situaciones a acciones- de manera tal de maximizar una señal numérica de recompensa.

Al aprendiz no se le especifica cuáles acciones ejecutar, sino que debe descubrir a través de prueba y error cuáles acciones producen la mayor cantidad de recompensa acumulada. En los casos más interesantes, las acciones pueden afectar no solo la recompensa inmediata, sino solamente el estado, recibiéndose una recompensa sólo en algunos estados o bien al final del episodio (Delayed-reward).

Se conoce simultáneamente como *Aprendizaje por Refuerzos* al problema de aprender por interacción, a la clase de métodos de solución de dicho problema, y al área de la *AI* (*Artificial intelligence*) que estudia el problema y los métodos de solución.

La idea de aprender por interacción con nuestro entorno es quizás la primera en aparecer cuando pensamos acerca de la naturaleza del aprendizaje. Por ejemplo, cuando un niño juega, mueve sus brazos, o mira alrededor, no tiene un "maestro" explicito, pero posee una conexión sensorial y motora con su entorno.

El ejercicio de dicha conexión produce información acerca de la relación causa-efecto y las consecuencias de las acciones que lleva a cabo, y respecto de qué hacer de manera tal de lograr objetivos. Así, aprender por interacción es una idea fundacional de muchas teorías del aprendizaje y la inteligencia.

El problema del Aprendizaje por Refuerzos puede ser formalizado empleando *Procesos de Decisión de Markov*.

La toma de decisiones secuencial involucra aprender sobre nuestro entorno y elegir acciones que maximizan el retorno esperado. El Aprendizaje por Refuerzos computacional, inspirado por estas ideas, las formalizo y produjo un impacto importante en robótica, Machine Learning y neurociencias.

El Aprendizaje por Refuerzos consiste en un agente que se encuentra en algún estado $s \in S$ inmerso en un entorno E y toma acciones $a \in A$ en busca de una meta. El agente puede ser modelado formalmente como una función f, que toma un historial de interacción como entrada, y devuelve una acción a tomar [Diplomatura en Ciencia de Datos, 2018a].

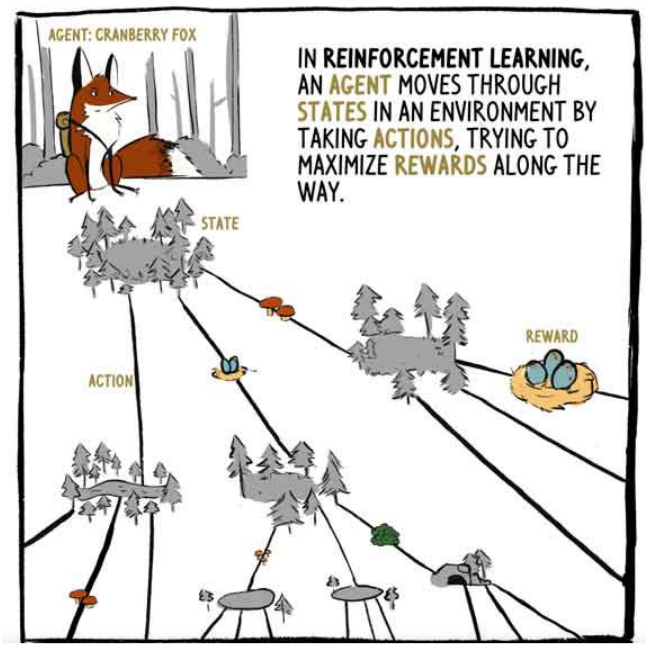


Figura 2.6: Elementos del Aprendizaje por Refuerzo [Gilman, 2018].

2.2.4 Deep Learning

El *Deep Learning* o *Aprendizaje Profundo* es una subcategoría del Machine Learning que intenta modelar abstracciones de alto nivel en datos usando arquitecturas compuestas de transformaciones no lineales múltiples, haciendo uso de Redes Neuronales para mejorar las soluciones ya propuestas en campos tales como el Speech Recognition (Reconocimiento Automático de Voz), Computer Vision (Visión por Computador) y el NLP [Schmidhuber, 2014].

El *Deep Learning* es parte de un conjunto más amplio de métodos basados en asimilar representaciones de datos. Una observación, como una imagen, puede ser representada en muchas formas, como por ejemplo un vector de píxeles, pero algunas representaciones hacen más fácil aprender tareas de interés como determinar si una imagen es una cara humana. Las investigación en este área intentan definir qué representaciones son mejores y cómo crear modelos para reconocer estas representaciones [Ciresan et al., 2012] [Krizhevsky et al., 2012].

Existen varias arquitecturas de *Deep Learning*, como por ejemplo MLP (Multilayer Perceptron), CNN (Convolutional Neural Network), RNN (Recurrent Neural Network).

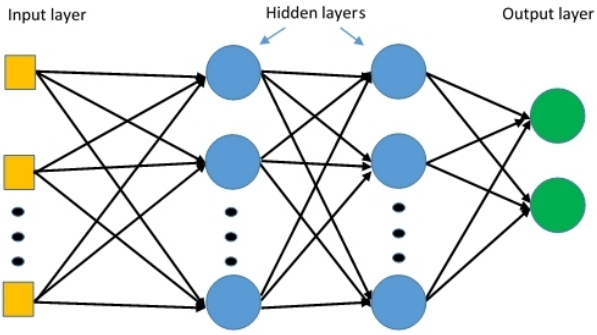


Figura 2.7: Arquitectura MLP [Glossary, 2017].

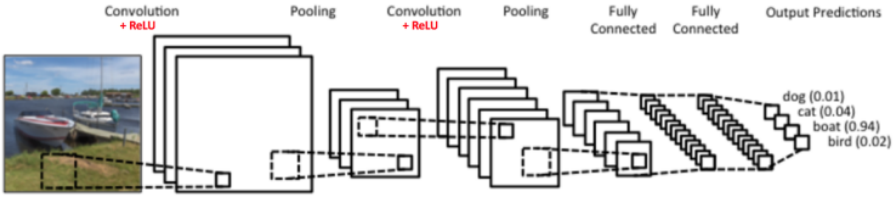


Figura 2.8: Arquitectura CNN [Karn, 2016].

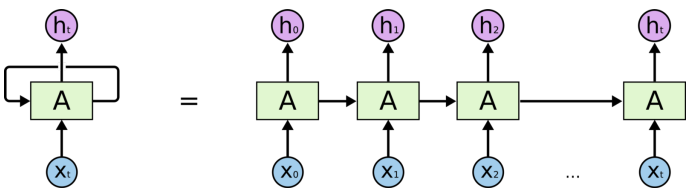


Figura 2.9: Arquitectura RNN [Radhakrishnan, 2017].

Una *Neural Network* o *Red Neuronal* (NN) es un sistema de programas y estructuras de datos que se aproxima al funcionamiento del cerebro humano. En un principio una NN se "adiestra" o se alimenta con grandes cantidades de datos y reglas acerca de las relaciones. Además como un NN suele implicar un gran número de procesadores funcionando en paralelo, teniendo cada uno de ellos su propia pequeña esfera de conocimiento y acceso a datos en su memoria local, hacen a estos problemas altamente paralelizables, por lo que la utilización de GPUs permite un aumento en el desempeño de varios órdenes de magnitud.

A diferencia del Machine Learning tradicional, el *Deep Learning* necesita menos esfuerzo humano en la creación de la representación a partir de la cual aprende el fenómeno. Esto lo logra por el gran número de combinaciones no lineales con las que calcula la salida. Pero sigue necesitando la misma cantidad de ejemplos etiquetados, o incluso muchos más, que una experiencia supervisada, dado que tiene una mayor cantidad de parámetros para optimizar.

2.3 Clasificadores Lineales

Un clasificador lineal es comúnmente usado en el aprendizaje supervisado para lograr determinar a qué clase pertenece un objeto, para tomar dicha decisión se basa en una combinación lineal de las características del objeto, comúnmente representadas en un vector llamado *feature vector*.

2.3.1 Support Vector Machine

Support Vector Machine o *Máquina de Soporte Vectorial*, comúnmente llamada **SVM** es un clasificador discriminativo definido formalmente por un hiperplano de separación. En otras palabras, dados los datos de entrenamiento etiquetados, el objetivo del algoritmo es encontrar un hiperplano óptimo en un espacio N-dimensional (N es longitud del *feature vector*) para separar claramente los puntos dados.

Los hiperplanos son límites de decisión que ayudan a clasificar los puntos de datos. Los puntos de datos que caen a ambos lados del hiperplano se pueden atribuir a diferentes clases. Además, la dimensión del hiperplano depende del número de características, si el número de características de entrada es 2 entonces el hiperplano es solo una línea, Si es 3 el hiperplano se convierte en un plano bidimensional. Se hace difícil imaginar cuando la cantidad de características excede las 3 [Gandhine, 2018].

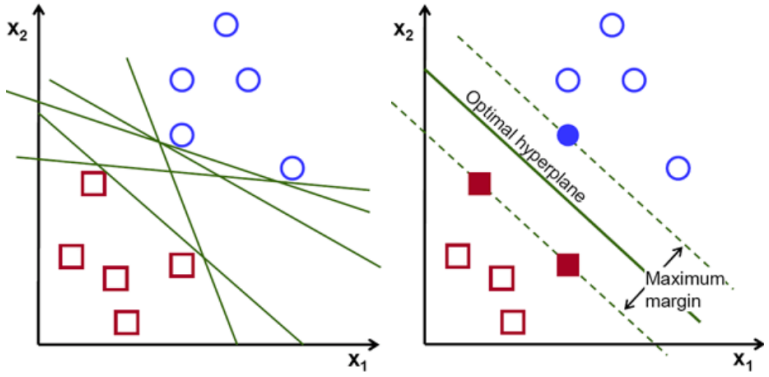


Figura 2.10: Ejemplo de hiperplano de una SVM [Gandhine, 2018].

El Soporte Vectorial son aquellos puntos de los datos que están más cerca del hiperplano e influyen en la posición y orientación del mismo, usando este Soporte Vectorial maximizamos el margen del clasificador. Eliminar dicho soporte cambiará la posición del hiperplano, por lo que estos son los puntos que nos ayudan a construir nuestro **SVM**.

Cuando se habla de hiperplano óptimo, se refiere a que hay muchos hiperplanos posibles que podrían elegirse, por lo que el óptimo sería aquel en que la distancia entre los puntos de las distintas clases sea máxima.

Para problemas multiclase, se puede aplicar en las SVM las técnicas *One vs All* o *One vs One*. Nos enfocaremos en la técnica *One vs All*, a donde para cada clase se construye un hiperplano entre esa clase y el resto de las clases. Por lo que si tenemos M clases, tendremos M SVMs. Luego para clasificar se realizan M predicciones, una por cada SVM, y nos quedamos con la región que más visitas recibe.

2.3.2 Logistic Regression

Logistic Regression o *Regresión Logística* es un algoritmo de clasificación usado cuando la variable objetivo o dependiente es categórica, por ejemplo, predecir si un correo electrónico es spam o no, ya que describe y estima la relación entre dicha variable binaria dependiente y las variables independientes. A diferencia de la regresión lineal, puede predecir directamente las probabilidades [Swaminathan, 2018].

Podemos representar lo que hace la Regresión Logística en la siguiente figura:

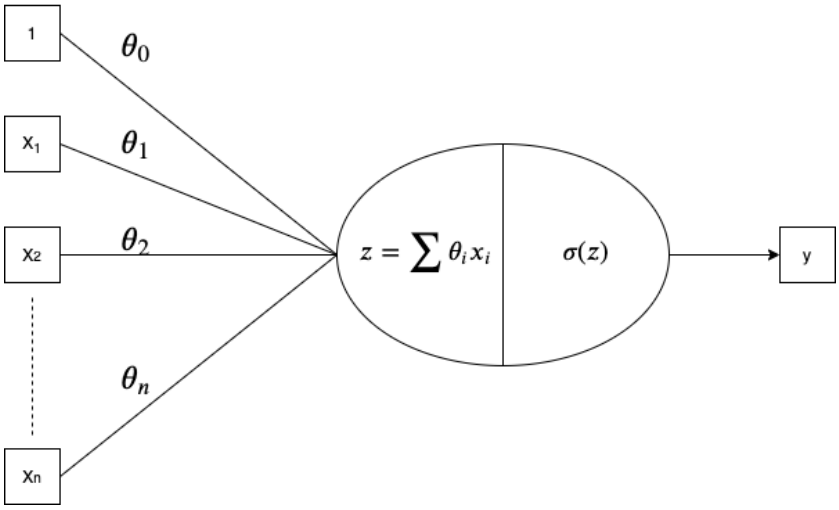


Figura 2.11: Regresión Logística representada como una red neuronal de una sola neurona.

En donde los valores de x corresponden a las features de nuestro problema. Mientras que y es la predicción, que vendría a ser la probabilidad.

Matemáticamente, lo podemos formular de esta forma:

$$y = h_\theta(x) = \sigma(\theta^T X) = \sigma(\sum_i^n \theta_i x_i) = \sigma(\theta_0 + \theta_1 x_1 + ... + \theta_n x_n)$$

Como podemos ver, la Regresión Logística tiene dos partes, en la primera se realiza una combinación lineal del vector de features x y el vector de coeficientes θ y en la segunda se aplica la función logística o sigmoide al resultado de la combinación lineal.

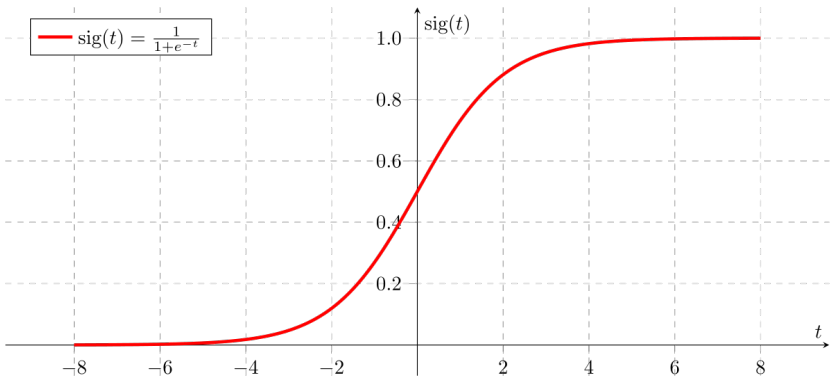


Figura 2.12: Función Logística o Sigmoide.

Características de la función logística:

- Matemáticamente, se puede expresar: $\sigma(z) = \frac{1}{1+e^{-z}}$
- Está acotada entre los valores 0 y 1.
- Podemos interpretar sus resultados como probabilidades.
- Para problemas de clasificación binaria podemos definir un umbral, en donde los valores menores al umbral corresponden a la clase 0 y los superiores al umbral a la clase 1.

Podemos plantear la hipótesis de la Regresión Logística Binaria de la siguiente manera:

Dado un conjunto de entrenamiento:

$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(d)}, y^{(d)})\} \text{ donde } x^{(i)} \in \mathbb{R}^n,\ y^{(i)} \in \{0, 1\}$$

Tenemos que

$$h_\theta(x) = \sigma(\theta^T x)$$

Donde

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \qquad x = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Nota: El termino θ_0 es comúnmente llamado *bias*.

En el caso de que nuestro problema sea, por ejemplo, clasificar si un correo electrónico es spam, esto será si y sólo si $\theta^T x \geq 0$. Esto se debe a que el umbral de la Regresión Logística se establece en $\sigma(z) = 0.5$.

La función de costo de la Regresión Logística para una sola instancia se define como sigue:

$$cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{if } y = 1 \\ -log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

Podemos escribir la función de costo para todas las instancias de la siguiente manera:

$$\begin{aligned} J(\theta) &= \sum_{i=1}^{n} cost(h_\theta(x^{(i)}), y^{(i)}) \\ &= -\sum_{i=1}^{n} [y^{(i)} * log(h_\theta(x^{(i)})) + (1 - y^{(i)}) * log(1 - h_\theta(x^{(i)}))] \end{aligned}$$

Para calcular θ se debe resolver el problema de optimización:

$$arg \min_\theta J(\theta)$$

Hay una variedad de métodos que se pueden usar para resolver este problema de optimización, uno de ellos es el Descenso por Gradiente. Como la función de costo es convexa, este método garantiza encontrar el mínimo global de la función.

Existen otros tipos de Regresiones Logísticas, las cuales son:

- Regresión Logística Multinomial: la variable objetivo tiene tres o más categorías nominales.
- Regresión Logística Ordinal: la variable objetivo tiene tres o más categorías ordinales.

2.4 Métricas

Para poder evaluar y comprender que tan preciso es un modelo que clasifica, se utilizan métricas de rendimiento. Para esto es necesario que el dataset contenga ejemplos ya anotados, es decir, que cada observación esté con su correspondiente resultado. De esta forma cuando se clasifica el dataset, podemos comparar cada resultado dado por modelo con el resultado original.

Normalmente uno pensaría que un modelo de clasificación es mejor que otro si su cantidad de aciertos es mayor, pero esta es una de las muchas métricas que son usadas a hora de definir la performance y calidad de clasificación de un modelo.

A continuación nombraremos y describiremos brevemente algunas de la métricas más usadas, pero antes definamos algunos algunos conceptos:

- **True Positives (TP)**: El modelo predice correctamente la clase positiva, es decir, selecciona algo que debe ser seleccionado.
- **True Negative (TN)**: El modelo predice correctamente la clase negativa, , es decir, no selecciona algo que no debe ser seleccionado.
- **False Positives (FP)**: El modelo predice incorrectamente la clase positiva, es decir, selecciona algo que no debe ser seleccionado.
- **False Negatives (FP)**: El modelo predice incorrectamente la clase negativa, es decir, no selecciona algo que debe ser seleccionado.

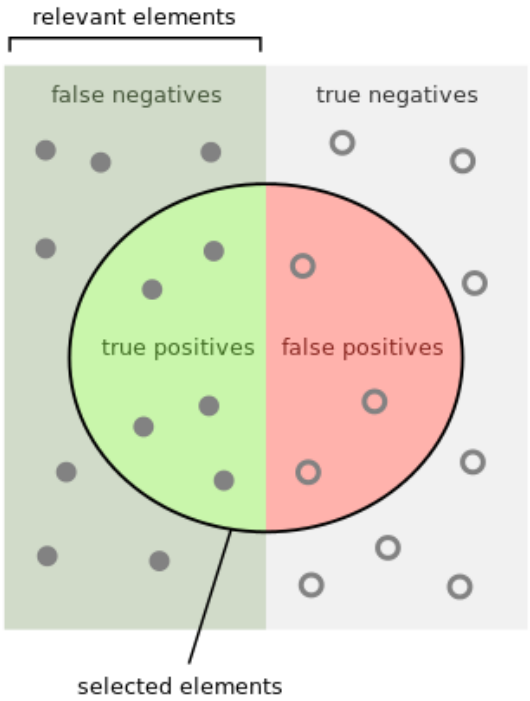


Figura 2.13: Resultados posibles para modelos de clasificación [contributors, 2020c].

2.4.1 Matriz de Confusión

Una *Matriz de Confusión* es una herramienta que permite la visualización del desempeño de un algoritmo que se emplea en aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Uno de los beneficios de las matrices de confusión es que facilitan ver si el sistema está confundiendo dos clases [Powers, 2011].

A continuación mostraremos una Matriz de Confusión para un problema de clasificación binaria:

		Predicted	
		Yes	**No**
Actual	**Yes**	TP	FN
	No	FP	TN

Esta es fácilmente extendible a un problema de clasificación multiclase.
Miremos la siguiente matriz de ejemplo, en donde tenemos 4 aviones, 15 autos y 8 tractores:

		Predicted		
		Avión	**Auto**	**Tractor**
Actual	**Avión**	3	1	0
	Auto	1	10	4
	Tractor	0	3	5

En la matriz ejemplo podemos notar que:

- De 4 aviones el sistema predijo que 1 era un auto
- De 15 autos el sistema predijo que 1 eran un avión y 4 eran tractores
- De 8 tractores el sistema predijo que 3 eran autos

A partir de la matriz se puede ver que el sistema tiene problemas distinguiendo entre autos y tractores, pero que puede distinguir razonablemente bien entre aviones y autos, además de distinguir perfectamente los aviones de los tractores.

2.4.2 Accuracy

La *Accuracy* o *Exactitud* es una métrica para evaluar modelos de clasificación. Informalmente, la accuracy es la fracción de predicciones que el modelo realizó correctamente.

Formalmente, la accuracy tiene la siguiente definición:

$$Accuracy = \frac{\#Predicciones\ Correctas}{\#Total\ Predicciones}$$

En la clasificación binaria, la *Accuracy* también se puede calcular en términos de la *Matriz de Confusión*:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Para un problema de clasificación multiclase, dada una *Matriz de Confusión* M, la *Accuracy* se calcula de la siguiente manera:

$$Accuracy = \frac{\sum M_{ii}}{\sum M_{ij}}$$

Nota: Un modelo que no produce **FP** ni **FN** tiene 100% de *Accuracy*.

Ejemplo

Supongamos que queremos saber si un correo electrónico es spam o no, para lo cual disponemos de 100 correos, de los cuales 80 no son spam y 20 si lo son.

Nuestro modelo detecta genera la siguiente matriz de confusión:

		Predicted	
		Spam	**Not Spam**
Actual	**Spam**	5	15
	Not Spam	10	70

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{5 + 70}{5 + 70 + 10 + 15} = \frac{75}{100} = 0.75$$

Es decir, que cuando el modelo predice de si un correo es spam o no, el 75% de las veces acierta.

2.4.3 Precision

La *Precision* intenta responder a la siguiente pregunta: ¿Qué proporción de identificaciones positivas fue correcta?.

La *Precision* se define de la siguiente manera para problemas de clasificación binaria:

$$Precision = \frac{TP}{TP + FP}$$

Mientras que para problemas de clasificación multiclase se define, dada una *Matriz de Confusión* M, la *Precision* para cada clase i de la siguiente manera:

$$Precision_i = \frac{M_{ii}}{\sum_j M_{ji}}$$

Nota: Un modelo que no produce **FP** tiene 100% de *Precision*.

Ejemplo

Siguiendo el mismo ejemplo mencionado en la subsección 2.4.2, calculemos la *Precision*:

$$Precision = \frac{TP}{TP + FP} = \frac{5}{5 + 10} = \frac{5}{15} = 0.33$$

Es decir, que el 33% de las veces que el modelo predice que un correo electrónico es spam, es correcto.

2.4.4 Recall

La *Recall* ó *Exhaustividad* intenta responder a la siguiente pregunta: ¿Qué proporción de positivos reales se identificó correctamente?.

La *Recall* se define de la siguiente manera para problemas de clasificación binaria:

$$Recall = \frac{TP}{TP + FN}$$

Mientras que para problemas de clasificación multiclase se define, dada una *Matriz de Confusión* M, la *Recall* para cada clase i de la siguiente manera:

$$Recall_i = \frac{M_{ii}}{\sum_j M_{ij}}$$

Nota: Un modelo que no produce **FN** tiene 100% de *Recall*.

Ejemplo

Siguiendo el mismo ejemplo mencionado en la subsección 2.4.2, calculemos la *Recall*:

$$Recall = \frac{TP}{TP + FN} = \frac{5}{5 + 15} = \frac{5}{20} = 0.25$$

Es decir, que de las 25% de las veces que el correo electrónico es spam, el modelo lo predice correctamente.

2.4.5 F1 Score

La *F1* es una combinación de las métricas *Precision* y *Recall* también llamada Media Armónica, cuya formula es la siguiente:

$$F1 = 2 * \frac{1}{\frac{1}{Recall} + \frac{1}{Precision}}$$

$$= 2 * \frac{Precision * Recall}{Precision + Recall}$$

Esta métrica es el objetivo de la mayoría de los modelos, ya que refleja el desbalance entre la *Precision* y la *Recall* de un clasificador, cualidad muy negativa para cualquier modelo.

Ejemplo

Siguiendo el mismo ejemplo mencionado en la subsección 2.4.2 y usando los resultados de las subsecciones 2.4.3 y 2.4.4, calculemos la *F1*:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} = 2 * \frac{0.33 * 0.25}{0.33 + 0.25} = 0.28$$

2.4.6 Macro-Precision, Macro-Recall y Macro-F1

Las métricas de *Macro-Precision* y *Macro-Recal*l son los promedios de la *Precision* y *Recall* respectivamente.

La definición formal es:

$$Macro - Precision = \frac{\sum_i^n Precision_i}{n}$$

$$Macro - Recall = \frac{\sum_i^n Recall_i}{n}$$

En palabras, son el promedio de la métrica sobre cada clase.

Mientras que la *Macro-F1* es el cálculo de la *F1* usando la *Macro-Precision* y *Macro-Recall*:

$$Macro - F1 = 2 * \frac{Macro - Precision * Macro - Recall}{Macro - Precision + Macro - Recall}$$

Estas métricas generalmente son las más usadas para problemas multiclase.

2.5 Sistemas Previos

El Análisis de Sentimiento es una tarea común de categorización de texto, una revisión de una película, libro o producto en la web expresa el sentimiento del autor hacia el producto, mientras que un texto editorial o político expresa el sentimiento hacia un candidato o acción política. Extraer el sentimiento del consumidor o público es, por lo tanto, relevante para los campos del marketing y la política.

Desde el año 2012 la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN) realiza el Taller sobre Análisis Semántico (TASS), cuyo objetivo es promover la investigación sobre el Análisis de Sentimiento en español. El TASS además de promover el Análisis de Sentimiento, promueve otras tareas relacionadas con el análisis semántico en español.

Existen diferentes versiones de la tarea Análisis de Sentimiento, la más simple es clasificar en dos clases (*Sentimiento positivo* y *Sentimiento negativo*), pero existen otras más complejas en donde se clasifica en cuatro o seis clases diferentes.

En [Murillo and Raventós, 2016], Casasola Murillo y otros introdujeron un sistema que se fundamenta en tres elementos básicos que son: la normalización del texto en la etapa de preprocesamiento identificando los potenciales marcadores de énfasis presentes en el mismo, la creación de vectores de características de dimensión reducida para disminuir el efecto de la dispersión de los datos, y la exploración del impacto del uso de diccionarios de polaridad que se generan mediante la utilización de diferentes modelos de representación del lenguaje asociados tanto al contexto local como global de los datos, usando una adaptación propia del algoritmo de Turney [Turney, 2002].

En [Cerón-Guzmán, 2016], Cerón-Guzmán combinó diversos sistemas con la correlación absoluta más baja entre sí. Estos sistemas son capaces de tratar con formas léxicas no estándar en los tweets, con el fin de mejorar la calidad del análisis del lenguaje natural. Para realizar la clasificación de polaridad, el enfoque utiliza características básicas que han probado su poder discriminativo, así como características de n-gramas de palabras y caracteres. Luego, las salidas de clasificadores de Regresión logística, que pueden ser etiquetas de clase o probabilidades para cada clase, se utilizan para construir conjuntos de clasificadores combinados.

En [Quirós et al., 2016], Quirós y otros estudiaron el uso de word embeddings (también conocidas como vectores de palabras) para representar tweets y luego examinar varios algoritmos de machine learning para clasificarlos. Los word embeddings pueden ayudar a capturar las relaciones semánticas y sintácticas de las palabras correspondientes, mostrando resultados prometedores en tareas de PLN, como el reconocimiento de entidades nombradas, extracción de relaciones, análisis de sentimiento o parseo.

En [Reyes-Ortiz et al., 2017], Reyes-Ortiz y otros propusieron un sistema que utiliza aprendizaje automático, el algoritmo de SVM y lexicones de polaridades semánticas a nivel de lemas para el español, como son iSOL e ML-SentiCon además de usar un lexicón propio obtenido a partir del corpus InterTASS. Las características extraídas de los lexicones son representadas mediante el modelo de BoW y son ponderadas utilizando la frecuencia de los términos, la cual expresa la ocurrencia del lema en cada tweet. Mostrando resultados prometedores en la experimentación con el uso de lexicones para el análisis de sentimiento a nivel de tweet.

En [Hurtado et al., 2017], Hurtado y otros atacaron el problema de clasificación utilizado redes neuronales debido a que han obtenido buenos resultados en tareas similares, explorando diferentes topologías de redes neuronales, como son Multilayer Perceptron (MLP) y con redes neuronales recurrentes, típicamente Long Short Term Memory (LSTM) y stack de redes convolucionales (CNN) y LSTM, así como diferentes tipos de representaciones de los tweets, como *Bag-of-Words* (BoW), *Bag-of-Chars* (BoC), colapsado de *embeddings* y representaciones secuenciales formadas por secuencias de embeddings o de vectores *one-hot* sobre palabras y caracteres.

En [Moctezuma et al., 2017], Moctezuma y otros propusieron una solución que se basa en un conjunto de clasificadores SVM combinados en un modelo no lineal creado con Programación Genética (GP). Utilizan B4MSA[1], el cual es un sistema de aprendizaje supervisado baseline basado en el clasificador SVM, un esquema de ponderación de términos basado en entropía y EvoDAG (Evolving Directed Acyclic Graph), con la particularidad de producir clasificadores de sentimiento que están débilmente vinculados a métodos dependientes del lenguaje. Un sistema GP que combina todos los valores de decisión predichos por los sistemas B4MSA.

[1] https://github.com/INGEOTEC/b4msa

Capítulo 3

Corpus

En este capítulo describiremos los corpus existentes con los cuales trabajamos, además del proceso que seguimos para construir nuestro propio corpus con tweets que hablan sobre el fútbol argentino.

Cada tweet está etiquetado con alguno de los siguientes labels:

- **P**: Sentimiento positivo
- **N**: Sentimiento negativo
- **NEU**: Sentimiento neutral
- **NONE**: Sin ningun sentimiento

3.1 Corpus de la SEPLN

Para el desarrollo de este trabajo se utilizaron los corpus que provee la SEPLN [1] (Sociedad Española para el Procesamiento del Lenguaje Natural) para el desarrollo del TASS[2] (Taller de Análisis Semántico). Este taller tiene varias ediciones, que van del año 2012 hasta la actualidad donde se está desarrollando la edición 2020, nosotros trabajamos con los corpus de la edición 2018[3], en la cual nos encontramos los siguientes cuatro corpus escritos en diferentes variantes del español:

- GeneralTASS
- InterTASS Spain
- InterTASS Costa Rica
- InterTASS Peru

Al ser corpus de distintas procedencias, se exhibe una gran cantidad de diferencias léxicas.

El etiquetado de estos corpus fue realizado de forma semiautomática, en donde primero se etiquetan los tweets usando un modelo de machine learning baseline para que luego anotadores humanos verifiquen todas las etiquetas.

[1] http://www.sepln.org/sepln
[2] http://tass.sepln.org/
[3] http://tass.sepln.org/2018/

Además es necesario aclarar que estos corpus están divididos en tres partes:

- Train
- Development
- Test

Donde la parte del *Train* se usa para entrenar los modelos, la parte del *Development* es usada para validarlo, y por último la parte del *Test* es usada para que luego la TASS evalúe nuestro modelo.

La edición de la cual se obtienen los corpus es importante debido a que estos varían durante las mismas, por ejemplo en la edición 2019[4] se agregaron nuevos tweets a los corpus ya existentes y se realizó una nueva revisión de las partes *Train*, *Development* y *Test*, además de agregar nuevos corpus con tweets pertenecientes a Uruguay y México. En la edición 2020[5], se trabajan con 3 labels en vez de 4, ya que la etiqueta NEU incluirá a la etiqueta NONE. Por lo que los corpus son distintos entre sí en las diferentes ediciones.

Además es necesario mencionar que los corpus se encuentran encodeados en el formato XML, a continuación hay un ejemplo:

```
<tweet>
  <tweetid>000000000000000000</tweetid>
  <user>0000000000</user>
  <content>Que lindo dia para salir a correrrrrrrr :D</content>
  <date>2020-02-17 20:00:00</date>
  <lang>es</lang>
  <sentiment>
    <polarity>
      <value>P</value>
    </polarity>
 </sentiment>
</tweet>
```

Listing 3.1: Tweet de ejemplo.

Como se puede ver, cada tweet incluye su ID (*tweetid*), el ID del usuario (*user*) y la fecha de creación (*date*).

Debido a restricciones en las Políticas y Acuerdos de la API de Twitter[6], está prohibido redistribuir un corpus que incluya contenido de texto o información sobre los usuarios. Sin embargo, es válido si esos campos se eliminan y en su lugar se proporciona el ID (*tweetid* y *user*).

[4]https://competitions.codalab.org/competitions/23005
[5]http://tass.sepln.org/2020/
[6]https://developer.twitter.com/en/developer-terms/agreement-and-policy

3.1.1 GeneralTASS

El corpus *GeneralTASS* contiene más de 68000 tweets, recolectados desde el 02-12-2011 hasta el 10-04-2012. El contexto de extracción de los mismo tiene un sesgo centrado en el español, sin embargo la diversidad de nacionalidades de los autores de los tweets hace hace que el corpus alcance una cobertura global en el mundo de habla hispana.

El corpus está dividido en un ***Training Set*** que consiste de 7219 tweets (aproximadamente 10%) y un ***Test Set*** con 60798 tweets (90%).

Nosotros usaremos solamente el ***Training Set*** como aditivo para entrenar en los corpus que nombraremos más adelante, en el cual los tweets están distribuidos de la siguiente manera:

	# Tweets	**%**
P	1652	22.08
P+	1232	17.07
N	847	11.73
N+	1335	18.50
NEU	670	9.28
NONE	1483	20.54
Total	7219	100

Tabla 3.1: Estadísticas del corpus GeneralTASS.

Como podemos ver en la tabla anterior, hay 6 labels en vez de 4, definidos de la siguiente manera:

- **P+**: Sentimiento fuertemente positivo
- **P**: Sentimiento positivo
- **N+**: Sentimiento fuertemente negativo
- **N**: Sentimiento negativo
- **NEU**: Sentimiento neutral
- **NONE**: Sin ningun sentimiento

Por lo que para poder trabajar con 4 labels, consideramos a los labels **P+** y **P** como **P** y **N+** y **N** como **N**, quedando la siguiente tabla:

	# Tweets	%
P	2884	39.95
N	2182	30.23
NEU	670	9.28
NONE	1483	20.54
Total	7219	100

Tabla 3.2: Estadísticas del corpus GeneralTASS transformado.

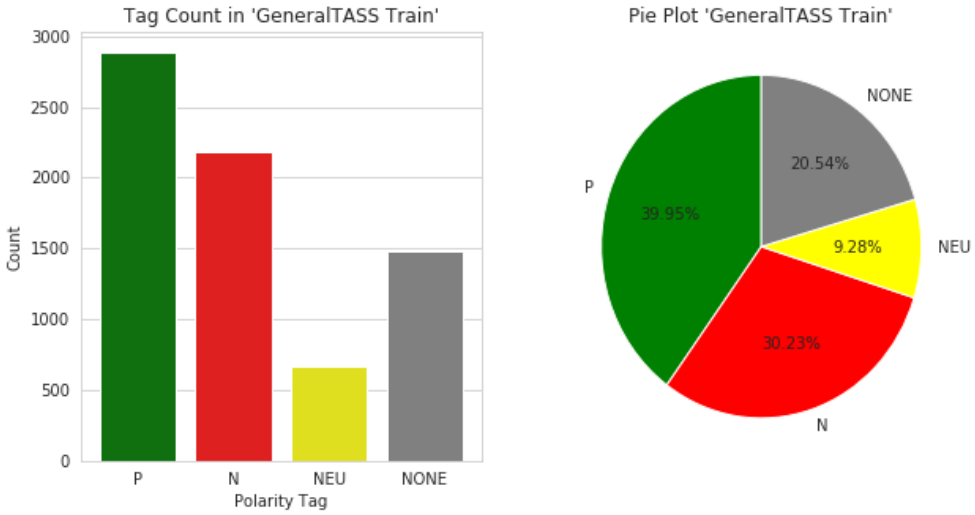


Figura 3.1: Gráficos de las estadísticas del corpus GeneralTASS.

3.1.2 InterTASS Spain

Este corpus consiste de 3413 tweets pertenecientes a España, recolectados desde el 18-07-2016 hasta el 09-01-2017.

Los tweets se encuentran distribuidos de la siguiente manera:

	Train		**Development**	
	# Tweets	**%**	**# Tweets**	**%**
P	318	31.55	156	30.83
N	418	41.47	219	43.28
NEU	133	13.19	69	13.64
NONE	139	13.79	62	12.25
Total	1008	100	506	100

Tabla 3.3: Distribución de tweets en InterTASS Spain Train y Development según su polaridad.

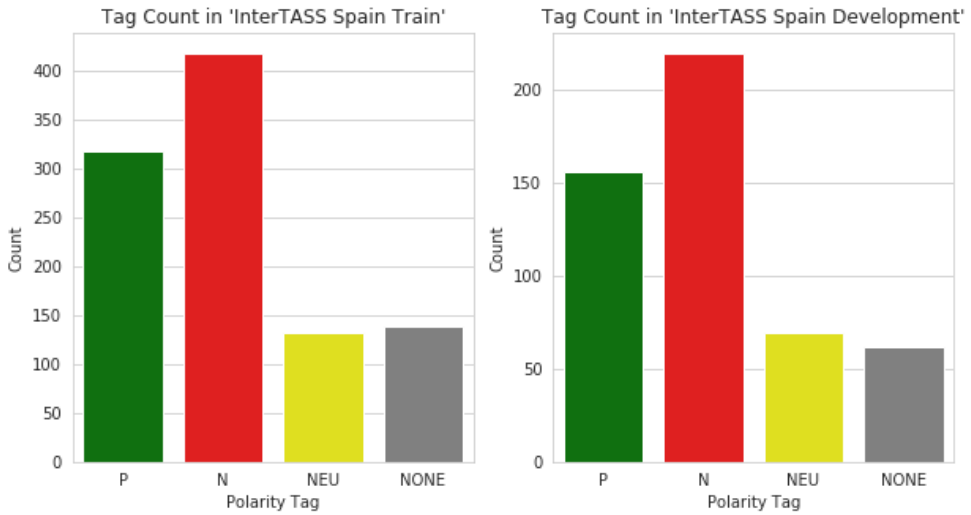


Figura 3.2: Gráficos de las distribuciones de tweets en InterTASS Spain Train y Development según su polaridad.

El conjunto de *Test* consiste de 1899 tweets, pero las etiquetas no se encuentran disponibles.

3.1.3 InterTASS Costa Rica

Este corpus consiste de 2333 tweets pertenecientes a Costa Rica, recolectados desde el 18-07-2016 hasta el 12-01-2017.

Los tweets se encuentran distribuidos de la siguiente manera:

	Train		**Development**	
	# Tweets	**%**	**# Tweets**	**%**
P	230	28.75	93	31
N	311	38.87	110	36.67
NEU	94	11.75	39	13
NONE	165	20.62	58	19.33
Total	800	100	300	100

Tabla 3.4: Distribución de tweets en InterTASS Costa Rica Train y Development según su polaridad.

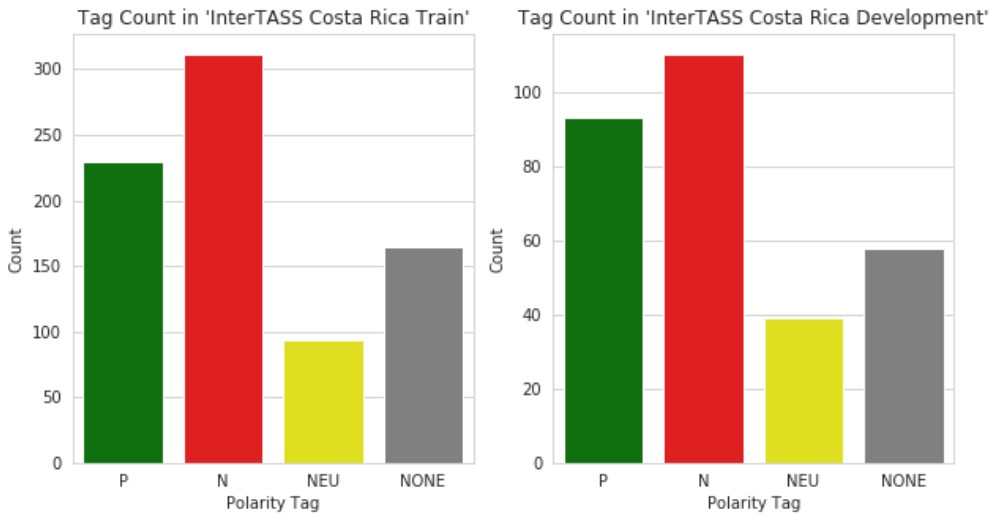


Figura 3.3: Gráficos de las distribuciones de tweets en InterTASS Costa Rica Train y Development según su polaridad.

El conjunto de *Test* consiste de 1233 tweets, pero las etiquetas no se encuentran disponibles.

3.1.4 InterTASS Peru

Este corpus consiste de 2333 tweets pertenecientes a Perú, recolectados desde el 18-07-2016 hasta el 12-01-2017.

Los tweets se encuentran distribuidos de la siguiente manera:

	Train		**Development**	
	# Tweets	**%**	**# Tweets**	**%**
P	231	23.1	95	19
N	242	24.2	106	21.2
NEU	166	16.6	61	12.2
NONE	361	36.1	238	47.6
Total	1000	100	500	100

Tabla 3.5: Distribución de tweets en InterTASS Peru Train y Development según su polaridad.

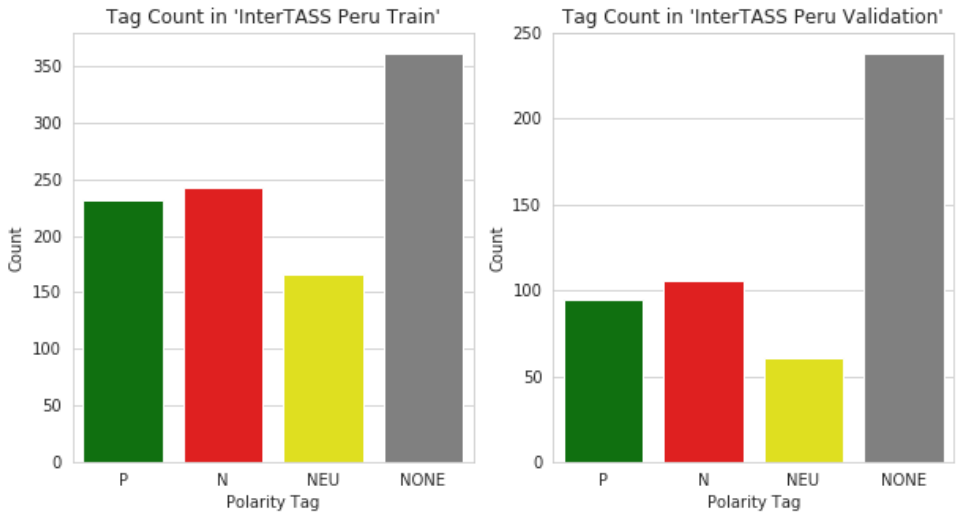


Figura 3.4: Gráficos de las distribuciones de tweets en InterTASS Peru Train y Development según su polaridad.

El conjunto de *Test* consiste de 1428 tweets, pero las etiquetas no se encuentran disponibles.

3.2 Construcción del Corpus de Fútbol Argentino

Para la construcción de nuestro propio corpus de tweets en español de Argentina, se siguieron una serie de pasos, como son el proceso de recolección, filtrado y etiquetado de los tweets, los cuales describiremos en las siguientes secciones. Para finalmente obtener un corpus que consiste de 1994 tweets.

3.2.1 Recolección

Para la obtención de los tweets se utilizo la libreria de Python Tweepy[7], la cual nos provee una manera más sencilla de usar API de Twitter.

Los tweets que se obtuvieron pertenecen al tópico futbolístico, más específicamente al fútbol argentino, para ello la librería Tweepy nos permite personalizar la búsqueda de los tweets usando keywords como por ejemplo los nombres de los equipos argentinos, como son Boca, River, Talleres, Belgrano, etc.

Obtuvimos dos dataset en fechas distintas. A continuación detallamos las características de cada dataset:

- Dataset 1:
  - Fecha de descarga: 31 de octubre de 2017
  - Cantidad de tweets: 26034
  - Acontecimiento:
    - Competición: Copa Conmebol Libertadores 2017
    - Partido: River Plate vs Lanús (Semifinal vuelta)
    - Resultado: Lanús 4 - River Plate 2
- Dataset 2:
  - Fecha de descarga: 10 de diciembre de 2017
  - Cantidad de tweets: 14473
  - Acontecimiento:
    - Competición: Superliga Argentina 2017-18
    - Partido: Estudiantes de La Plata vs Boca Juniors (Fecha 12 - Último Partido)
    - Resultado: Estudiantes de La Plata 0 - Boca Juniors 1 (Gol del colombiano Wilmar Barrios)

Finalmente unimos los dos datasets para obtener un total de 40507 tweets.

[7] https://www.tweepy.org/

3.2.2 Filtrado

Luego de la recolección de los tweets, procedimos a realizar un filtrado sobre los tweets del dataset para quedarnos solamente con aquellos que hablen sobre fútbol argentino, esto es debido a que las keywords mencionadas anteriormente hacen que el scraper atrape tweets que no hablen de fútbol, por ejemplo usando la keyword "*boca*", se atrapan tweets que utilizan la palabra boca haciendo referencia a la parte del cuerpo humano, otro ejemplo es que al usar la palabra estudiantes, nos encontramos con tweets que hablan sobre los estudiantes de las universidades.

Para esta etapa de filtrado participaron 7 personas, en donde la tarea consistía en analizar cada tweet del dataset con el fin de determinar si el mismo hablaba sobre algún tema relacionado al fútbol argentino o no.

Para ello, la persona debe etiquetar el tweet las etiquetas **Y** (Yes) si el tweet hablaba del fútbol argentino, **N** (No) si el tweet no hablaba del futbol argentino o **U** (Unknown) si no tiene certeza de que el tweet sea **Y** o **N**. A continuación describiremos más en detalle los criterios que se usaron para el uso de cada etiqueta:

- **Y**: El tweet habla explícita o implícitamente de uno o más clubes de fútbol argentino, o de una situación que involucra a los clubes en un contexto futbolístico. Esto incluye referencias a partidos, jugadores, árbitros, hinchada, dirigentes, etc.
- **N**: El tweet NO habla explícita o implícitamente de uno o más clubes de fútbol argentino, o de una situación que involucra a los clubes en un contexto futbolístico.
- **U**: El anotador no puede deducir de la información del tweet si la etiqueta es **Y** o **N**.

En la sección 7.1 se puede encontrar el documento que usó cada persona como guía para el filtrado. En este documento también podemos encontrar la Tabla 7.1 con ejemplos de tweets anotados.

Una estrategia que utilizamos para obtener un mejor filtrado, fue duplicar el dataset para que sobre un mismo tweet haya dos anotaciones.

Veamos algunas estadísticas sobre el filtrado del dataset:

Etiqueta	**Anotación 1**	**Anotación 2**
Y	20348	19979
N	18536	19371
U	1623	1157
Total	40507	40507

Tabla 3.6: Cantidad de etiquetas de filtrado por anotación sobre el dataset.

Combinación	Cantidad	Porcentaje (%)
Y & Y	14986	37.0
N & N	14293	35.29
U & U	78	0.19
Y & U	1829	4.52
Y & N	8526	21.05
N & U	795	1.96
Total	40507	100

Tabla 3.7: Combinaciones de etiquetas de filtrado puestas sobre el dataset.

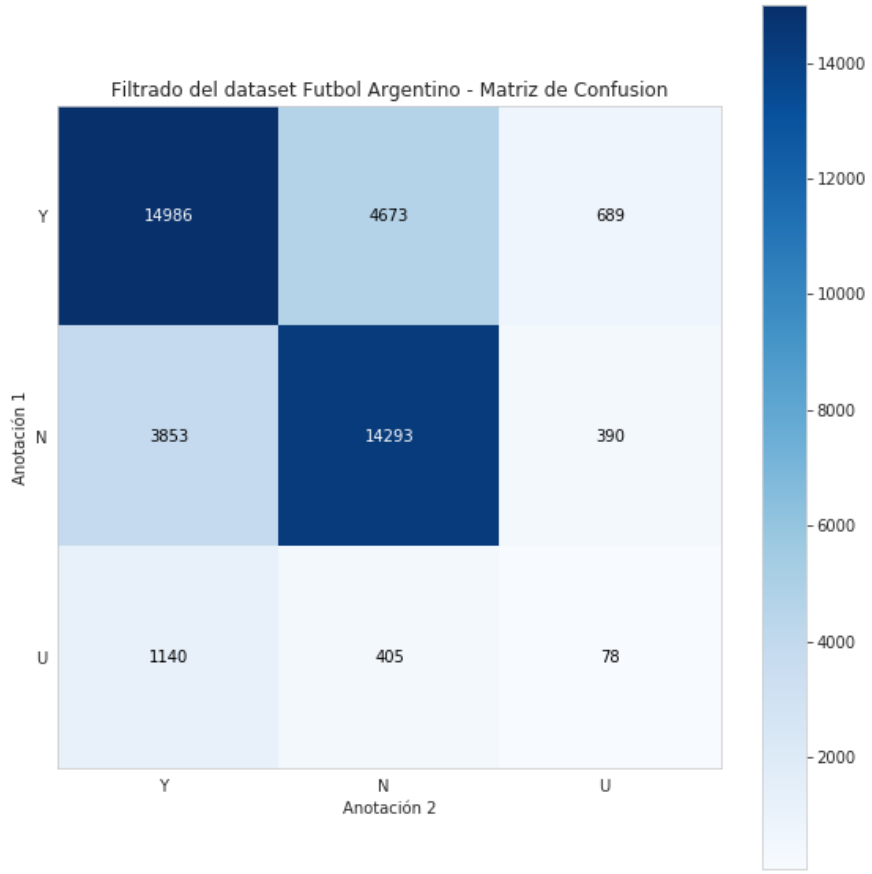


Figura 3.5: Matriz de confusión entre las anotaciones del filtrado hechas sobre el dataset.

En la Figura 3.5 podemos apreciar que entre las dos anotaciones hechas en el dataset, hubo una alta coherencia en las etiquetas **Y** y **N**, por lo que en la mayoría de los casos se pudo distinguir cuales tweets hablaban de fútbol argentino y cuáles no. Sin embargo también hubo una significativa confusión entre las etiquetas **Y** y **N**, lo cual puede deberse deberse al hecho de que las personas que participaron en el filtrado no tienen tanto conocimiento del ámbito del fútbol argentino.

Veamos algunos tweets de ejemplos y el contexto en cual están escritos:

Tweet	**Contexto**
Cuando tu equipo clasifica por primera vez a una final de Copa Libertadores. Emociones que causa el deporte más lindo del mundo. https://t.co/4153uCXsLT	Se está hablando implícitamente de Lanús que, después de ganarle a River, pasó a la final de la Copa Libertadores 2017
Se debería haber revisado la jugada del penal" https://t.co/AuzpGrNVwS https://t.co/MacXJhzieE	Se está hablando implícitamente de una jugada del primer tiempo, cuando River ganaba 2 a 0, en la que no cobran un penal a favor de River por una mano en el área de Lanús
Que les quede bien claro. Podemos ganar o perder, pero JAMÁS abandonar. Lo dicen @funesmoriofi25 y @SebadriussiOk https://t.co/Rd3eRacyBY https://t.co/wCIHGgdBVZ	Se está hablando implícitamente de una muestra de apoyo a River, nombrando a dos ex jugadores de River, Ramiro Funes Mori y Sebastián Driussi
Lo que pasó ayer en la Copa Libertadores fue un fiasco, mal el arbitro y sus 6 practicantes	Se está hablando del mal arbitraje en el partido de River y Lanús
JAJAJAJA!!! Que picante este Laucha!!! https://t.co/3vHvZmb6dP	Se está hablando del jugador de Lanús, Lautaro Acosta
Algo así de GRANDE vivimos anoche en el Colon.No tengo palabras. En realidad tengo pero sobran.Solo Gracias !	Se está hablando de un evento en el Teatro Colón, por lo que pudo haberse confundido con el equipo Club Atlético Colón
El VAR les hizo 4 goles y es el nuevo goleador de la Copa Libertadores. Felicitaciones. https://t.co/EUnCbhiSnS	Se está hablando implícitamente del partido entre River y Lanús, haciendo referencia a la utilización del VAR en el partido

Tabla 3.8: Ejemplos de tweets que se confunden entre las etiquetas **Y** y **N**.

Luego nos quedamos con aquellos tweets en donde las etiquetas **Y** de cada criterio coinciden, generando un nuevo dataset con un total de 14986 tweet que representan el 37% del dataset recolectado, logrando con esto un filtrado más robusto.

Finalmente calculamos el Coeficiente kappa de Cohen, el cual mide la concordancia entre dos examinadores en sus correspondientes clasificaciones de **N** elementos en **C** categorías mutuamente excluyentes [Sim and Wright, 2005].

La ecuación para **k** es:

$$k = \frac{p_o - p_e}{1 - p_e}$$

Donde:

- p_o: porcentaje de acuerdo relativo entre los observadores (idéntico a la accuracy).
- p_e: probabilidad hipotética de acuerdo por azar, utilizando los datos observados para calcular las probabilidades de que cada observador clasifique aleatoriamente cada categoría.

El Coeficiente kappa de Cohen que se obtiene para las anotaciones hechas en el filtrado del dataset es de $k = 0.48$.

Según la literatura de Landis y Koch [Landis and Koch, 1977] podemos interpretar el valor k se la siguiente manera:

- $k < 0 \longrightarrow$ Sin acuerdo
- $0 \leq k \leq 0.20 \longrightarrow$ Acuerdo leve
- $0.21 \leq k \leq 0.40 \longrightarrow$ Acuerdo justo
- $0.41 \leq k \leq 0.60 \longrightarrow$ Acuerdo moderado
- $0.61 \leq k \leq 0.80 \longrightarrow$ Acuerdo sustancial
- $0.81 \leq k \leq 1 \longrightarrow$ Acuerdo casi perfecto

Por lo que el nivel de acuerdo obtenido entre las dos anotaciones es aceptable.

3.2.3 Etiquetado de Polaridad

Una vez que logramos filtrar aquellos tweets que hablan sobre fútbol argentino, procedimos a la etapa de etiquetado de polaridad. Como este procedimiento demandaba mucho tiempo, reducimos significativamente la cantidad de tweets del dataset, quedándonos solamente con 3198 tweets que representan aproximadamente el 22% de los tweets.

Para el etiquetado de los tweets participaron 3 personas, esta tarea tenía similitudes con la de filtrado, analizar cada tweet con el fin de determinar el sentimiento expresado por el mismo.

Para ello, la persona debe etiquetar el tweet las etiquetas **P** (Positivo), **N** (Negativo), **NEU** (Neutro) o **NONE** (No expresa sentimiento) [Nakov et al., 2016]. A continuación describiremos en detalle los criterios que se usaron para el uso de cada etiqueta [Mohammad, 2016] [Cambria et al., 2017]:

- **P**: Se está usando lenguaje positivo, como son las expresiones de apoyo, admiración o actitud positiva, en las que el twittero demuestra felicidad, relajación o indulgencia.
- **N**: Se está usando lenguaje negativo, como son las expresiones de crítica, fracasos o actitud negativa, en las que el twittero demuestra enojo, tristeza, violencia o emociones negativas.
- **NEU**: Se están usando el lenguaje positivo y negativo al mismo tiempo, o se está expresando un sentimiento que resulta difícil de determinar, como un posible sarcasmo.
- **NONE**: No se está usando ni lenguaje positivo ni lenguaje negativo, por lo que el twittero no indica estado emocional alguno, como son los tweets periodísticos que no expresan la opinión del twittero o las preguntas no retóricas.

En la sección 7.2 se puede encontrar el documento que usó cada persona como guía para el etiquetado de polaridad. En este documento también podemos encontrar la Tabla 7.2 con ejemplos de tweets anotados.

Al igual que en la etapa de filtrado, duplicamos el dataset para que sobre un mismo tweet haya dos anotaciones.

A continuación mostramos algunas estadísticas sobre el etiquetado de polaridad:

Etiqueta	**Anotación 1**	**Anotación 2**
P	1313	1261
N	1027	719
NEU	504	580
NONE	354	638
Total	3198	3198

Tabla 3.9: Cantidad de etiquetas de polaridad por anotación sobre el dataset.

Combinación	Cantidad	Porcentaje (%)
P & P	1050	32.83
N & N	557	17.42
NEU & NEU	185	5.78
NONE & NONE	202	6.32
P & N	88	2.75
P & NEU	200	6.25
P & NONE	186	5.82
N & NEU	328	10.26
N & NONE	216	6.75
NEU & NONE	186	5.82
Total	3198	100

Tabla 3.10: Combinaciones de etiquetas de polaridad puestas sobre el dataset.

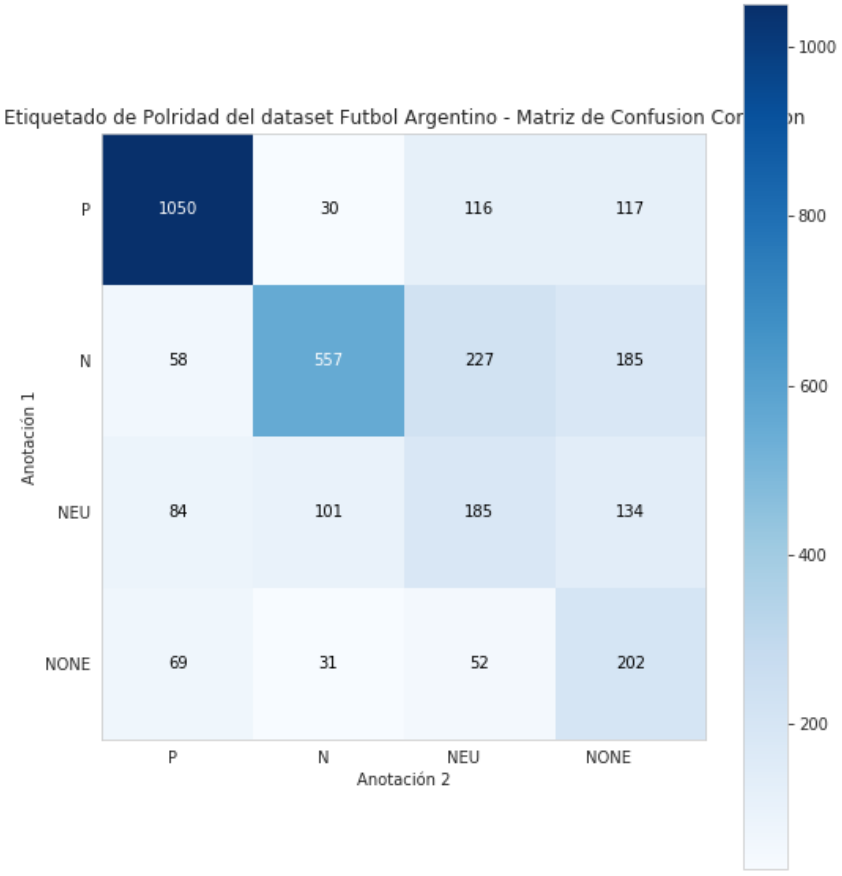


Figura 3.6: Matriz de confusión entre las anotaciones de polaridad hechas sobre el dataset.

En la Figura 3.6 podemos apreciar que entre las dos anotaciones hechas en el dataset, hubo una alta coherencia en las etiquetas **P** y **N**, por lo que se pudo distinguir bastante bien los tweets positivos y negativos. En las etiquetas **NEU** y **NONE**, hubo una coherencia muy baja, ya que al parecer es complicado distinguir las mismas del resto de las etiquetas.

Veamos algunos tweets de ejemplo:

Tweet	Anotación alternativa
Me cuentan por la cucaracha que un tal Pinola se fue a River para asegurarse el mundial y ganar la copa. Un cosa voy a decir: JAJAJAJAJAJAJA	N
Los hinchas de river que devolvieron las entradas. La mejor decisión de su vida. #Pechazo	N
River volvió a ser River.	P
Cero autocrítica de Gallardo. Hoy River se vió perjudicado y cuántas veces se vió beneficiado?? NO ME BANCO A LOS LLORONES. Felicitaciones a Lanús por este triunfo histórico. Las cosas como son	N
Por suerte Driussi se quedó y prefirió jugar la copa con River que irse a Rusia por la plata y el doping. Lo mismo que Alario. #Respect	P

Tabla 3.11: Ejemplos de tweets que se confunden entre la etiqueta **NEU** y el resto de las etiquetas.

Tweet	Anotación alternativa
#Video Lanús eliminó a River en la Libertadores pero Wilmar Roldán fue la figura por estar en el ojo de la tormenta. https://t.co/znmXOAsboP	P
Se terminó. #River perdió una serie increíble y quedó eliminado en la Copa Libertadores.	N
Vá el pronóstico para esta noche. Lanús 1 - River 1. #SriSri #ElGurúNoFalla	NEU
La hinchada de Rosario Central desplego una bandera gigante de unos 500 metros de largo y 40 de ancho previo al clásico. ¡IMPRESIONANTE! https://t.co/GwpoPYuUV4	P
Estudiantes no pudo con Boca y cayó 1 a 0 en el cierre del año.	N
Exclusivas declaraciones del dt de Lanus “FUE UN PARTIDO RARISIMO” https://t.co/dv7NR7GguK	NEU

Tabla 3.12: Ejemplos de tweets que se confunden entre la etiqueta **NONE** y el resto de las etiquetas.

Después de hacer realizar el etiquetado de polaridad, para el corpus final nos vamos a quedar solamente con aquellos tweets donde las etiquetas **P**, **N**, **NEU** y **NONE** coincidan.

A diferencia de cómo se realiza el etiquetado de los corpus de la SEPLN (forma semiautomática), el hecho de que anotadores humanos etiqueten todos los tweets y se duplique el dataset para finalmente quedarnos con aquellos en donde etiquetas las etiquetas coincidan, nos genera un corpus de mayor confianza.

Por lo que finalmente nos queda un corpus de 1994 tweets, distribuidos de la siguiente manera:

	# **Tweets**	%
P	1050	52.66
N	557	27.93
NEU	185	9.28
NONE	202	10.13
Total	1994	100

Tabla 3.13: Distribución de tweets del corpus Fútbol Argentino según su polaridad.

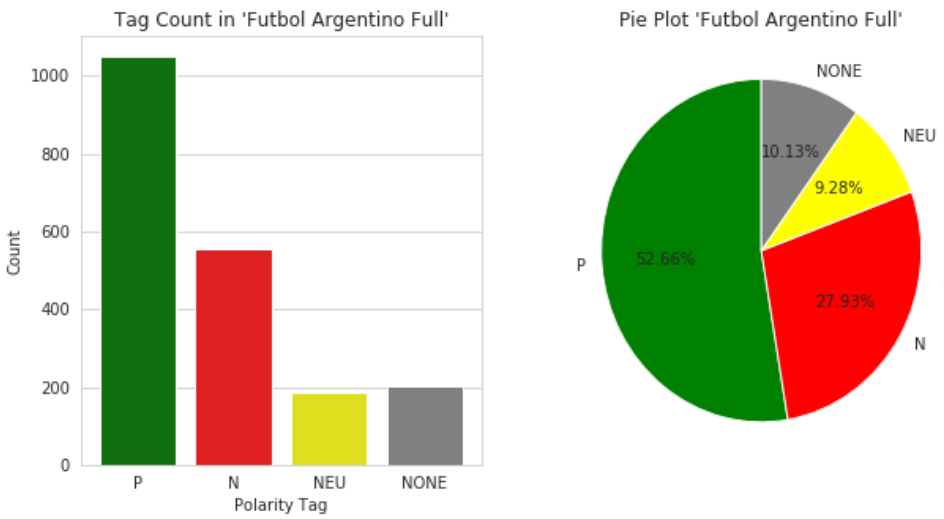


Figura 3.7: Gráficos de las distribuciones de tweets del corpus Fútbol Argentino según su polaridad.

Luego de esto, dividimos el corpus Fútbol Argentino en tres partes (*Train*, *Development* y *Test*) tal que tengan la misma distribución.

En la siguiente tabla podemos ver cómo están distribuidos los tweets en las distintas partes:

	Train		**Development**		**Test**	
	# Tweets	**%**	**# Tweets**	**%**	**# Tweets**	**%**
P	758	52.82	92	57.5	200	50.13
N	389	27.11	48	30	120	30.08
NEU	140	9.76	11	6.88	34	8.52
NONE	148	10.31	9	5.62	45	11.28
Total	1435	100	160	100	399	100

Tabla 3.14: Distribución de tweets del corpus Fútbol Argentino Train, Development y Test según su polaridad.

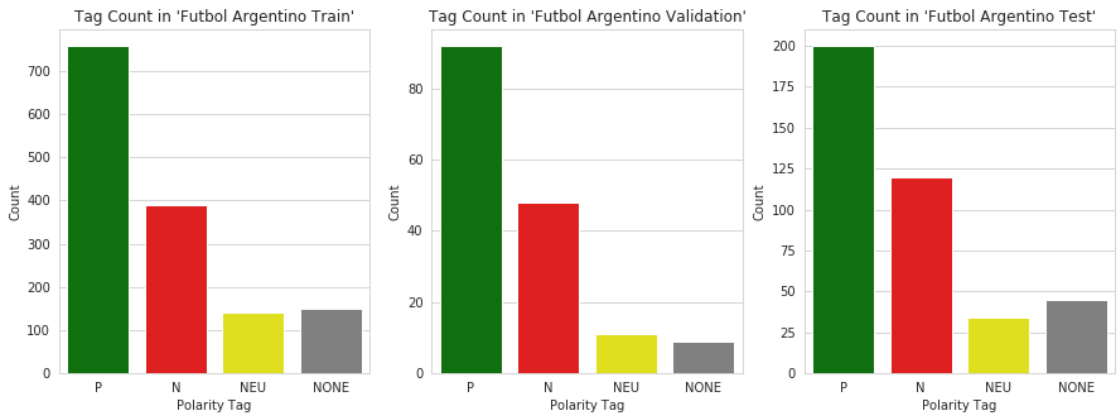


Figura 3.8: Gráficos de las distribuciones de tweets del corpus Fútbol Argentino Train, Development y Test según su polaridad.

Nuevamente calculamos el Coeficiente kappa de Cohen para las anotaciones hechas en el etiquetado de polaridad, obteniendo un $k = 0.47$, siendo este un nivel de acuerdo aceptable según la literatura de Landis y Koch [Landis and Koch, 1977].

Capítulo 4

Sistemas de Predicción

En este capítulo describiremos las distintas técnicas de preprocesamiento que se fueron aplicando a los tweets con el fin de normalizarlos, además de los distintos sistemas que se fueron desarrollando para la problemática que se describe en este trabajo.

4.1 Preprocesamiento: Limpieza y Normalización de tweets

La etapa de preprocesamiento es muy importante en el área de PLN, ya que los datos con lo que se trabaja son "ruidosos". Por lo que es en esta etapa donde el texto pasa por un proceso de limpieza y normalización.

Cada tweet que se encuentra en el corpus de entrada pasa por un proceso de tokenización en el cual el texto del tweet se separa en palabras y expresiones.

Para la realización de esta tarea usamos un tokenizador propio, el cual está basado en el tokenizer que nos provee una librería Python llamada NLTK[1] (Natural Language Toolkit).

El tokenizer provisto por NLTK se llama *TweetTokenizer* el cual es un tokenizador diseñado para Twitter, el mismo tiene la capacidad de distinguir ciertos elementos propios de la red social, como son menciones de usuarios, hashtags, emoticones/emojis y direcciones de correos electrónicos, entre otros.

Nuestro tokenizer está basado en *TweetTokenizer* con el agregado de nuestras reglas, que son las siguientes:

- **Reducción de la longitud de las palabras**:
  Reemplaza la secuencia de caracteres repetidos en de longitud 3 o mayor a una secuencia de 2 caracteres [Montejo-Ráez and Díaz-Galiano, 2016].
  Por ejemplo:
  - hooollaaaa $\longrightarrow$ hoollaa
  - agguanteeee $\longrightarrow$ agguantee
  - gooooollllllll $\longrightarrow$ gooll
- **Convertir todo el texto a minúsculas** [Montejo-Ráez and Díaz-Galiano, 2016].

[1] https://www.nltk.org/

– **Mapeo de menciones de usuarios a un placeholder**:
Reemplazamos del texto todas las menciones de usuario específicas de Twitter "*@user*" [Montejo-Ráez and Díaz-Galiano, 2016] por el placeholder TW_USERNAME.

– **Mapeo a un placeholder o eliminación de los "hashtags"**:
Reemplazamos o eliminamos del texto todas los hashtags "*#topic*", en caso de ser reemplazados, se hacen por el placeholder TW_HASHTAG [Cerón-Guzmán, 2016].

– **Removemos del texto las URLs y direcciones de correo electrónico** [Cerón-Guzmán, 2016].

– **Removemos del texto los números**:
La supresión de los números es porque no aportan nada para la detección de la polaridad [Navas-Loto and Rodríguex-Doncel, 2017].

– **Remover los signos de puntuación**.

– **Reemplazo de jergas usadas frecuentemente en Twitter por su forma correcta** [Navas-Loto and Rodríguex-Doncel, 2017].
Por ejemplo:

- 'q', 'k', 'qu', 'ke', 'qe' $\longrightarrow$ 'que'
- 'xq', 'pq', 'xque', 'porq' $\longrightarrow$ 'porque'
- 'tb', 'tmb' $\longrightarrow$ 'tambien'
- 'd' $\longrightarrow$ 'de'

– **Mapeo de todas las expresiones que denoten "gracia"**:
Reemplazo de todas las expresiones que denoten risas (e.g. "jajaja") por el placeholder LAUGHT_EXPRESSION. Esto conlleva un desafío ya que estas expresiones pueden adoptar diversas formas, es decir, "jajajaja", "jejeje", "hahaha", "lol" e incluso estas expresiones pueden contener errores ortográficos como "jajajjjajaaj", "llooooollll".

Para abordar la mayoría de estos casos hemos usado expresiones regulares para estandarizar las diferentes formas [Quirós et al., 2016].

– **Mapeo de palabras que se encuentren en lexicones por placeholders**:
El uso de lexicones de polaridad para el análisis de sentimiento a nivel de oraciones es indispensable, de esta forma podemos determinar la carga emocional de las palabras en cada tweet.

Los lexicones usados en este trabajo son:

- **iSOL** (Lexicon Mejorado de Opiniones en Español), el cual es una versión mejorada del recurso original llamado SOL, este contiene un total 8135 palabras clasificadas en dos categorías positiva y negativa [Molina-González et al., 2013].
- **ElhPolar** está compuesto por 5199 lemmas clasificados en positivos y negativos [Urizar and Roncal, 2013].
- **ML-SentiCon** está constituido por 11302 entradas o lemas en español, este lexicon contiene lemas polarizados con valores que van de -1.0 "negativo" hasta +1.0 "positivo", otra característica es que el lexicón está dividido en 8 capas donde cada capa está dividida en positivos y negativos además de que las capas están ordenadas, desde la primera hasta la octava, de manera que las capas posteriores contienen todos los lemas de las anteriores, y añaden algunos nuevos [Cruz et al., 2014].

Reemplazamos todas las palabras positivas y negativas en el tweet haciendo uso de los lexicones nombrados anteriormente, si la palabra en el tweet aparecía en las palabras positivas de alguno de los lexicones era cambiada por POS_WORD y si aparecía en las palabras negativas era cambiada por NEG_WORD.

– **Mapeo de emoticones por placeholders**:
Reemplazamos todos los emoticones que se encuentren en el tweet por los placeholders POS_EMOTI y NEG_EMOTI si son emoticones positivos y negativos respectivamente [Cerón-Guzmán, 2016].

A continuación mostramos una tabla para el mapeo utilizado:

Emoticones	**Polaridad**
:–), :), :D, :o), :], D:3, :c), :>, =], 8), =), :}, :^), :–D, 8–D, 8D, x–D, xD, X–D, XD, =–D, =D, =–3, =3, B^D, :'), :*, :–*, :^*, ;–), ;), *–), *), ;–], ;], ;D, ;^), >:P, :–P, :P, X–P, x–p, xp, XP, :–p, :p, =p, :–b, :b, <3, :o, o:, *.*, 0:3, :3, xd	Positivo
>:[, :–(, :(, :-c, :-<, :<, :-[, :[, :{, ;(, :-||, >:(, :'–(, :'(, D:<, D=, v.v, :S, :$, :__(, :C, D:, –_–, :/, :@	Negativo

Tabla 4.1: Lista de emoticones positivos y negativos [Quirós et al., 2016].

– **Mapeo de emojis por placeholders**:
Reemplazamos todas emojis que se encuentren en el tweet por los placeholder POS_EMOJI y NEG_EMOJI si son emojis positivos y negativos respectivamente, para ello utilizamos una lista con el código unicode de emojis positivos y negativos[2].

– **Eliminación de emojis que no aportan al sentimiento del texto**:

Aquellos emojis que no aportan ningún valor al sentimiento del tweet son removidos del texto, los mismos son aquellos usados para las banderas.[3]

– **Eliminación de Stopwords**:

Las Stopwords son palabras que son muy frecuentes pero que no aportan gran valor semántico, como articulos, pronombres, preposiciones [Montejo-Ráez and Díaz-Galiano, 2016]. Por ejemplo:

- de
- por
- con

[2]https://emojipedia.org/
[3]https://emojipedia.org/flags/

– **Aplicar Stemming al texto**:

Algunas veces las diferentes palabras pueden hacer referencia al mismo concepto, esto se debe a que este concepto puede ser representado por variantes morfológicas de una misma familia de palabras, esto nos permite relacionar aquellos tweets que contienen alguna de estas palabras mediante su raíz (o stem).

Este proceso es conocido como Stemming, resumiendo nos permite obtener la raíz de cada palabra, por ejemplo:

- maravilloso $\longrightarrow$ maravill
- maravilla $\longrightarrow$ maravill
- maravillarse $\longrightarrow$ maravill
- escribo $\longrightarrow$ escrib
- escribíamos $\longrightarrow$ escrib
- escribimos $\longrightarrow$ escrib

– **Aplicar Lemmatization al texto**:

La lematización, a diferencia de Stemming, reduce las palabras flexionadas adecuadamente asegurando que la palabra raíz pertenece al idioma. En la lematización, la palabra raíz se llama Lemma. Un lema es la forma canónica, la forma del diccionario o la forma de cita de un conjunto de palabras.

Para ello usamos el Lematizador TreeTagger [Schmid, 1995].

Algunos ejemplos:

- escribo $\longrightarrow$ escribir
- mirando $\longrightarrow$ mirar
- portátiles $\longrightarrow$ portátil

– **Reemplazo de todas las letras acentuadas por sus versiones sin acentuar** [Montejo-Ráez and Díaz-Galiano, 2016].

– **Remover las repeticiones consecutivas de placeholders**:

Remover las repeticiones de los placeholder TW_HASHTAG [Luque and Pérez, 2018] y TW_USERNAME.

Es importante aclarar que todas reglas listadas anteriormente no fueron usadas en su totalidad, si no que se fueron probando distintas combinaciones, quedándonos con las que mejores resultados se obtenían.

4.2 Representaciones "Bag of Words"

El pipeline diseñado en este trabajo para el armado de los sistemas consiste en obtener cada tweet y tokenizarlo, es decir, transformarlo en una lista de tokens usando el tokenizador propio que describimos en la sección 4.1.

Generando así el llamado vocabulario del corpus, que consiste de las palabras utilizadas, distintas y únicas.

Luego de esto aplicamos el enfoque de BoW (Bag-of-Words), en el que la representación de los tweet es un vector en un espacio euclídeo, generando un matriz rala (sparse), donde cada columna es una feature. Donde estas features son principalmente palabras del vocabulario del corpus, generados a partir de la tokenización [contributors, 2020a].

En el mencionado BoW, podemos considerar cada palabra del vocabulario como una columna de la matriz, o bien obtener otro tipo de representaciones como secuencias de palabras, comúnmente llamadas *n-gramas*, las cuales pueden ser de una palabra (unigramas), dos palabras (bigramas), tres palabras (trigramas) y así sucesivamente. Por lo que básicamente cuenta la frecuencia de las palabras.

Veamos un ejemplo, con los siguientes dos tweets usando unigramas:

- **Tweet 1**: aguante river loco
- **Tweet 2**: vamo vamo river plate

Formándose la siguiente matriz:

Tweet	**aguante**	**loco**	**plate**	**river**	**vamo**
aguante river loco	1	1	0	1	0
vamo vamo river plate	0	0	1	1	2

También usamos *TF-IDF* (*Term frequency - Inverse document frequency*) que expresa cuán relevante es una palabra para un documento en una colección. El valor tf-idf aumenta proporcionalmente al número de veces que una palabra aparece en el documento, pero es compensada por la frecuencia de la palabra en la colección de documentos, lo que permite manejar el hecho de que algunas palabras son generalmente más comunes que otras [Rajaraman and Ullman, 2011].

TF-IDF se calcula de la siguiente manera:

$$tfidf(t, d, D) = tf(t, d) * idf(t, D)$$

A $tf(t, d)$ se lo denomina **frecuencia de término** y a $idf(t, D)$ **frecuencia inversa de documento**. Donde:

- t: Término
- d: Documento
- D: Colección de documentos

Siguiendo el ejemplo anterior, para *TF-IDF* se forma la siguiente matriz:

Tweet	**aguante**	**loco**	**plate**	**river**	**vamo**
aguante river loco	0.6316672	0.6316672	0	0.44943642	0
vamo vamo river plate	0	0	0.4261596	0.30321606	0.8523192

Para el cálculo de matriz anterior se usó la librería de Python scikit-learn [Pedregosa et al., 2011].

4.3 Embeddings

Los *Word Embeddings* son representaciones de vectoriales de baja dimensión, en donde donde las palabras o frases del vocabulario se asignan a vectores de números reales [Mikolov et al., 2013]. Este vector guarda información semántica, lo que permite que pueda ser asociado o disociado a otros vectores (palabras) según distintos contextos gramaticales, demostrando aumentar la performance en muchas tareas de PNL.

Además de que los word embeddings pueden ser entrenados de manera no supervisada utilizando grandes cantidades de texto sin formato.

Cuando los word embeddings son utilizados en tareas supervisadas, proporcionan información sólida para palabras que son raras o invisibles en los datos de entrenamiento. Esto es particularmente es muy útil cuando los datos de entrenamiento son escasos, como en esta competencia.

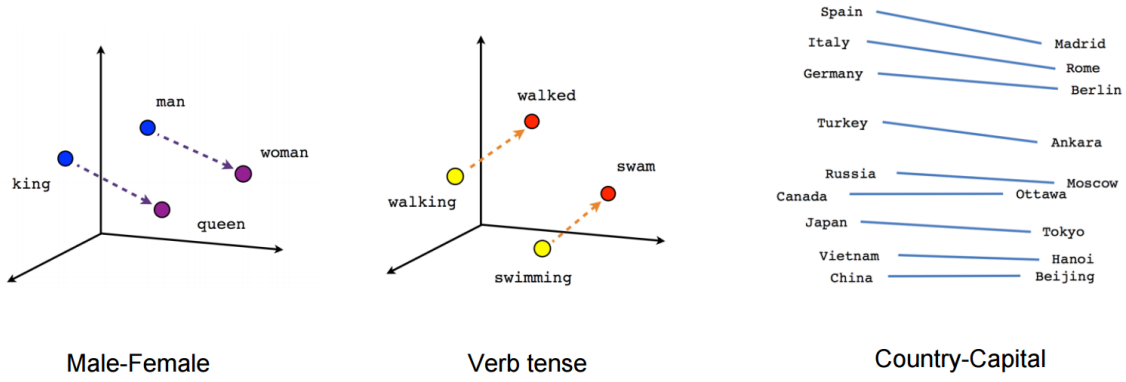


Figura 4.1: Ejemplo de Word Embeddings.

Para este trabajo, utilizamos la librería fastText[4] [Bojanowski et al., 2016], que permite aprender representaciones de texto y clasificadores de texto. Además fastText nos provee vectores de palabras pre-entrenados con texto de Common Crawl[5] y Wikipedia[6], pero nosotros decidimos no usarlos ya que quisimos que los vectores de palabras estuvieran entrenados dentro del contexto de Twitter, por lo que entrenamos nuestros word embeddings usando un corpus de aproximadamente 66 millones de tweets provisto por la UBA (Universidad de Buenos Aires), también usamos el corpus SBWCE (Spanish Billion Word Corpus and Embeddings) [Cardellino, 2019].

Para la entrenamiento de los word embeddings fuimos por dos caminos, el primero realizando un preprocesamiento básico en donde solamente separamos el texto en tokens y el segundo usando un preprocesamiento más complejo utilizando algunas de las técnicas nombradas en la sección 4.1. Además de esto, fastText nos permite configurar ciertos parámetros al momento de entrenar el vector de palabras, como son la dimensión del vector y el tamaño de la palabra (n-grama).

Hay varias formas de utilizar los word embeddings para la problemática planteada en este trabajo, los enfoques van desde el promedio simple de vectores para cada palabra en el tweet,

[4]https://fasttext.cc/
[5]https://commoncrawl.org/
[6]https://www.wikipedia.org/

hasta el uso de arquitecturas más complejas como CNN o RNN. Nosotros decidimos utilizar el promedio simple para calcular el embedding de un solo tweet.

4.4 Data Augmentation

Una técnica muy utilizada en el área de Computer Vision es la de Data Augmentation, la cual genera nuevos datos de manera sintética a partir de los datos reales, introduciendo variabilidad en los mismos. La manera en que se generan nuevos datos son mediante rotaciones, flips y shifts randoms, flips random.

Esta técnica es muy usada para la regularización en modelos de Deep Learning tratando de reducir el overfitting, siendo exitoso en la mayoría de los casos.

En contraparte con el área de PLN, donde no hay muchas variantes de Data Augmentation, una técnica común es la de reemplazar las palabras por sinónimos usando algún diccionario.

En este trabajo, como los datasets con los que trabajamos son pequeños, lo que hicimos para poder aplicar la técnica de Data Augmentation, fue traducir el texto del tweet a otros idiomas y luego aplicar el proceso inverso, traduciendolos de vuelta al idioma original.

Esto genera tweets con variantes léxicas y sintácticas, pero manteniendo la esencia o el significado del mismo [Luque and Pérez, 2018].

La herramienta que usamos para generar estos nuevos tweets fue una libreria de Python llamada **googletrans**[7] basada en la API de Google Translate.

Los idiomas que usamos para traducir los tweets fueron:

- Inglés
- Francés
- Portugués
- Italiano
- Alemán
- Árabe

La siguiente tabla muestra un ejemplo del resultado de la técnica utilizada:

Tweet Original	**Tweet Sintético**
Es muy raro el sentimiento que tengo ahora, aunque en fin... Qué más dará	– Es muy extraño el sentimiento que tengo ahora, aunque de todos modos ... ¿Qué más dará? – El sentimiento que tengo ahora es muy extraño, incluso si ... ¿Qué más darás? – Muy extraño sentimiento que tengo ahora, aunque de todos modos ... ¿qué va a dar?

Tabla 4.2: Ejemplo de Data Augmentation sobre un tweet.

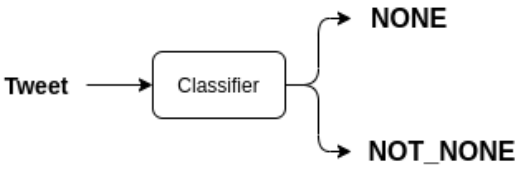
[7] https://pypi.org/project/googletrans/

4.5 Clasificadores en Cascada

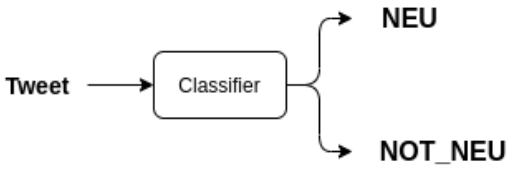
Este enfoque trata de atacar el problema usando la técnica de stacking de clasificadores o clasificadores en cascada.

Para ello se entrenaron distintos clasificadores "especializados" en clasificar los tweet en una cierto conjunto de etiquetas.

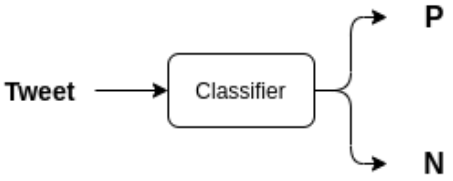
Particularmente, para este trabajo se entrenaron los siguiente clasificadores:



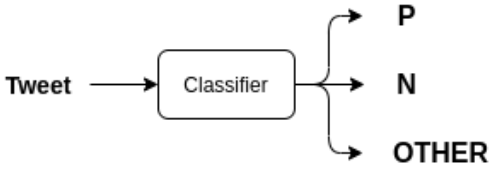
(a) Clasificador NONE, NOT_NONE



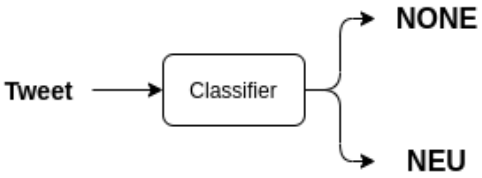
(b) Clasificador NEU, NOT_NEU



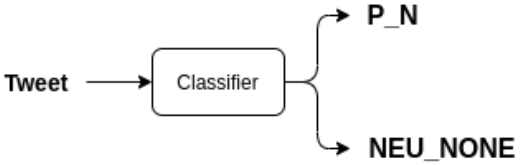
(c) Clasificador P, N



(d) Clasificador P, N, OTHER



(e) Clasificador NONE, NEU



(f) Clasificador P_N, NEU_NONE

Figura 4.2: Clasificadores "especializados".

Para luego combinarlos formando distintas arquitecturas, particularmente se diseñaron 3 arquitecturas.

A continuación podemos ver los esquemas de las arquitecturas diseñadas:

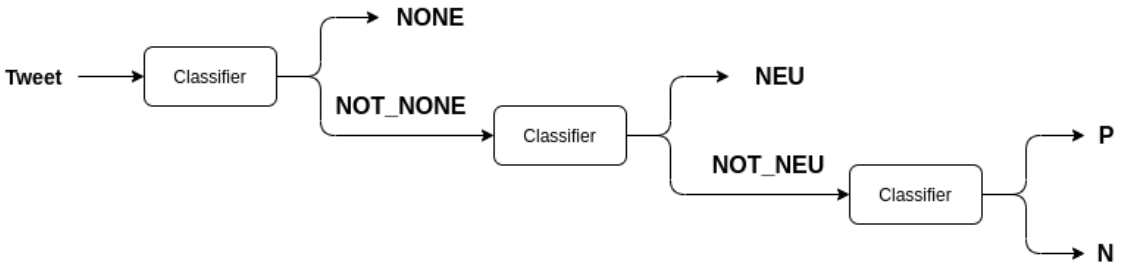


Figura 4.3: Clasificadores en Cascada - Arquitectura 1.

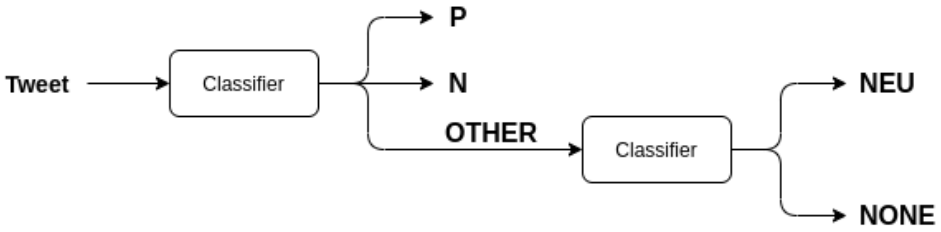


Figura 4.4: Clasificadores en Cascada - Arquitectura 2.

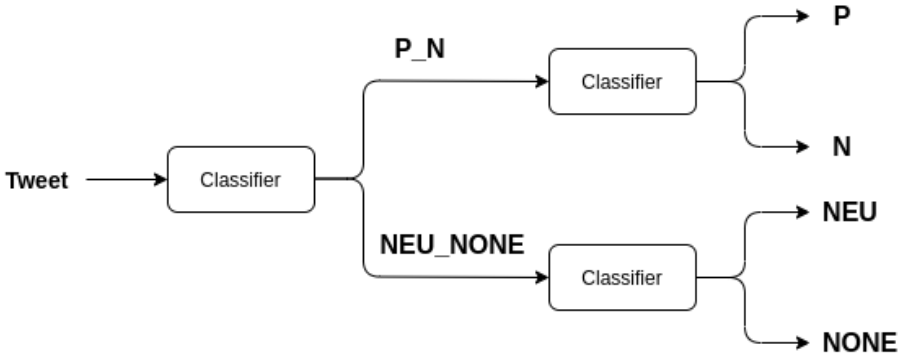


Figura 4.5: Clasificadores en Cascada - Arquitectura 3.

Capítulo 5

Experimentos y Resultados

En este capítulo describiremos los resultados obtenidos sobre los distintos corpus, usando los sistemas de clasificación propuestos en el capítulo 4, junto con las técnicas de preprocesamiento nombras en la sección 4.1.

En algunos experimentos usamos solamente los corpus *InterTASS* y *Fútbol Argentino*, y en otros usamos, además de los corpus antes mencionados, el corpus *GeneralTASS* como aditivo para entrenar.

Es importante destacar que la *TASS* no permite usar un corpus anotado ajeno a los proporcionados por ellos, aun así permiten usar recursos lingüísticos como lexicones y word embeddings. También es importante nombrar que ya que estamos en el contexto de la *TASS*, las métricas que se usan para evaluar los sistemas son la *Accuracy*, *Macro-Precision*, *Macro-Recall* y *Macro-F1*. Pero los sistemas se clasifican usando la *Macro-F1* y *Accuracy*.

Si bien las etiquetas de los conjuntos de *Test* de los corpus *InterTASS* no se encuentran disponibles, la *TASS* nos permite evaluar nuestros sistemas la siguiente pagina web: `http://tass.sepln.org/2018/task-1/private/evaluation/evaluate.php`

En donde los resultados deben enviarse en un archivo de texto con el siguiente formato:

```
tweet_id \t polarity
```

5.1 Clasificador Baseline

Un clasificador baseline es un sistema "básico" que se usa para poder definir una cota inferior de performance para las métricas.

Nuestro modelo baseline es un clasificador *dummy* que siempre predice la etiqueta más frecuente que haya visto en el dataset de entrenamiento.

	Development				**Test**			
Corpus	**M-Prec**	**M-Rec**	**M-F1**	**Acc**	**M-Prec**	**M-Rec**	**M-F1**	**Acc**
InterTASS Spain	0.108	0.250	0.151	0.433	0.101	0.250	0.144	0.404
InterTASS Spain + GeneralTASS	0.077	0.250	0.118	0.308	0.085	0.250	0.126	0.338
InterTASS Costa Rica	0.092	0.250	0.134	0.367	0.100	0.250	0.142	0.398
InterTASS Costa Rica + GeneralTASS	0.077	0.250	0.118	0.310	0.072	0.250	0.112	0.287
InterTASS Peru	0.119	0.250	0.161	0.476	0.028	0.250	0.050	0.111
InterTASS Peru + GeneralTASS	0.048	0.250	0.080	0.190	0.075	0.250	0.116	0.301
Fútbol Argentino	0.144	0.250	0.183	0.575	0.125	0.250	0.167	0.501
Fútbol Argentino + GeneralTASS	0.144	0.250	0.183	0.575	0.125	0.250	0.167	0.501

Tabla 5.1: Resultados del modelo baseline.

A continuación veremos métricas más detalladas de los sistemas entrenados con los corpus:

- InterTASS Spain
- Fútbol Argentino

respectivamente.

InterTASS Spain

		Predicted				
		P	**N**	**NEU**	**NONE**	**# Tweets True**
Actual	**P**	0	156	0	0	156
	N	0	219	0	0	219
	NEU	0	69	0	0	69
	NONE	0	62	0	0	62
	# Tweets Pred	0	506	0	0	506

Tabla 5.2: Matriz de confusión del corpus InterTASS Spain Development.

Polaridad	**Hits**	**# Tweets True**	**# Tweets Pred**	**Prec**	**Rec**	**F1**
P	0	156	0	0.000 (0/0)	0.000 (0/156)	0.000
N	219	219	506	0.433 (219/506)	1.000 (219/219)	0.604
NEU	0	69	0	0.000 (0/0)	0.000 (0/69)	0.000
NONE	0	62	0	0.000 (0/0)	0.000 (0/62)	0.000
	219	506	506			

Tabla 5.3: Metricas de cada etiqueta sobre el corpus InterTASS Spain Development.

Fútbol Argentino

		Predicted				
		P	**N**	**NEU**	**NONE**	**# Tweets True**
Actual	**P**	92	0	0	0	92
	N	48	0	0	0	48
	NEU	11	0	0	0	11
	NONE	9	0	0	0	9
	# Tweets Pred	160	0	0	0	0

Tabla 5.4: Matriz de confusión del corpus Fútbol Argentino Development.

Polaridad	**Hits**	**# Tweets True**	**# Tweets Pred**	**Prec**	**Rec**	**F1**
P	92	92	160	0.575 (92/160)	1.000 (92/92)	0.730
N	0	48	0	0.000 (0/0)	0.000 (0/48)	0.000
NEU	0	11	0	0.000 (0/0)	0.000 (0/11)	0.000
NONE	0	9	0	0.000 (0/0)	0.000 (0/9)	0.000
	92	160	160			

Tabla 5.5: Metricas de cada etiqueta sobre el corpus Fútbol Argentino Development.

5.2 Representaciones "Bag of Words"

Experimentamos con 2 vectorizadores de scikit-learn [Pedregosa et al., 2011]

- *CountVectorizer*
- *TfidfVectorizer*

y 2 clasificadores lineales:

- Support Vector Machine con un Kernel Lineal
- Logistic Regression

Realizando un tuneo de hiperparametros, aplicando GridSearch y Cross Validation, obteniendo los siguientes resultados:

	Development				**Test**			
Corpus	**M-Prec**	**M-Rec**	**M-F1**	**Acc**	**M-Prec**	**M-Rec**	**M-F1**	**Acc**
InterTASS Spain	0.452	0.363	0.403	0.538	0.605	0.385	0.470	0.565
InterTASS Spain + GeneralTASS	0.602	0.398	0.479	0.545	0.536	0.396	0.456	0.554
InterTASS Costa Rica	0.533	0.458	0.493	0.557	0.420	0.388	0.403	0.491
InterTASS Costa Rica + GeneralTASS	0.492	0.416	0.451	0.503	0.445	0.409	0.426	0.498
InterTASS Peru	0.382	0.381	0.382	0.506	0.405	0.389	0.397	0.345
InterTASS Peru + GeneralTASS	0.423	0.432	0.427	0.388	0.400	0.393	0.397	0.410
Fútbol Argentino	0.638	0.452	0.529	0.788	0.727	0.440	0.548	0.704
Fútbol Argentino + GeneralTASS	0.626	0.451	0.525	0.769	0.811	0.465	0.591	0.707

Tabla 5.6: Resultados sobre cada corpus usando BoW.

A continuación veremos las técnicas de preprocesamiento y métricas más detalladas de los sistemas entrenados con los corpus:

- InterTASS Spain
- Fútbol Argentino

InterTASS Spain

De las técnicas de preprocesamiento presentadas en la sección 4.1, hicimos la siguiente selección para el corpus *InterTASS Spain*:

1. Reduce la secuencia de caracteres repetidos en una palabra, a 2 caracteres.
2. Convierte todo el texto del tweet a minúscula.
3. Reemplaza todas las menciones de usuario "*@user*" y los hashtags "*#topic*" por los placeholders TW_USERNAME y TW_HASHTAG respectivamente.
4. Remueve del texto todas las URL, direcciones de correo electrónico, números, signos de puntuación y emojis que no representen algún sentimiento.
5. Mapea todas las jergas en el texto, por su versión correcta.
6. Mapea las palabras del texto que estén presentes en los lexicones mencionados en la sección 4.1.
7. Mapea todos los emojis presentes en el texto por los placeholders POS_EMOJI y NEG_EMOJI, según el sentimiento que presenten.
8. Se le aplica la Lematización a todas las palabras del texto, para obtener el lema de las mismas.
9. Finalmente, se reduce las repeticiones de los placeholders TW_HASHTAG y TW_USERNAME a solamente una repetición.

También es importante nombrar que se usa unigramas como representación de secuencias de palabras.

		Predicted				
		P	**N**	**NEU**	**NONE**	**# Tweets True**
Actual	**P**	97	59	0	0	156
	N	43	172	2	2	219
	NEU	24	43	1	1	69
	NONE	21	39	0	2	62
	# Tweets Pred	185	313	3	5	506

Tabla 5.7: Matriz de confusión del corpus InterTASS Spain Development.

Polaridad	**Hits**	**# Tweets True**	**# Tweets Pred**	**Prec**	**Rec**	**F1**
P	97	156	185	0.524 (97/185)	0.622 (97/156)	0.569
N	172	219	313	0.550 (172/313)	0.785 (172/219)	0.647
NEU	1	69	3	0.333 (1/3)	0.014 (1/69)	0.028
NONE	2	62	5	0.400 (2/5)	0.032 (2/62)	0.060
	272	506	506			

Tabla 5.8: Metricas de cada etiqueta sobre el corpus InterTASS Spain Development.

Fútbol Argentino

De las técnicas de preprocesamiento presentadas en la sección 4.1, hicimos la siguiente selección para el corpus *Fútbol Argentino*:

1. Reduce la secuencia de caracteres repetidos en una palabra, a 2 caracteres.
2. Convierte todo el texto del tweet a minúscula.
3. Reemplaza todas las menciones de usuario "*@user*" y los hashtags "*#topic*" por los placeholders TW_USERNAME y TW_HASHTAG respectivamente.
4. Remueve del texto todas las URL, direcciones de correo electrónico, números, signos de puntuación y emojis que no representen algún sentimiento.
5. Mapea todas las jergas en el texto, por su versión correcta.
6. Se le aplica la Lematización a todas las palabras del texto, para obtener el lema de las mismas.
7. Remueve las stopwords presentes en el texto.
8. Finalmente, se reduce las repeticiones de los placeholders TW_HASHTAG y TW_USERNAME a solamente una repetición.

También es importante nombrar que se usa unigramas como representación de secuencias de palabras.

		Predicted				
		P	**N**	**NEU**	**NONE**	**# Tweets True**
Actual	**P**	89	3	0	0	92
	N	12	36	0	0	48
	NEU	5	5	1	0	11
	NONE	5	4	0	0	9
	# Tweets Pred	111	48	1	0	160

Tabla 5.9: Matriz de confusión del corpus Fútbol Argentino Development.

Polaridad	**Hits**	**# Tweets True**	**# Tweets Pred**	**Prec**	**Rec**	**F1**
P	89	92	111	0.802 (89/111)	0.967 (89/92)	0.877
N	36	48	48	0.750 (36/48)	0.750 (36/48)	0.750
NEU	1	11	1	1.000 (1/1)	0.091 (1/11)	0.167
NONE	0	9	0	0.000 (0/0)	0.000 (0/9)	0.000
	126	160	160			

Tabla 5.10: Metricas de cada etiqueta sobre el corpus Fútbol Argentino Development.

5.3 Embeddings

Para este experimento usamos como base los Sistemas BoW de la sección 5.2 y le agregamos el uso de Word Embeddings. En algunos de los resultados no vemos mejoras con respecto al uso de los sistemas sin Word Embeddings.

En la siguiente tabla podemos ver los resultados, las celdas en verde son aquellas en la cuales las métricas mejoraron y las celdas en rojo son aquellas en las cuales las métricas empeoraron.

	Development				Test			
Corpus	**M-Prec**	**M-Rec**	**M-F1**	**Acc**	**M-Prec**	**M-Rec**	**M-F1**	**Acc**
InterTASS Spain	0.477	0.379	0.422	0.549	0.563	0.390	0.461	0.569
InterTASS Spain + GeneralTASS	0.605	0.402	0.483	0.545	0.540	0.401	0.460	0.560
InterTASS Costa Rica	0.526	0.461	0.491	0.557	0.430	0.395	0.412	0.504
InterTASS Costa Rica + GeneralTASS	0.543	0.426	0.477	0.517	0.440	0.414	0.426	0.508
InterTASS Peru	0.423	0.410	0.416	0.520	0.439	0.405	0.421	0.368
InterTASS Peru + GeneralTASS	0.425	0.433	0.429	0.394	0.394	0.388	0.391	0.407
Fútbol Argentino	0.651	0.470	0.546	0.812	0.768	0.466	0.580	0.724
Fútbol Argentino + GeneralTASS	0.652	0.475	0.550	0.806	0.750	0.467	0.575	0.719

Tabla 5.11: Resultados sobre cada corpus usando Embeddings.

A continuación veremos métricas más detalladas de los sistemas entrenados con los corpus:

- InterTASS Spain + GeneralTASS
- Fútbol Argentino

InterTASS Spain + GeneralTASS

		Predicted				
		P	**N**	**NEU**	**NONE**	**# Tweets True**
Actual	**P**	120	28	0	8	156
	N	65	145	0	9	219
	NEU	26	35	1	7	69
	NONE	25	27	0	10	62
	# Tweets Pred	236	235	1	34	506

Tabla 5.12: Matriz de confusión del corpus InterTASS Spain Development.

Polaridad	**Hits**	**# Tweets True**	**# Tweets Pred**	**Prec**	**Rec**	**F1**
P	120	156	236	0.508 (120/236)	0.769 (120/156)	0.612
N	145	219	235	0.617 (145/235)	0.662 (145/219)	0.639
NEU	1	69	1	1.000 (1/1)	0.014 (1/69)	0.029
NONE	10	62	34	0.294 (10/34)	0.161 (10/62)	0.208
	276	506	506			

Tabla 5.13: Metricas de cada etiqueta sobre el corpus InterTASS Spain Development.

Fútbol Argentino

		Predicted				
		P	**N**	**NEU**	**NONE**	**# Tweets True**
Actual	**P**	90	2	0	0	92
	N	9	39	0	0	48
	NEU	6	4	1	0	11
	NONE	4	5	0	0	9
	# Tweets Pred	109	50	1	0	160

Tabla 5.14: Matriz de confusión del corpus Fútbol Argentino Development.

Polaridad	**Hits**	**# Tweets True**	**# Tweets Pred**	**Prec**	**Rec**	**F1**
P	90	92	109	0.826 (90/109)	0.978 (90/92)	0.896
N	39	48	50	0.780 (39/50)	0.812 (39/48)	0.796
NEU	1	11	1	1.000 (1/1)	0.091 (1/11)	0.167
NONE	0	9	0	0.000 (0/0)	0.000 (0/9)	0.000
	130	160	160			

Tabla 5.15: Metricas de cada etiqueta sobre el corpus Fútbol Argentino Development.

5.4 Data Augmentation

Para este experimento usamos la técnica de Data Augmentation sobre corpus de la *InterTASS*, *GeneralTASS* y *Fútbol Argentino*, para luego usar los Sistemas *BoW* de la Sección 5.2. Solo se vieron mejoras en algunos de los corpus.

En la siguiente tabla podemos ver los resultados, las celdas en verde son aquellas en la cuales las métricas mejoraron y las celdas en rojo son aquellas en las cuales las métricas empeoraron.

	Development				**Test**			
Corpus	**M-Prec**	**M-Rec**	**M-F1**	**Acc**	**M-Prec**	**M-Rec**	**M-F1**	**Acc**
InterTASS Spain	0.447	0.391	0.417	0.551	0.436	0.394	0.414	0.562
InterTASS Spain + GeneralTASS	0.620	0.404	0.489	0.549	0.493	0.396	0.439	0.557
InterTASS Costa Rica	0.531	0.473	0.501	0.563	0.409	0.390	0.399	0.492
InterTASS Costa Rica + GeneralTASS	0.524	0.415	0.463	0.500	0.445	0.414	0.429	0.508
InterTASS Peru	0.396	0.401	0.398	0.440	0.409	0.406	0.407	0.390
InterTASS Peru + GeneralTASS	0.408	0.419	0.414	0.386	0.395	0.389	0.392	0.407
Fútbol Argentino	0.641	0.457	0.534	0.794	0.736	0.469	0.573	0.724
Fútbol Argentino + GeneralTASS	0.761	0.502	0.605	0.781	0.724	0.472	0.572	0.712

Tabla 5.16: Resultados sobre cada corpus usando Data Augmentation.

A continuación veremos métricas más detalladas de los sistemas entrenados con los corpus:

- InterTASS Spain + GeneralTASS
- Fútbol Argentino

InterTASS Spain + GeneralTASS

		Predicted				
		P	**N**	**NEU**	**NONE**	**# Tweets True**
Actual	**P**	114	38	0	4	156
	N	59	152	0	8	219
	NEU	27	34	1	7	69
	NONE	23	28	0	11	62
	# Tweets Pred	223	252	1	30	506

Tabla 5.17: Matriz de confusión del corpus InterTASS Spain Development.

Polaridad	**Hits**	**# Tweets True**	**# Tweets Pred**	**Prec**	**Rec**	**F1**
P	114	156	223	0.511 (114/223)	0.731 (114/156)	0.602
N	152	219	252	0.603 (152/252)	0.694 (152/219)	0.645
NEU	1	69	1	1.000 (1/1)	0.014 (1/69)	0.029
NONE	11	62	30	0.367 (11/30)	0.177 (11/62)	0.239
	278	506	506			

Tabla 5.18: Metricas de cada etiqueta sobre el corpus InterTASS Spain Development.

Fútbol Argentino

		Predicted				
		P	**N**	**NEU**	**NONE**	**# Tweets True**
Actual	**P**	89	3	0	0	92
	N	11	37	0	0	48
	NEU	5	5	1	0	11
	NONE	5	4	0	0	9
	# Tweets Pred	110	49	1	0	160

Tabla 5.19: Matriz de confusión del corpus Fútbol Argentino Development.

Polaridad	**Hits**	**# Tweets True**	**# Tweets Pred**	**Prec**	**Rec**	**F1**
P	89	92	110	0.809 (89/110)	0.967 (89/92)	0.881
N	37	48	49	0.755 (37/49)	0.771 (37/48)	0.763
NEU	1	11	1	1.000 (1/1)	0.091 (1/11)	0.167
NONE	0	9	0	0.000 (0/0)	0.000 (0/9)	0.000
	127	160	160			

Tabla 5.20: Metricas de cada etiqueta sobre el corpus Fútbol Argentino Development.

5.5 Clasificadores en Cascada

Para este último experimento, probamos las distintas arquitecturas de clasificadores en cascada.

5.5.1 Arquitectura 1

	Development				Test			
Corpus	**M-Prec**	**M-Rec**	**M-F1**	**Acc**	**M-Prec**	**M-Rec**	**M-F1**	**Acc**
InterTASS Spain	0.602	0.368	0.457	0.542	0.608	0.381	0.468	0.563
InterTASS Spain + GeneralTASS	0.404	0.395	0.400	0.536	0.428	0.406	0.417	0.551
InterTASS Costa Rica	0.617	0.433	0.509	0.563	0.351	0.368	0.359	0.506
InterTASS Costa Rica + GeneralTASS	0.593	0.417	0.490	0.533	0.449	0.388	0.416	0.504
InterTASS Peru	0.410	0.405	0.408	0.354	0.389	0.395	0.392	0.477
InterTASS Peru + GeneralTASS	0.575	0.398	0.471	0.324	0.312	0.378	0.342	0.464
Fútbol Argentino	0.740	0.532	0.619	0.787	0.598	0.520	0.556	0.727
Fútbol Argentino + GeneralTASS	0.629	0.598	0.613	0.794	0.518	0.499	0.508	0.709

Tabla 5.21: Resultados sobre cada corpus usando la arquitectura 1 que se vio en la Figura 4.3 de Clasificadores en Cascada.

A continuación veremos métricas más detalladas de los sistemas entrenados con los corpus:

- InterTASS Spain
- Fútbol Argentino

InterTASS Spain

		Predicted				
		P	**N**	**NEU**	**NONE**	**# Tweets True**
Actual	**P**	92	62	2	0	156
	N	35	177	7	0	219
	NEU	21	44	4	0	69
	NONE	17	44	0	1	62
	# Tweets Pred	165	327	13	1	506

Tabla 5.22: Matriz de confusión del corpus InterTASS Spain Development.

Polaridad	**Hits**	**# Tweets True**	**# Tweets Pred**	**Prec**	**Rec**	**F1**
P	92	156	165	0.558 (92/165)	0.590 (92/156)	0.573
N	177	219	327	0.541 (177/327)	0.808 (177/219)	0.648
NEU	4	69	13	0.308 (4/13)	0.058 (4/69)	0.098
NONE	1	62	1	1.000 (1/1)	0.016 (1/62)	0.032
	274	506	506			

Tabla 5.23: Metricas de cada etiqueta sobre el corpus InterTASS Spain Development.

Fútbol Argentino

		Predicted				
		P	**N**	**NEU**	**NONE**	**# Tweets True**
Actual	**P**	84	6	0	2	92
	N	8	38	0	2	48
	NEU	4	5	1	1	11
	NONE	4	2	0	3	9
	# Tweets Pred	100	51	1	8	106

Tabla 5.24: Matriz de confusión del corpus Fútbol Argentino Development.

Polaridad	**Hits**	**# Tweets True**	**# Tweets Pred**	**Prec**	**Rec**	**F1**
P	84	92	100	0.840 (84/100)	0.913 (84/92)	0.875
N	38	48	51	0.745 (38/51)	0.792 (38/48)	0.768
NEU	1	11	1	1.000 (1/1)	0.091 (1/11)	0.167
NONE	3	9	8	0.375 (3/8)	0.333 (3/9)	0.353
	126	160	160			

Tabla 5.25: Metricas de cada etiqueta sobre el corpus Fútbol Argentino Development.

5.5.2 Arquitectura 2

	Development				Test			
Corpus	**M-Prec**	**M-Rec**	**M-F1**	**Acc**	**M-Prec**	**M-Rec**	**M-F1**	**Acc**
InterTASS Spain	0.425	0.393	0.408	0.536	0.436	0.410	0.422	0.558
InterTASS Spain + GeneralTASS	0.400	0.400	0.400	0.516	0.441	0.421	0.431	0.557
InterTASS Costa Rica	0.474	0.459	0.466	0.540	0.397	0.389	0.393	0.474
InterTASS Costa Rica + GeneralTASS	0.457	0.457	0.457	0.533	0.398	0.406	0.402	0.468
InterTASS Perú	0.441	0.380	0.408	0.492	0.459	0.374	0.412	0.301
InterTASS Perú + GeneralTASS	0.415	0.444	0.429	0.434	0.401	0.400	0.400	0.419
Fútbol Argentino	0.534	0.534	0.534	0.762	0.591	0.541	0.565	0.717
Fútbol Argentino + GeneralTASS	0.609	0.570	0.589	0.806	0.505	0.479	0.491	0.697

Tabla 5.26: Resultados sobre cada corpus usando la arquitectura 2 que se vio en la Figura 4.4 de Clasificadores en Cascada.

A continuación veremos métricas más detalladas de los sistemas entrenados con los corpus:

- InterTASS Spain + GeneralTASS
- Fútbol Argentino

InterTASS Spain + GeneralTASS

		Predicted				
		P	**N**	**NEU**	**NONE**	**# Tweets True**
Actual	**P**	113	30	2	11	156
	N	61	130	8	20	219
	NEU	22	34	4	9	69
	NONE	20	24	4	14	62
	# Tweets Pred	216	218	18	54	506

Tabla 5.27: Matriz de confusión del corpus InterTASS Spain Development.

Polaridad	**Hits**	**# Tweets True**	**# Tweets Pred**	**Prec**	**Rec**	**F1**
P	113	156	216	0.523 (113/216)	0.724 (113/156)	0.608
N	130	219	218	0.596 (130/218)	0.594 (130/219)	0.595
NEU	4	69	18	0.222 (4/18)	0.058 (4/69)	0.092
NONE	14	62	54	0.259 (14/54)	0.226 (14/62)	0.241
	261	506	506			

Tabla 5.28: Metricas de cada etiqueta sobre el corpus InterTASS Spain Development.

Fútbol Argentino

		Predicted				
		P	**N**	**NEU**	**NONE**	**# Tweets True**
Actual	**P**	84	6	1	1	92
	N	8	33	4	3	48
	NEU	2	6	1	2	11
	NONE	3	2	0	4	9
	# Tweets Pred	97	47	6	10	160

Tabla 5.29: Matriz de confusión del corpus Fútbol Argentino Development.

Polaridad	**Hits**	**# Tweets True**	**# Tweets Pred**	**Prec**	**Rec**	**F1**
P	84	92	97	0.866 (84/97)	0.913 (84/92)	0.889
N	33	48	47	0.702 (33/47)	0.688 (33/48)	0.695
NEU	1	11	6	0.167 (1/6)	0.091 (1/11)	0.118
NONE	4	9	10	0.400 (4/10)	0.444 (4/9)	0.421
	122	160	160			

Tabla 5.30: Metricas de cada etiqueta sobre el corpus Fútbol Argentino Development.

5.5.3 Arquitectura 3

Corpus	Development				Test			
	M-Prec	**M-Rec**	**M-F1**	**Acc**	**M-Prec**	**M-Rec**	**M-F1**	**Acc**
InterTASS Spain	0.526	0.362	0.429	0.547	0.553	0.383	0.453	0.566
InterTASS Spain + GeneralTASS	0.399	0.395	0.397	0.512	0.418	0.406	0.412	0.536
InterTASS Costa Rica	0.550	0.429	0.482	0.557	0.459	0.383	0.418	0.514
InterTASS Costa Rica + GeneralTASS	0.480	0.415	0.445	0.547	0.381	0.380	0.380	0.500
InterTASS Peru	0.408	0.413	0.410	0.480	0.414	0.406	0.410	0.380
InterTASS Peru + GeneralTASS	0.412	0.440	0.426	0.418	0.396	0.400	0.398	0.408
Fútbol Argentino	0.617	0.562	0.588	0.781	0.594	0.520	0.555	0.719
Fútbol Argentino + GeneralTASS	0.549	0.514	0.531	0.775	0.563	0.519	0.540	0.712

Tabla 5.31: Resultados sobre cada corpus usando la arquitectura 3 que se vio en la Figura 4.5 de Clasificadores en Cascada.

A continuación veremos métricas más detalladas de los sistemas entrenados con los corpus:

- InterTASS Spain
- Fútbol Argentino

InterTASS Spain

		Predicted				
		P	**N**	**NEU**	**NONE**	**# Tweets True**
Actual	**P**	93	62	1	0	156
	N	35	183	1	0	219
	NEU	22	47	0	0	69
	NONE	16	45	0	1	62
	# Tweets Pred	166	337	2	1	506

Tabla 5.32: Matriz de confusión del corpus InterTASS Spain Development.

Polaridad	**Hits**	**# Tweets True**	**# Tweets Pred**	**Prec**	**Rec**	**F1**
P	93	156	166	0.560 (93/166)	0.596 (93/156)	0.578
N	183	219	337	0.543 (183/337)	0.836 (183/219)	0.658
NEU	0	69	2	0.000 (0/2)	0.000 (0/69)	0.000
NONE	1	62	1	1.000 (1/1)	0.016 (1/62)	0.032
	277	506	506			

Tabla 5.33: Metricas de cada etiqueta sobre el corpus InterTASS Spain Development.

Fútbol Argentino

		Predicted				
		P	**N**	**NEU**	**NONE**	**# Tweets True**
Actual	**P**	84	5	1	2	92
	N	8	35	4	1	48
	NEU	3	5	3	0	11
	NONE	4	2	0	3	9
	# Tweets Pred	99	47	8	6	160

Tabla 5.34: Matriz de confusión del corpus Fútbol Argentino Development.

Polaridad	**Hits**	**# Tweets True**	**# Tweets Pred**	**Prec**	**Rec**	**F1**
P	84	92	99	0.848 (84/99)	0.913 (84/92)	0.880
N	35	48	47	0.745 (35/47)	0.729 (35/48)	0.737
NEU	3	11	8	0.375 (3/8)	0.273 (3/11)	0.316
NONE	3	9	6	0.500 (3/6)	0.333 (3/9)	0.400
	125	160	160			

Tabla 5.35: Metricas de cada etiqueta sobre el corpus Fútbol Argentino Development.

5.6 Inspección de Modelos

En esta sección haremos una inspección del modelo *Logistic Regression* sobre la representación *BoW* con *TF-IDF*. Veremos las instancias entrenadas con los siguientes corpus:

- InterTASS Spain
- Fútbol Argentino

InterTASS Spain

	Positivos		Negativos	
	Feature	**Peso**	**Feature**	**Peso**
P	POS_WORD	4.96081704	NEG_WORD	-2.95765413
	TW_HASHTAG	1.49350298	a	-1.05985715
	mucho	1.31051086	ese	-0.99785839
	POS_EMOJI	1.16887082	saber	-0.96688695
	TW_USERNAME	0.97649684	yo	-0.87181337
N	NEG_WORD	3.72356574	POS_WORD	-2.84572673
	ni	1.45286187	TW_USERNAME	-1.98691427
	porque	1.34657259	POS_EMOJI	-1.2412853
	mismo	1.2675081	semana	-1.1004771
	yo	1.2208654	ahora	-0.86039725
NEU	mas	1.37519939	POS_WORD	-1.3353895
	ese	1.01931638	poder	-1.11962007
	estar	0.88899131	TW_HASHTAG	-0.9509006
	casa	0.83514847	y	-0.84599809
	aunque	0.80455705	hoy	-0.79638206
NONE	o	1.29062504	POS_WORD	-2.12006924
	fecha	1.09677637	NEG_WORD	-2.11670766
	a	1.06900932	mucho	-1.4410048
	proximo	0.93013886	ser	-1.10865496
	para	0.89464235	LAUGHT_EXPRESSION	-1.05589982

Tabla 5.36: Features más relevantes para cada etiqueta de modelo *Logistic Regression* con representación *BoW* en la instancia *InterTASS Spain*.

Acá vemos que el modelo asigna correctamente pesos a features importantes para determinar la pertenencia o no pertenencia a algunas clases, como por ejemplo POS_WORD y POS_EMOJI para acercar el tweet a la clase P, y NEG_WORD para alejarlo de la misma.

Sin embargo, vemos también que algunas features son incorrectamente considerados importantes para la clasificación. En particular, varias stopwords son consideradas relevantes. Se debe en parte a que las stopwords no son removidas del texto. Esta decisión fue tomada experimentalmente ya que la eliminación de stopwords resultaba en peores métricas.

Veamos el siguiente tweet de ejemplo:

"@jonathanchacon te deseo muchísima suerte en tu próxima aventura tío"

Al aplicarle el preprocesamiento al contenido del tweet, queda como sigue:

"TW_USERNAME tu POS_WORD muchisima POS_WORD en tu proximo POS_WORD tio"

El cual nuestro sistema clasificó correctamente con la etiqueta **P**. En la siguiente tabla podemos ver los features que generó nuestro sistema, con los respectivos pesos para cada etiqueta.

		Pesos			
Feature	**TF-IDF**	**P**	**N**	**NEU**	**NONE**
tu	0.45121548018843954	0.61719369	-0.37704688	0.13700047	-0.46798226
tio	0.3838969875536279	-0.21856589	0.10205439	-0.26563386	0.40886369
proximo	0.4287903659397659	-0.36386282	-0.10028474	-0.40802661	0.93013886
muchisima	0.5517143951943799	-0.05938522	0.10054781	-0.03454095	-0.02106677
en	0.18997779626861014	-0.40112548	-0.29460923	0.08912455	0.76685696
TW_USERNAME	0.1109090730506062	0.97649684	-1.98691427	0.10529447	0.80325579
POS_WORD	0.335239251935218	4.96081704	-2.84572673	-1.3353895	-2.12006924
Bias		-1.03804281	-0.65166631	-1.75782846	-1.19678291
Result		0.66291122	-2.00048012	-2.41106756	-1.33972674

Tabla 5.37: Features y pesos para cada etiqueta de un tweet de ejemplo.

Veamos otro tweet de ejemplo:

"@KharanosGame Que Kinox no me quiere zi yo zoy buena perzona"

Al aplicarle el preprocesamiento al contenido del tweet, queda como sigue:

"TW_USERNAME que kinox NEG_WORD yo querer zi yo zoy POS_WORD perzona"

El cual nuestro sistema clasificó correctamente con la etiqueta **N**. En la siguiente tabla podemos ver los features que generó nuestro sistema, con los respectivos pesos para cada etiqueta.

		Pesos			
Feature	**TF-IDF**	**P**	**N**	**NEU**	**NONE**
yo	0.5712908982392552	-0.87181337	1.2208654	0.20681879	-1.00521835
querer	0.7028200055564803	-0.41222266	-0.19633825	0.2908453	0.45784822
que	0.24305552777648415	-0.56380186	0.60651199	0.4957437	-0.90147801
TW_USERNAME	0.19839787126697705	0.97649684	-1.98691427	0.10529447	0.80325579
POS_WORD	0.19989574077717154	4.96081704	-2.84572673	-1.3353895	-2.12006924
NEG_WORD	0.20316184999095313	-2.95765413	3.72356574	-0.06718032	-2.11670766
Bias		-1.03804281	-0.65166631	-1.75782846	-1.19678291
Result		-1.37835674	-0.15133309	-1.57446657	-2.36284214

Tabla 5.38: Features y pesos para cada etiqueta de un tweet de ejemplo.

Fútbol Argentino

	Positivos		Negativos	
	Feature	**Peso**	**Feature**	**Peso**
P	♡	3.33994582	var	-2.47388377
	gracias	2.6446532	river	-2.23819914
	amar	2.55546563	si	-2.08794774
	aguante	2.2683811	penal	-1.54485136
	vida	2.25027126	riber	-1.37496272
N	var	2.07295678	♡	-2.44317156
	river	1.74499434	TW_HASHTAG	-2.20550102
	robar	1.6673976	amar	-1.98981372
	cagaron	1.60053368	racing	-1.80274973
	ayer	1.39063828	boca	-1.77356729
NEU	ver	1.68035554	♡	-1.38088257
	si	1.48932478	TW_HASHTAG	-1.30995554
	lanus	1.35201705	gracias	-0.9241709
	extrañar	1.25470401	siempre	-0.85369453
	entender	1.22310206	partido	-0.80917964
NONE	TW_HASHTAG	2.8187451	♡	-1.97407746
	gol	1.43077551	gracias	-1.20982377
	victoria	1.42564159	amar	-0.99665682
	tras	1.27381784	♡	-0.9717712
	vs	1.16702212	futbol	-0.96075577

Tabla 5.39: Features más relevantes para cada etiqueta de modelo *Logistic Regression* con representación *BoW* en la instancia *Fútbol Argentino*.

Acá vemos que el modelo asigna correctamente pesos a features importantes para determinar la pertenencia o no a algunas clases, como por ejemplo “gracias” y “amar” para acercar el tweet a la clase P. Notar que el emoji ♡ acerca el tweet a la clase P y lo aleja del resto de las clases.

Sin embargo, vemos también algunas features que parecen incorrectas, como por ejemplo “var” y “river”, que alejan al tweet de la clase P y la acercan a la clase N. Asi como tambien las features “racing” y “boca” alejan el tweet de la clase N. Esto se debe a un sesgo existente en los datos de entrenamiento, que fueron recolectados en un contexto favorable para determinados equipos y desfavorables para otros (ver subsección 3.2.1). Una posible solución a esto es, en el preprocesamiento, mapear todos los nombres propios de equipos del fútbol argentino a un placeholder.

Veamos el siguiente tweet de ejemplo:

“Como no amar a los jugadores y ex jugadores de river si son lo mas”

Al aplicarle el preprocesamiento al contenido del tweet, queda como sigue:

“amar jugador ex jugador river si mas”

El cual nuestro sistema clasificó correctamente con la etiqueta **P**. En la siguiente tabla podemos ver los features que generó nuestro sistema, con los respectivos pesos para cada etiqueta.

		Pesos			
Feature	**TF-IDF**	**P**	**N**	**NEU**	**NONE**
si	0.2850459347365819	-2.08794774	0.71110218	1.48932478	0.42695145
river	0.15496970861046958	-2.23819914	1.74499434	0.59715438	0.0180542
mas	0.31467830701143656	1.86248662	-1.09718662	-0.66643011	-0.73426669
jugador	0.6585316028191224	-0.40073515	0.59864487	-0.24397841	0.12128562
ex	0.5154555649950696	-0.1910646	0.03086071	-0.08668619	0.27776672
amar	0.31040692035430134	2.55546563	-1.98981372	-0.36520628	-0.99665682
Bias		-0.24832938	-0.71936478	-2.29071679	-2.11559498
Result		-0.17340719	-0.79902562	-2.302074	-2.30847652

Tabla 5.40: Features y pesos para cada etiqueta de un tweet de ejemplo.

Veamos otro tweet de ejemplo:

"@Liberotyc que dejen de llorar las gallina river y boca se cansan de robar a los chicos"

Al aplicarle el preprocesamiento al contenido del tweet, queda como sigue:

"TW_USERNAME dejar llorar gallina river boca cansar robar chico"

El cual nuestro sistema clasificó correctamente con la etiqueta **N**. En la siguiente tabla podemos ver los features que generó nuestro sistema, con los respectivos pesos para cada etiqueta.

		Pesos			
Feature	**TF-IDF**	**P**	**N**	**NEU**	**NONE**
robar	0.33521262380026623	-1.12744096	1.6673976	-0.22263758	-0.59131661
river	0.13955510581351369	-2.23819914	1.74499434	0.59715438	0.0180542
llorar	0.33521262380026623	-0.99392991	1.12219503	0.32839828	-0.53064791
gallina	0.3930700606700862	-0.61711666	0.97894603	-0.23263869	-0.19402449
dejar	0.3173305555097199	-0.70991465	0.7941489	-0.10933799	0.06338925
chico	0.46418397866298633	-0.23539685	0.14448259	0.18917923	-0.05859453
cansar	0.4904300099011251	0.02285918	-0.02267416	-0.01491124	-0.00954514
boca	0.13926931869270792	1.08239621	-1.77356729	-0.19997015	0.06425037
TW_USERNAME	0.15875234892063872	-0.88497047	0.32124025	0.57253361	0.42770983
Bias		-0.24832938	-0.71936478	-2.29071679	-2.11559498
Result		-1.82744267	0.95600739	-2.15452572	-2.50035379

Tabla 5.41: Features y pesos para cada etiqueta de un tweet de ejemplo.

5.7 Análisis de Errores

En esta sección veremos las principales fuentes de error para la clasificación de los tweets del modelo modelo *Logistic Regression* sobre la representación *BoW* con *TF-IDF*. Veremos las instancias entrenadas con los siguientes corpus:

- InterTASS Spain
- Fútbol Argentino

InterTASS Spain

Si miramos la matriz de confusión de la Tabla 5.7 vemos que se confunden con frecuencia los tweets con etiquetas NEU y NONE con las etiquetas P o N. Analizemos algunos tweets:

#	Tweet	Ground Truth	Prediction
1	Ayer fue un día de emociones muy encontrada se acerca septiembre por un lado y por el otro una personita que siempre me hace sonreír	NEU	P
2	Soy muy obvia pero me la suda	NEU	N
3	@ItsMeCar0l ¡Hooola, buenas! Soy el que tenía los ojos rojos en la fiesta de Elle ¡Te sigo !	NONE	P

Tabla 5.42: Ejemplos de tweets con etiquetas NEU y NONE que se confunden con las etiquetas P o N.

En el tweet 1, vemos que el contenido del mismo tiene muchas expresiones que resaltan una actitud positiva, como por ejemplo "siempre me hace sonreír", pero la etiqueta del tweet es NEU. Por lo que consideramos que el tweet fue mal anotado.

En el tweet 2, notamos una actitud negativa, la cual es "me la suda", por lo que es muy posible que también haya sido mal anotado.

En el tweet 3, hay una actitud positiva, la cual es "Hooola, buenas", por lo que es muy posible que también haya sido mal anotado.

Otra conclusión que podemos sacar de la matriz de confusión es que entre las etiquetas P y N hay bastantes confusiones. Analizemos algunos casos:

#	**Tweet**	**Ground Truth**	**Prediction**
1	@KikeMlaga @sanchezcastejon ¿Socialistas honrados? Esto es un oxímoron.	N	P
2	Que os den, buenas noches	N	P
3	@Rokkowen TODOS SIEMPRE tienen carita de super buenos y adorables pero luego la lían	N	P
4	@Pablogorrito @Pablo_VzP No sabe a mierda, está muy buena	P	N
5	@vivanlospipis ser es muy malo pero bueno te salvas porque te queremos igual	P	N

Tabla 5.43: Ejemplos de tweets que se confunden entre las etiquetas P y N.

En el tweet 1, nuestra predicción fue errónea, ya que analizando el contenido del tweet notamos que es una pregunta retórica, haciendo referencia a que no existen los socialistas honrados. Nuestro modelo le dio más peso a la palabra "honrados" para decantarse por la etiqueta P.

En el tweet 2, un anotador humano dudaría de si etiquetar el tweet como positivo, negativo o neutro, ya que la frase "Que os den" es negativa, y "buenas noches" es positiva. Al haber dos expresiones, positivas y negativas, el tweet podría ser etiquetado con la etiqueta NEU. Nuestro modelo le da más peso a "buenas", acercandolo a la etiqueta P.

En el tweet 3, hay varias referencias positivas, pero al usar la conjunción "pero" invierte el sentido a la frase. El modelo no lo tiene en cuenta, por lo que le está dando más peso a las referencias positivas, acercandolo a la etiqueta P.

En el tweet 4, el contenido es claramente positivo. Sin embargo nuestro modelo está dando más peso a la expresión "mierda", lo que lo acerca a la etiqueta N, sin tener en cuenta que este término se encuentra negado. La solución a este problema es hacer un mejor tratamiento de las negaciones en nuestros modelos.

En el tweet 5, el contenido es claramente positivo, notar la importancia de la frase que se encuentra después del "pero". Sin embargo nuestro modelo le está dando más importancia a la palabra "malo", acercandolo a la etiqueta N.

Fútbol Argentino

Si miramos la matriz de confusión de la Tabla 5.9 vemos son pocos aquellos tweets con etiquetas NEU y NONE que se confunden con las etiquetas P o N. Analizemos algunos tweets:

#	Tweet	Ground Truth	Prediction
1	Parece que el Var en la cancha de Lanús lo manejaban peritos de Gendarmería	NEU	N
2	90 MINUTOS DE FÚTBOL - RIVER SE CONFIÓ Y LANÚS SE LO DIO VUELTA - 01/11/2017: https://t.co/pYKlHQqzI3 a través de @YouTube	NONE	N

Tabla 5.44: Ejemplos de tweets con etiquetas NEU y NONE que se confunden con las etiquetas P o N.

En el tweet 1, se está criticando el uso del "Var", en principio no es una actitud negativa, por lo que la etiqueta NEU estaría bien puesta. Pero debido al sesgo del dataset, el uso de la palabra "Var" hace que el modelo lo acerque a la etiqueta N.

En el tweet 2, esta bien puesta la etiqueta NONE ya que es un tweet periodístico. Pero al hablar de River y teniendo en cuenta el sesgo del dataset, el modelo acerca el tweet a la etiqueta N.

Otra conclusión que podemos sacar de la matriz de confusión es que entre las etiquetas P y N pocas confusiones. Analizemos algunos casos:

#	Tweet	Ground Truth	Prediction
1	Uh no tengo cambio tenes mas chiquito? -Boca festejando que hace un año es puntero	N	P
2	@EstebanPerez_5 @NaiAranda1 TE GANO HURACAN BOLUDO	N	P
3	Me vuelvo loco. Saquenle la Copa a River y densela a los tucumanos. Cantaron el Ji Ji Ji del Indio. Qué groso	P	N
4	Sos el River de mi bidha	P	N

Tabla 5.45: Ejemplos de tweets que se confunden entre las etiquetas P y N.

En el tweet 1, se está haciendo una pregunta retórica, siendo el sentido general del contenido una crítica hacia Boca. El modelo no se da cuenta de ello, por lo que da peso positivo a la palabra "festejando", acercando el tweet a la etiqueta P.

En el tweet 2, el contenido es una burla, por lo que la etiqueta N está bien puesta. Sin embargo, nuestro modelo le da más peso a la palabra "GANO", haciendo que se decante por la etiqueta P.

En el tweet 3 y 4, el contenido del tweet es positivo. Pero cómo se está hablando de River, dado el sesgo del dataset, el modelo acerca dichos tweets a la etiqueta N.

Capítulo 6

Conclusiones y Trabajo Futuro

En este trabajo, hemos estudiado un problema complejo como el Análisis de Sentimiento en Tweets, en donde las principales dificultades radican en aspectos relacionados al dominio. Entre estas dificultades se encuentran la multilingualidad de las diversas variantes del español, la falta de contexto debido a la limitada longitud de los tweets, y el lenguaje informal comúnmente usado en redes sociales que lleva a errores de ortografía y la utilización de términos especiales como jergas, emoticones y emojis.

En la construcción de nuestro corpus, se prestó especial atención al diseño de todas las etapas para la obtención del mismo. En la recolección de los tweets incluimos las particularidades del español de Argentina, como ser palabras, expresiones y dialectos propios del país. En el filtrado de los tweets creamos una guía para que los anotadores puedan discriminar los tweets y así quedarnos solamente con aquellos que hablan del fútbol argentino. Finalmente para el etiquetado de polaridad también diseñamos una guía con criterios claros, teniendo en cuenta varios aspectos del lenguaje, para que se pueda identificar de mejor manera la polaridad de los tweets.

Comparando el proceso de etiquetado que seguimos nosotros con la forma semiautomática seguida por la SEPLN, destacamos que las etiquetas en los tweets de nuestro corpus fueron íntegramente puestas por anotadores humanos. A esto se le suma el hecho de que cada tweet fue etiquetado dos veces, generando un corpus de mayor confianza. Esperamos que en un futuro nuestro corpus pueda formar parte de los conjuntos de datos de las próximas ediciones del TASS para la tarea de Análisis de Sentimiento.

Como podemos ver en los resultados del capítulo 5, los sistemas desarrollados clasifican bien los tweets con etiquetas **P** y **N**, pero son poco efectivos en la detección de tweets con etiquetas **NEU** y **NONE**. El hecho de que haya pocos representantes de estas clases en los corpus y su similitud tanto con los tweets positivos como con los negativos pueden llegar a ser causas del bajo desempeño en su detección. Haciendo inspección de los modelos y análisis de errores pudimos ver otras dificultades para nuestros clasificadores, y posibles formas de superarlas.

Como trabajo futuro, proponemos realizar Análisis de Sentimiento basado en aspectos, en donde nuestros sistemas deberán identificar la polaridad de los aspectos que se encuentren en los tweets. Para ello necesitamos realizar una anotación manual de los distintos aspectos encontrados. Para la extracción de éstos podríamos usar un sistema de NER (Named Entity Recognition) para poder localizar y clasificar en categorías predefinidas, como personas, organizaciones, lugares y otras entidades nombradas. Por último, proponemos implementar modelos de redes neuronales profundas, utilizando modelos de lenguaje neuronales basados en Redes Neuronales Recurrentes (RNNs) o Transformers, como por ejemplo *ELMo* [Peters et al., 2018]

o *BERT* [Devlin et al., 2018] respectivamente. Se ha demostrado que estos modelos permiten mejorar el desempeño en múltiples tareas de PLN, gracias al pre-entrenamiento que se puede realizar con grandes cantidades de datos sin etiquetar.

Capítulo 7

Anexo de Documentos

Etiquetado de Corpus sobre Fútbol Argentino

7.1 Filtrado de tweets

Instrucciones: Dado un mensaje de Twitter, etiquetarlo según si este habla de fútbol argentino o no:

- Yes (**Y**)
- No (**N**)
- Unknown (**U**)

7.1.1 Criterios para etiquetado

Yes (Y)

El tweet habla explícita o implícitamente de uno o más clubes de fútbol argentino, o de una situación que involucra a los clubes en un contexto futbolístico.
Esto incluye referencias a partidos, jugadores, árbitros, hinchada, dirigentes, etc.

No (N)

El tweet **NO** habla explícita o implícitamente de uno o más clubes de fútbol argentino, o de una situación que involucra a los clubes en un contexto futbolístico.

Unknown (U)

El anotador no puede deducir de la información del tweet si la etiqueta es **Y** o **N**.

Para darse una pequeña idea sobre lo explicado anteriormente, pueden ver en el Cuadro 1 de abajo ejemplos de tweets con su respectiva anotación.

	Tweet	Tag
1	En River se quejan por fallos que lo perjudicaron. Y está bien. Pero la autocríticadebe incluir la flojísima actuación. Y el bochorno que significa que le den vuelta unaventaja de de 3-0 con 4 goles en poco más de 20 minutos. Eso fue hazaña de Lanús.	Y
2	Me estoy imaginando todo lo que puede hacer con esa boca https://t.co/SH3EMKkwpA	N
3	La lluvia fue mística... A mi no me jodanFueron los que hoy ya no están presentes pero hicieron sentir que estaban ahí connosotros#Lanus	U
4	Lo mire por Fox pero fue hermoso!Despertó riber del sueño eterno! Volviste a larealidad !!! Ya te pasó con los cuervos. Felicitaciones Lanús	Y
5	Acabo de escupir sangre por la boca mi vida cada vez va a mejor	N
6	@Belgrano Gracias Torta y CD!!!!	U
7	#Clarín Iván Marcone: una confesión y un diálogo revelador con el árbitro https://t.co/3sBXUuGhttps://t.co/vBThzIZ1PL	Y
8	39': Lo sigue intentando el Racing que empieza a acechar con insistencia el área delBurgos #RRClive (0-0) https://t.co/F7FNEhBZkH	N
9	ESTO ES LA COPA LIBERTADORES LA CONCHA BIEN DE TU MADRECHAMPIONS LEAGUE NI QUE NADA https://t.co/zL0gy6ii6z	U
10	@La12tuittera Y todo lo q hizo el delincuente del @TanoAngelici para q a boca nole dieran una sancion mas dura , despues del gas pimienta!	Y
11	Los chicos de @CAOUvoley presentes en la Copa Argentina sub 19 en Chapadmalal#CopasArgentinas2017 #ValoresEnJuego https://t.co/wU92pz96aC	N
12	Gran encuentro! @Leanderacing con @LiamArreche y @estebansimaro Jugadores deSelección Voley y enfermos de #RACING https://t.co/7Sq8tzJaFe https://t.co/iTLeTgFQH6	N

Tabla 7.1: Ejemplos de tweets anotados.

Nota:

- 2^{do} y 5^{to} item: hablan de boca, pero sobre la parte del cuerpo.
- 7^{mo} item: Iván Marcone es un jugador de Lanus.
- 8^{vo} item: habla sobre Racing pero de España.
- 10^{mo} item: Angelici es el presidente de Boca.
- 11^{vo} y 12^{vo} item: hablan sobre Racing, pero no de fútbol sino de voley.

7.1.2 Keywords de equipos del fútbol argentino:

- Boca (apodos: xeneize, bostero)
- River (apodos: millonario, la banda, gallina)
- Banfield (apodos: el taladro)
- Colon (apodos: el sabalero)
- San Lorenzo (apodos: ciclon, cuervo)
- Talleres (apodos: albiazul, matador, tallarin)
- Huracan (apodos: globo)
- Velez Sarsfield (apodos: el fortin)
- Belgrano (apodos: pirata, celeste)
- Godoy Cruz (apodos: tomba)
- Lanus (apodos: granate)
- Racing (apodos: la academia)
- Independiente (apodos: rojo, diablos rojos)
- Newells (apodos: leproso, la lepra)
- San Martin SJ (apodos: santo sanjuanino, verdinegro)
- Gimnasia de la Plata (apodos: lobo)
- Atletico Tucuman (apodos: el decano)
- Estudiantes (apodos: picha)
- Defensa y Justicia (apodos: halcon)
- Temperley (apodos: el gasolero)
- Rosario Central (apodos: canallas)
- Chacarita (apodos: chaca, funebrero)
- Tigre (apodos: matador)
- Argentinos Juniors (apodos: bicho)
- Arsenal (apodos: arse)

7.1.3 Opciones para el etiquetado

Para etiquetar los tweets, vamos a usar las siguientes letras:

- **y** $\longleftrightarrow$ Yes (Y)
- **n** $\longleftrightarrow$ No (N)
- **u** $\longleftrightarrow$ Unknown (U)

7.2 Polaridad del tweet

Instrucciones: Dado un mensaje de Twitter, identificar si el mensaje es:

- Positivo (**P**)
- Negativo (**N**)
- Neutral (**NEU**)
- No expresa sentimiento (**NONE**)

7.2.1 Criterios para anotación de polaridad

Positivo (P)

Se esta usando lenguaje positivo.
Por ejemplo:

(a) Expresiones de apoyo

(b) Admiración

(c) Actitud positiva

(d) Resaltado de éxitos

(e) Estado emocional positivo

El twittero[1] demuestra felicidad, admiración, relajación, indulgencia[2].

Negativo (N)

Se esta usando lenguaje negativo.
Por ejemplo:

(a) Expresiones de critica

(b) Emitir un juicio critico

(c) Actitud negativa

(d) Fracasos

(e) Emociones negativas

El twittero demuestra tristeza, enojo, violencia.

Neutral (NEU)

(a) Se esta usando lenguaje positivo y negativo al mismo tiempo.

(b) Se expresa un sentimiento pero no se puede determinar cual es.
Ej: Posible sarcasmo.

El twittero demuestra sentimientos pero no pueden ser identificados como claramente positivos o negativos.

[1] **Twittero**: persona que escribió el tweet.
[2] **Indulgencia**: que perdona o disculpa sus errores y de los demás.

No expresa sentimiento (NONE)

No se esta usando ni lenguaje positivo ni lenguaje negativo.
Por ejemplo:

(a) Tweets periodísticos que no expresan la opinión del twittero.

(b) Preguntas no retóricas[3].

El twittero no indica estado emocional alguno.

Preguntas retóricas y Citas textuales

(a) Pregunta retórica[4]: en caso de que el tweet sea una pregunta retórica, determinar el sentimiento del mismo en base al contenido de la pregunta.

(b) Cita textual[5]: en caso de que el tweet sea una cita textual, determinar el sentimiento del mismo en base al contenido de la cita.

Para darse una pequeña idea sobre lo explicado anteriormente, pueden ver en el Cuadro 1 de abajo ejemplos de tweets con su respectiva anotación de polaridad y su correspondiente caso en base a lo explicado en cada subsección correspondiente (Positivo, Negativo, Neutral, No expresa sentimiento, Preguntas retóricas y Citas textuales).

[3]**Preguntas no retóricas**: preguntas que se realizan con el objetivo de que se nos brinde una respuesta con la información que buscamos.

[4]**Preguntas retóricas**: preguntas que se realizan sin esperar una respuesta e incluso pueden no contar con un destinatario específico.

[5]**Cita textual**: el texto es literalmente copiado tal cual como se dijo, se encuentra entre comillas.

Tweet	Sentiment	Case
Todavía no caigo que estamos en la final de la Copa Libertadores, en mi puta vida pensé que iba a poder vivir esto	P	(d)
Es imposible obviar lo mal que jugó River en el segundo tiempo con el perjuicio que recibió desde lo arbitral porque ambas fueron reales.	N	(a)
En River se quejan por fallos que lo perjudicaron. Y está bien. Pero la autocrítica debe incluir la flojísima actuación. Y el bochorno que significa que le den vuelta una ventaja de de 3-0 con 4 goles en poco más de 20 minutos. Eso fue hazaña de Lanús.	NEU	(a)
Driussi y Ramiro Funes Mori se la bancan: tras la eliminación de River, compartieron estas imágenes atacando a Boca.	NONE	(a)
@mercado_river El VAR lo de la CONMEBOL,y los de la "nueva AFAno" RIVER PLATE EL MAS GRANDE LEJOS BOSTERO VIVIRAS DE NOSOTROS	P	(a)
Más allá del flojisimo 2T de River, se cumplió lo que se venía rumoreando, los arbitros iban a perjudicar a River, lo del VAR es increíble	N	(a) (b)
En menos de una semana quedó afuera Newbery y River. Linda semanita de futbol me tocó.	NEU	(b)
¿Alemán está jugando vs #Boca ?	NONE	(b)
Sand porqué no te vas a la puta que te parió?	N	Pregunta retórica
.@JulioFalcioniDT en @directvsportsar: "Dirigir a Boca es lo máximo en Argentina y del futbol mundial deber estar ahí con Barcelona y Real Madrid, sin ninguna duda" https://t.co/ZvkfIp8DDx	P	Cita textual

Tabla 7.2: Ejemplos de tweets anotados y casos correspondientes.

7.2.2 Opciones para el etiquetado de polaridad

Para etiquetar los tweets según su polaridad, vamos a usar las siguientes letras:

- **p** $\longleftrightarrow$ Positivo (P)
- **n** $\longleftrightarrow$ Negativo (N)
- **u** $\longleftrightarrow$ Neutral (NEU)
- **x** $\longleftrightarrow$ No expresa sentimiento (NONE)

Bibliografía

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag.

[Bojanowski et al., 2016] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

[Cambria et al., 2017] Cambria, E., Das, D., Bandyopadhyay, S., and Feraco, A. (2017). *A Practical Guide to Sentiment Analysis*, chapter 4, pages 61–83. Springer Publishing Company, Incorporated, 1st edition.

[Cardellino, 2019] Cardellino, C. (2019). Spanish Billion Words Corpus and Embeddings.

[Cerón-Guzmán, 2016] Cerón-Guzmán, J. A. (2016). JACERONG at TASS 2016: An ensemble classifier for sentiment analysis of spanish tweets at global level. In Villena-Román, J., Cumbreras, M. Á. G., Cámara, E. M., Díaz-Galiano, M. C., Martín-Valdivia, M. T., and López, L. A. U., editors, *Proceedings of TASS 2016: Workshop on Sentiment Analysis at SEPLN co-located with 32nd SEPLN Conference (SEPLN 2016), Salamanca, Spain, September 13th, 2016*, volume 1702 of *CEUR Workshop Proceedings*, pages 35–39. CEUR-WS.org.

[Ciresan et al., 2012] Ciresan, D. C., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. *CoRR*, abs/1202.2745.

[contributors, 2020a] contributors, W. (2020a). Bag-of-words model — Wikipedia, the free encyclopedia.

[contributors, 2020b] contributors, W. (2020b). Natural language processing — Wikipedia, the free encyclopedia.

[contributors, 2020c] contributors, W. (2020c). Precision and recall — Wikipedia, the free encyclopedia.

[Cruz et al., 2014] Cruz, F. L., Troyano, J. A., Pontes, B., and Ortega, F. J. (2014). Ml-senticon: A multilingual, lemma-level sentiment lexicon. *Procesamiento de Lenguaje Natural*, 53:113–120.

[Devlin et al., 2018] Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

[Diplomatura en Ciencia de Datos, 2018a] Diplomatura en Ciencia de Datos, A. A. y. s. A. (2018a). Aprendizaje por refuerzo.

[Diplomatura en Ciencia de Datos, 2018b] Diplomatura en Ciencia de Datos, A. A. y. s. A. (2018b). Introducción al Aprendizaje Automático.

[Gandhine, 2018] Gandhine, R. (2018). Support Vector Machine - Introduction to Machine Learning Algorithms. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

[Gilman, 2018] Gilman, R. (2018). Intuitive RL: Intro to Advantage-Actor-Critic (A2C). https://hackernoon.com/intuitive-rl-intro-to-advantage-actor-critic-a2c-4ff545978752.

[Glossary, 2017] Glossary, M. L. (2017). Neural Networks - Architectures - MLP. https://ml-cheatsheet.readthedocs.io/en/latest/architectures.html#mlp.

[Hinton and Sejnowski, 1999] Hinton, G. and Sejnowski, T. J. (1999). *Unsupervised Learning: Foundations of Neural Computation*. The MIT Press.

[Hurtado et al., 2017] Hurtado, L., Pla, F., and González, J. (2017). ELiRF-UPV en TASS 2017: Análisis de Sentimientos en Twitter basado en Aprendizaje Profundo. In *Proceedings of TASS 2017: Workshop on Sentiment Analysis at SEPLN co-located with 33nd SEPLN Conference (SEPLN 2017), Murcia, Spain, September 19th, 2017*, volume 1896 of *CEUR Workshop Proceedings*, pages 29–34. CEUR-WS.org.

[Karn, 2016] Karn, U. (2016). An Intuitive Explanation of Convolutional Neural Networks. https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

[Landis and Koch, 1977] Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

[Luque and Pérez, 2018] Luque, F. M. and Pérez, J. M. (2018). Atalaya at TASS 2018: Sentiment analysis with tweet embeddings and data augmentation. In Cámara, E. M., Almeida-Cruz, Y., Díaz-Galiano, M. C., Estévez-Velarde, S., Cumbreras, M. Á. G., Vega, M. G., Gutiérrez, Y., Montejo-Ráez, A., Montoyo, A., Muñoz, R., Piad-Morffis, A., and Villena-Román, J., editors, *Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN, TASS@SEPLN 2018, co-located with 34nd SEPLN Conference (SEPLN 2018), Sevilla, Spain, September 18th, 2018*, volume 2172 of *CEUR Workshop Proceedings*, pages 29–35. CEUR-WS.org.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

[Moctezuma et al., 2017] Moctezuma, D., Graff, M., Miranda-Jiménez, S., Tellez, E. S., Coronado, A., Sánchez, C. N., and Ortiz-Bejar, J. (2017). A Genetic Programming Approach to Sentiment Analysis for Twitter: TASS'17. In *Proceedings of TASS 2017: Workshop on Sentiment Analysis at SEPLN co-located with 33nd SEPLN Conference (SEPLN 2017), Murcia, Spain, September 19th, 2017*, volume 1896 of *CEUR Workshop Proceedings*, pages 23–28. CEUR-WS.org.

[Mohammad, 2016] Mohammad, S. M. (2016). A practical guide to sentiment annotation: Challenges and solutions. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, San Diego, CA, USA, June 12-17, 2016*, pages 174–179. Association for Computational Linguistics.

[Mohri et al., 2012] Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012). *Foundations of Machine Learning*. The MIT Press.

[Molina-González et al., 2013] Molina-González, M. D., Martínez-Cámara, E., Martín-Valdivia, M. T., and Perea-Ortega, J. M. (2013). Semantic orientation for polarity classification in spanish reviews. *Expert Syst. Appl.*, 40(18):7250–7257.

[Montejo-Ráez and Díaz-Galiano, 2016] Montejo-Ráez, A. and Díaz-Galiano, M. C. (2016). Participación de SINAI en TASS 2016. In Villena-Román, J., Cumbreras, M. Á. G., Cámara, E. M., Díaz-Galiano, M. C., Martín-Valdivia, M. T., and López, L. A. U., editors, *Proceedings of TASS 2016: Workshop on Sentiment Analysis at SEPLN co-located with 32nd SEPLN Conference (SEPLN 2016), Salamanca, Spain, September 13th, 2016*, volume 1702 of *CEUR Workshop Proceedings*, pages 41–45. CEUR-WS.org.

[Murillo and Raventós, 2016] Murillo, E. C. and Raventós, G. M. (2016). Evaluación de modelos de representación del texto con vectores de dimensión reducida para análisis de sentimiento. In Villena-Román, J., Cumbreras, M. Á. G., Cámara, E. M., Díaz-Galiano, M. C., Martín-Valdivia, M. T., and López, L. A. U., editors, *Proceedings of TASS 2016: Workshop on Sentiment Analysis at SEPLN co-located with 32nd SEPLN Conference (SEPLN 2016), Salamanca, Spain, September 13th, 2016*, volume 1702 of *CEUR Workshop Proceedings*, pages 23–28. CEUR-WS.org.

[Nakov et al., 2016] Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., and Stoyanov, V. (2016). Semeval-2016 task 4: Sentiment analysis in twitter. In Bethard, S., Cer, D. M., Carpuat, M., Jurgens, D., Nakov, P., and Zesch, T., editors, *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 1–18. The Association for Computer Linguistics.

[Navas-Loto and Rodríguex-Doncel, 2017] Navas-Loto, M. and Rodríguex-Doncel, V. (2017). OEG at TASS 2017: Spanish Sentiment Analysis of tweets at document level. In *Proceedings of TASS 2017: Workshop on Sentiment Analysis at SEPLN co-located with 33nd SEPLN Conference (SEPLN 2017), Murcia, Spain, September 19th, 2017*, volume 1896 of *CEUR Workshop Proceedings*, pages 43–49. CEUR-WS.org.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Peters et al., 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *CoRR*, abs/1802.05365.

[Powers, 2011] Powers, D. M. W. (2011). Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. *Machine Learning Technologies*, 2:37–63.

[Quirós et al., 2016] Quirós, A., Segura-Bedmar, I., and Martínez, P. (2016). LABDA at the 2016 TASS challenge task: Using word embeddings for the sentiment analysis task. In Villena-Román, J., Cumbreras, M. Á. G., Cámara, E. M., Díaz-Galiano, M. C., Martín-Valdivia, M. T., and López, L. A. U., editors, *Proceedings of TASS 2016: Workshop on Sentiment Analysis at SEPLN co-located with 32nd SEPLN Conference (SEPLN 2016), Salamanca, Spain, September 13th, 2016*, volume 1702 of *CEUR Workshop Proceedings*, pages 29–33. CEUR-WS.org.

[Radhakrishnan, 2017] Radhakrishnan, P. (2017). Introduction to Recurrent Neural Network. https://towardsdatascience.com/introduction-to-recurrent-neural-network-27202c3945f3.

[Rajaraman and Ullman, 2011] Rajaraman, A. and Ullman, J. D. (2011). Data Mining (PDF). In *Mining of Massive Datasets*, pages 1–17, USA. Cambridge University Press.

[Reyes-Ortiz et al., 2017] Reyes-Ortiz, J. A., Paniagua-Reyes, F., Priego, B., and Tovar, M. (2017). LexFAR en la competencia TASS 2017: Análisis de sentimientos en Twitter basado en lexicones. In *Proceedings of TASS 2017: Workshop on Sentiment Analysis at SEPLN co-located with 33nd SEPLN Conference (SEPLN 2017), Murcia, Spain, September 19th, 2017*, volume 1896 of *CEUR Workshop Proceedings*, pages 51–57. CEUR-WS.org.

[Russell and Norvig, 2009] Russell, S. J. and Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, 3rd edition.

[Schmid, 1995] Schmid, H. (1995). Improvements in part-of-speech tagging with an application to german. In *In Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

[Schmidhuber, 2014] Schmidhuber, J. (2014). Deep learning in neural networks: An overview. *CoRR*, abs/1404.7828.

[Sim and Wright, 2005] Sim, J. and Wright, C. C. (2005). The kappa statistic in reliability studies: Use, interpretation, and sample size requirements. *Physical Therapy*, 85(3):257–268.

[StackExchange, 2017] StackExchange (2017). Clustering Plots. https://stats.stackexchange.com/questions/253926/what-are-the-x-and-y-axes-of-clustering-plots.

[Swaminathan, 2018] Swaminathan, S. (2018). Logistic Regression - Detailed Overview. https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc.

[Turney, 2002] Turney, P. D. (2002). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews.

[Urizar and Roncal, 2013] Urizar, X. S. and Roncal, I. S. V. (2013). Elhuyar at TASS 2013. In *Proceedings of the Sentiment Analysis Workshop at SEPLN (TASS2013), Madrid, Spain, September 20th, 2013*.

Los abajo firmantes, miembros del Tribunal de evaluación de tesis, damos fe que el presente ejemplar impreso se corresponde con el aprobado por este Tribunal.