Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Detección de misoginia en textos cortos mediante clasificadores supervisados

Tesis presentada para obtener el grado de:

Ingeniera en Tecnologías de la información

Presenta:

Valeria Vera Lagos

Dirección de tesis:

Dr. José Arturo Olvera López

Codirección de tesis:

Dra. Josefina Guerrero García

Octubre 2021, Puebla, México

Agradecimientos

A mis padres. Gracias, mamá y papá por dar todo para poder soltarme y dejarme encontrar mis sueños, incluso en los momentos más difíciles. Gracias por su apoyo incondicional y por creer siempre en mis capacidades y ambiciones más grandes. Detrás de mis logros, están ustedes, nada de lo que soy hoy lo habría logrado sin su ayuda.

A los profesores de la Facultad de Ciencias de la Computación, de quienes aprendí tanto y siempre me orientaron para encontrar mi camino profesional deseado. Especialmente a la Doctora Josefina Guerrero, quien me guio dentro de la investigación desde mis primeros semestres de la carrera hasta esta última defensa de tesis. El apoyo y conocimiento que me brindó son las bases de mi carrera. Agradezco también al Doctor Arturo Olvera, por aceptar dirigir mi proyecto de tesis y ayudarme a definir y cumplir mis objetivos. Gracias por el tiempo dedicado y por guiarme para concluir este trabajo de manera exitosa.

A mis amigos de carrera, con los que por 5 años compartí materias, proyectos, sueños y experiencias. Gera, Javi, Pastor, Fer, Marcos y Cesi, gracias por la motivación, el aprendizaje y el esfuerzo en cada trabajo. Agradezco nuestra amistad, en la que crecimos juntos y nos hizo llegar más lejos de lo que alguna vez imaginamos.

Por último, agradezco a todos aquellos que no nombro, pero fueron pilares a lo largo de mi carrera. Gracias a esos amigos que conocí a través de la investigación, de quienes aprendí lo lejos que se puede llegar cuando crees firmemente. Gracias a los amigos que compartieron conmigo espacio y tiempo para reír, escucharnos, crecer y hasta vivir. Gracias a aquellas mujeres en esta área: profesoras, amigas, familiares, compañeras de hogar y de clases que sirvieron como inspiración para pelear por espacios igualitarios.

A todas y todos los que me apoyaron para permitirme dedicar tiempo y pasión a mi preparación académica, ¡muchas gracias!

Índice

Índice de Ilustraciones

Índice de Tablas

Capítulo 1. Introducción

1. 1. Resumen

El análisis de sentimientos es un área de estudio que involucra el uso y manejo de información para la detección de estados de ánimo, emociones y actitudes, expresadas en texto y busca determinar la actitud de una persona relacionada a objetos, servicios, lugares, eventos o temas en específico.

Con el crecimiento exponencial de las plataformas de microblogging, ha incrementado con ellas la violencia hacia las mujeres a través de la réplica de discursos de odio, fomentando la cultura misógina con la que vivimos día a día. Aunque recientemente ha existido gran esfuerzo para identificar la misoginia de manera automática, es aún difícil distinguirla entre sexismo o lenguaje agresivo hacia las mujeres debido a su sutileza y el fuerte contexto social que conlleva. Hasta ahora, la mayoría de los análisis de este problema se enfocan en textos en inglés y la poca cantidad de análisis con textos en español no distingue su lugar de origen complicando la clasificación debido a las diferencias culturales y lingüísticas dentro del mismo idioma.

En este trabajo de tesis se lleva a cabo un análisis lingüístico sobre un corpus de opiniones en español de México. Se implementan diversas técnicas de preprocesamiento a dicho corpus a través de herramientas y librerías como “Spaceling”, “Freeling” y “NLTK” para entrenar un clasificador capaz de discernir si una opinión conocida como “Tweet” es misógina o no.

1. 2. Antecedentes del Proyecto

Los discursos de odio y agresión hacia las mujeres es una realidad que viven miles de mexicanas en las redes sociales todos los días, reflejando, a través del lenguaje, los patrones de misoginia que existen en nuestra sociedad.

Durante los últimos años, el número artículos que analizan el discurso de odio utilizado en la plataforma antes mencionada, ha aumentado. Dichos artículos ayudan a entender cómo es que el lenguaje violento afecta directamente a las víctimas de este; lamentablemente, la mayoría analizan la violencia considerando la lengua inglesa y aquellos en español no distinguen diferencias lingüísticas entre países hispanohablantes.

A pesar de que más de la mitad de las mujeres mexicanas ha enfrentado alguna vez una situación violenta [1] y se ha reportado un aumento del 43% en los casos de feminicidio en México en los últimos cinco años [2], existe una falta de datos relacionados a la violencia verbal en las mujeres mexicanas. Actualmente, no existen corpus en español de México con opiniones misóginas y mucho menos artículos que evalúen el lenguaje de nuestro país a través del análisis de sentimiento de textos cortos.

1. 3. Objetivo general

Discernir tweets misóginos de no misóginos provenientes de usuarios ubicados en México a través de un clasificador de sentimientos para textos cortos en español

1. 4. Objetivos específicos

1. Crear un conjunto de datos construido a partir de textos cortos misóginos en español de México.
2. Analizar las diferentes técnicas pertenecientes al procesamiento de lenguaje natural para determinar el enfoque en el que se basaría la clasificación de textos cortos y misoginia.
3. Hacer un análisis del desempeño obtenido a partir de cada técnica, para conocer qué técnica logra entrenar mejor al clasificador.
4. Clasificar un conjunto de tweets de usuarios con ubicación en México y analizar los resultados.

1. 5. Contribución de la tesis

Se contribuirá con la propuesta de diferentes técnicas de preprocesamiento y clasificación de textos cortos en español, así como la técnica adecuada de clasificación de dicho texto corto a partir de un corpus propio en idioma español.

Capítulo 2. Marco teórico

2. 1. Lenguaje Natural

Un lenguaje natural es aquel que ha evolucionado con el tiempo para fines de comunicación humana, como el español. A diferencia de los lenguajes formales que están definidos por reglas preestablecidas, el lenguaje natural es el medio que utilizamos de manera cotidiana para establecer comunicación con las demás personas. Una de las áreas en las que tiene aplicación la Inteligencia Artificial es el Procesamiento de Lenguaje Natural, el cual consiste en la utilización de un lenguaje para comunicarnos con la computadora, debiendo ésta entender, aprender e interactuar con el usuario. El procesamiento del lenguaje natural se beneficia de información semántica. Se define la semántica en general como el significado de una palabra, frase, oración o cualquier texto en algún lenguaje que el ser humano utilice, así como el estudio de estos. El manejo y modelado de información semántica representa en particular un problema para los sistemas de información.

2. 2. Análisis de sentimientos

Una de las tareas comunes dentro del Procesamiento de Lenguaje Natural se enfoca en encontrar un subconjunto de características semánticas dentro del conjunto total de características de textos cortos, que mejor exprese un sentimiento. Dicha solución se ha encontrado a través de la literatura con diferentes títulos como análisis de sentimiento, microblogging o minería de opinión. La mayoría de los trabajos utilizan Twitter como una fuente primaria de datos debido a su facilidad de acceso. De la misma manera, gran parte de la literatura busca encontrar la polaridad de los sentimientos, positivo o negativo y en el caso más general, encontrar emociones; entre las más populares están, felicidad o buen humor, enojo o agresividad y tristeza.

2. 3. Twitter

Twitter es una plataforma de microblogging que permite compartir opiniones en tiempo real. Es considerada como una de las redes sociales más populares del mundo. La red permite compartir mensajes de texto de corta longitud, con un máximo de 280 caracteres

(originalmente 140), llamados tuits o tweets. Como se observa en la ilustración 1, cada tweet puede contener, hashtags, emojis, menciones a otros usuarios, imágenes, videos o hipervínculos.

Las menciones se distinguen por el símbolo de arroba @ seguido del usuario que se busca mencionar o notificar, al hacer esto se crea un hipervínculo al perfil del usuario mencionado.

Los hashtags son palabras o conjuntos de palabras sin espacios, que comienzan con un símbolo numérico # y se emplean para agrupar los mensajes sobre un determinado tema.

Ilustración 1- Ejemplo de Tweet con hashtag, mención e imagen

2. 4. Misoginia

La misoginia es el desprecio, aversión o como sus raíces lo describen, el odio hacia las mujeres; se manifiesta como denigración, discriminación o violencia contra la mujer.

En [42] se refiere a la misoginia como el ambiente social en el que las mujeres tienen que enfrentar hostilidad, de diversos tipos, por el simple hecho de ser mujeres en un mundo de estándares creados por hombres. Explica una diferencia entre dicho ambiente y el sexismo el cual es un comportamiento discriminatorio entre hombres y mujeres, pero no es específico hacia un sexo. En [43] mencionan que la misoginia es una de las bases para la opresión de las mujeres en las sociedades dominadas por hombres basadas en prejuicios e ideologías. Por

último, en [44] se menciona que, aunque sucede más comúnmente en hombres, la misoginia también es practicada por mujeres contra otras mujeres o incluso hacia sí mismas.

2. 5. Modelos de representación de palabras

Una de las tareas más importantes del Procesamiento de Lenguaje Natural, es crear modelos de lenguaje o de representación de palabras a través de valores numéricos y estructuras que puedan ser procesados posteriormente por el aprendizaje de máquina.

Actualmente los métodos más comunes para crear modelos de representación son Bolsa de Palabras, TF-IDF, N-gramas y técnicas conocidas como incrustaciones de palabras (Word2Vec, GLoVe, ELMo y BERT). La principal diferencia entre los primeros métodos mencionados y las técnicas de incrustaciones de palabras es cómo categorizan las semejanzas semánticas entre palabras. En modelos como Bolsa de Palabras y métodos como TF-IDF se tiene la desventaja de ignorar el orden y relación de las palabras en los textos, enfocándose en su frecuencia de la aparición. A diferencia de los N-gramas que otorgan relevancia a la secuencia de palabras, pero ignoran la semántica de las relaciones. Este problema es resuelto a través de incrustaciones de palabras ya que vincula palabras que son similares en semántica. Su principal desventaja es que necesitan muestras grandes de textos en el lenguaje donde se va a generar el aprendizaje de máquina.

2.5. 1. N-gramas

Un N-grama, en Procesamiento de Lenguaje Natural, es un conjunto de n unidades lingüísticas consecutivas en un documento de texto. Estas unidades pueden ser palabras, sílabas, números o signos de puntuación. Los 1-gramas se conocen como unigramas, los 2-gramas se conocen como bigramas o digramas y los 3-gramas como trigramas.

Una vez teniendo un corpus agrupado por N-gramas es posible generar un modelo que demuestre relevancia en la secuencia de palabras de los textos.

En [40] Reportaron que al mantener unigramas se obtuvieron mejores resultados que al transformar con bigramas en términos de clasificación de sentimiento, por otro lado [41] reportó el resultado contrario obteniendo un mejor desempeño con bigramas y trigramas en

la misma área. Por lo que su impacto puede depender del conjunto y características a clasificar.

2.5. 2. Bolsa de palabras

En esta técnica, las palabras de un conjunto de datos se modelan como un diccionario para posteriormente representar cada texto del conjunto en función de la aparición de las palabras que contiene. Dicha aparición se puede representar con un valor binario 0 o 1 dependiendo de si existe o no en el texto, o con un entero para representar la frecuencia de aparición. Por ejemplo la bolsa de palabras de la tabla 1 creada a partir de los siguientes textos:

Texto 1 “Hola, cómo estás??”

Texto 2 “Bien gracias, ¿tú cómo estás?”

Diccionario	"hola"	"cómo"	"estás"	"bien"	"gracias"	"tú"	"?"	"¿"	“,”
Texto 1	1	1	1	0	0	0	2	0	1
Texto 2	0	1	1	1	1	1	1	1	1

Tabla 1- Ejemplo de bolsa de palabras

Una de las principales desventajas de representar la frecuencia es que palabras que no aportan valor al texto, conocidas como palabras vacías, pueden ser más recurrentes y sesgar la representación del texto. Lo cual se puede solucionar a través de la eliminación de dichas palabras o a través de la representación utilizando la medida de TF-IDF.

2.5. 3. Frecuencia de Término – Frecuencia Inversa de Documento (TF- IDF)

La medida TF-IDF expresa la relevancia de una palabra dentro de un texto perteneciente a un conjunto de datos. Este valor aumenta proporcionalmente al número de veces que una palabra aparece en un texto y es compensado por la frecuencia de la palabra en el conjunto de datos total, lo que balancea la aparición de palabras generalmente más comunes.

Está compuesto por el producto de dos medidas, frecuencia de término (TF) y frecuencia inversa de documento (IDF). Para calcular la frecuencia de término $TF(t)$ ecuación 1, se obtiene la densidad de la palabra en el conjunto, es decir, el número de veces que el

término *t* ocurre en el conjunto. Donde n es la frecuencia del término y L el número de términos en el conjunto. Mientras que, el logaritmo acota el valor máximo de importancia de palabras que se repiten sustancialmente.

$$TF(t) = \frac{\log_2(n) + 1}{\log_2(L)} \tag{1}$$

En cambio, el $IDF(p)$ Frecuencia inversa de término ecuación 2 es una medida que expresa si la palabra es común o no, en el conjunto de datos. Este calcula la frecuencia de textos con palabras específicas: divide el número total de textos Nt entre el número de textos que contienen la palabra fp, y se toma el logaritmo de ese cociente.

$$IDF(p) = \log(1 + \frac{N_t}{f_p}) \tag{2}$$

A través de esta medida única para cada palabra del texto se puede generar un modelo de representación del corpus completo como se observa en la tabla 2.

Diccionario	"hola"	"cómo"	"estás"	"bien"	"gracias"	"tú"	"?"	"¿"	“,”
Texto 1	0.134	0.274	0.256	0	0	0	.394	0	.3999
Texto 2	0	0.274	0.256	0.013	0.089	0.111	.394	.410	.3999

Tabla 2- Ejemplo de representación de palabras con TD-IDF

2. 6. Preprocesamiento de texto

A través de la literatura se ha demostrado un aumento en la precisión de los modelos de clasificación con ayuda del preprocesamiento de texto. El idioma español, al ser un lenguaje rico en morfología con una gran cantidad de flexiones, pronombres clíticos, diminutivos y aumentativos, ha vuelto su preprocesamiento una tarea difícil que suele obtener resultados diferentes a los de otros idiomas, por lo que algunos han optado por traducir las palabras a otros idiomas para tratarlos, pero los resultados no fueron favorecedores [7]. El primer método de preprocesamiento es la extracción, el cual separa por palabras el texto a analizar; su objetivo es poder implementar técnicas de preprocesamiento a cada palabra. Una vez

separadas las palabras se pueden implementar métodos como remover palabras vacías, lematizar y radicalizar.

2.6. 1. Palabras vacías

Es el nombre que reciben las palabras sin significado como artículos, pronombres, preposiciones, son aquellas que deben ser removidas previo al análisis ya que no tienen significado en la clasificación. Existen diferentes técnicas para lograrlo las más utilizadas en la literatura son:

2.6.1.1 Método clásico: Se basa en remover las palabras vacías comparándolas con otras de listas predefinidas.

2.6.1.2 Métodos Z: Basados en la ley de Zipf: Además del método clásico, se utilizan métodos de creación de palabras vacías basadas en el resto del texto, los cuales son, remover las palabras más frecuentes (TF-High) y remover las que solo aparecen una vez. [33]

2.6. 2. Lematización

Consiste en identificar el lema o raíz de una palabra con su forma canónica. Se busca agrupar los lemas de las palabras para reducir el texto. [33] Por ejemplo, correría, corrí, correremos, el lema o raíz sería correr. Esta técnica podría encontrarse palabras con la misma raíz, pero con diferente significado y estas deberían relacionarse a diferentes lemas. Por otro lado, palabras con diferente raíz, pero mismo significado deberían de ser asignadas al mismo lema. Normalmente se procesa en dos etapas, la primera busca la palabra en su forma flexionada, si no es encontrada, se busca información acerca de su forma canónica, categoría gramatical o alguna otra relación con su contexto [39]. Por lo que este método suele ser más preciso, pero computacionalmente costoso. En [48] se demostró que, al lematizar un conjunto de entrenamiento de textos cortos misóginos en español, aumenta la precisión del clasificador, al contrario de un conjunto de la misma categoría en inglés, donde se reduce la precisión.

2.6. 3. Radicalización

En inglés conocida como “Stemming” es el procedimiento de convertir palabras en raíces. Esta toma una solución más cruda que la lematización al cortar el final de las palabras sin un entendimiento del contexto en el que se utiliza la palabra. Debido a esto la radicalización

puede no regresar una palabra como tal y suele ser menos preciso, otro de sus principales problemas es que puede devolver una raíz muy corta y encontrar relaciones entre palabras que realmente no existen (overstemming) y también puede devolver raíces demasiado extensas que no se logran relacionar (understemming). A pesar de dichos problemas, a través de la literatura ha demostrado obtener buenos resultados. Este método usualmente se puede clasificar en tres grandes grupos, truncamiento, estadísticos y mixtos [34].

2.6.3.1 Truncamiento

como el nombre lo sugiere estos métodos buscan remover sufijos o prefijos de una palabra, el método más sencillo es el Truncate (n) Stemmer, el cual conserva n letras de la palabra y trunca el resto. Otra solución es el S-Stemmer un algoritmo que reduce el plural de las palabras y as convierte en singular, un método relativamente sencillo en inglés pero que llega a ser difícil de trasladar en español. Existen algunos otros como Lovins Stemmer, el cual remueve el sufijo más largo de una palabra y después ajusta dicho lema a una palabra válida. Una extensión de este es Dawson Stemmer, que cubre una lista mucho más completa y organizada por lo que lo vuelve más rápido, pero como desventaja dicha lista se vuelve poco reusable en otros idiomas [34]. Otro de ellos es Porters Stemmer un método muy popular basado en reglas o condiciones donde la palabra se va transformando hasta obtener el lema de ella [35] a través de este método surgió el método "Snowball" el cual permite ser flexible con diferentes lenguajes, es más pesado que el método de Lovins pero produce una mejor reducción de datos. Por último, el método de Paice/Husk Stemmer es un algoritmo iterativo con 120 reglas relacionadas a la última letra de un sufijo [36] va remplazando las palabras en cada iteración por lo que se vuelve un proceso pesado y puede llevar a la sobre lematización.

2.6.3.2 Estadísticos

son aquellos Stemmers basados en técnicas de análisis estadísticos, la mayoría de ellos remueve los prefijos de las palabras después de implementar procedimientos o selección estadística. [34] Uno muy utilizado en la literatura es N-gram Stemmer, un N-gram es un conjunto de n caracteres extraídos de una palabra, la idea detrás de este método es que palabras similares tendrán una gran cantidad de n-grams en común [37,38] por ejemplo la palabra introducción se compondrá de in tr-od-uc-ci-on 5 bigramas o int-rod-ucc ion 4 trigramas etc. El método irá asociando los n gramas parecidos. Su ventaja es que es un

Stemmer de lenguaje independiente y la desventaja es que requiere una gran cantidad de memoria para almacenar y relacionar los n gramas. Otro Stemmer de lenguaje independiente es HMM Stemmer, está basado en un concepto del modelo escondido de Markov HMMs, donde autómatas de estados finitos transición a través de funciones de probabilidad [37]. Su desventaja es que es un método no supervisado y en algunos lenguajes puede llegar a causar sobre lematización. Por último, YASS Stemmer está basado en probabilidad y un corpus previo del lenguaje a lematizar y [37] menciona tiene buenos resultados en lenguajes que por naturaleza contienen sufijos.

2.6.3.3 Métodos Mixtos

Flexivo & Derivacional, se necesita un corpus demasiado extenso donde las variantes se definen dependiendo de la sintaxis del lenguaje [33]. Uno de ellos es Krovetz Stemmer (Kstem) el cual utiliza la propiedad flexiva, es efectivo para remover sufijos que se encuentren en un diccionario [38], por lo que debe ser extenso para ser eficaz. Al igual que el método Xerox, es eficaz debido a que es un corpus grande en inglés, disponible mucho más corto en otros pocos idiomas. Este último está basado en un corpus en ingles que provee un análisis morfológico de las palabras y su forma base, dividido en pronombres, verbos, adjetivos y pronombres nominativos.

2. 7. Aprendizaje automático

El aprendizaje automático es una tarea perteneciente al área de la inteligencia artificial, que proporciona a las computadoras la capacidad de aprender y reconocer patrones, al exponerse a un conjunto de datos. Sucede a través de métodos que infieren un modelo a partir de las categorías en las que se agrupa un corpus, de tal manera que automáticamente pueda categorizar nueva información. Actualmente se utilizan dos algoritmos de aprendizaje automático: supervisado y no supervisado.

Los algoritmos de aprendizaje supervisados son capaces de clasificar al entrenarse con un conjunto de entrenamiento, previamente categorizado. Por otro lado, los algoritmos de aprendizaje no supervisados aprenden del conjunto de datos sin conocer la categoría de la información por lo que buscan encontrar sus características y agruparlos.

Los métodos o algoritmos que realizan el proceso de clasificación, se les denomina clasificadores. Los clasificadores supervisados más populares para la tarea de análisis de sentimiento son Naïve Bayes, modelo de Máxima Entropía, Árbol de decisión potenciado, Bosques aleatorios [8], Máquinas de Vectores de Soporte (SVM), K-Vecinos más cercanos (KNN) [9] y Redes Neuronales Artificiales [52].

2. 8. Naïve Bayes (NB)

Es un clasificador basado en el teorema de Bayes ecuación 3, el cual asume que las características a clasificar son independientes entre sí, este vincula la probabilidad un evento x_i dado otro evento c_j, con la probabilidad de c_j dado x_i.

$$P(x_i|c_j) = \frac{P(c_j|x_i)P(x_i)}{P(c_j)} \tag{3}$$

Dado que la ocurrencia de los términos de un texto importa en la en el análisis de sentimiento, la variante más utilizada en esta área es Multinomial Naïve Bayes.

2.8. 1. Multinomial Naïve Bayes

Debido a que la probabilidad del texto en conjunto no aporta información a la clasificación, la probabilidad del texto se asume como la probabilidad conjunta de los términos que aparecen en él. En el teorema multinomial Naïve Bayes ecuación 4, sean $x_1 \ldots . x_i$ los términos independientes, como las palabras de un texto, y c_j la clase a la que el texto pertenece, al aplicar la definición de probabilidad condicional la probabilidad del modelo se calcula de la siguiente manera:

$$P(x_1, \ldots, x_i|c_j) = P(x_1|c_j)P(x_2|x, c_j) \ldots P(x_i|x_{i-1}, \ldots, x_2, x_1, c_j) \tag{4}$$

Para evitar una estimación costosa de la probabilidad dado que i sea un número grande, este modelo considera que la probabilidad de los términos es independiente entre sí, por lo que la función de probabilidad de que un término x_i pertenezca a un texto de la clase c_j puede ser representada como en la ecuación 5:

$$P(x_i|c_j) = \Pi_1^i P(x_i|c_j) \quad (5)$$

La mayor ventaja de los clasificadores de Bayes es su simplicidad, poca utilización de memoria y tiempo de entrenamiento relativamente bajo. Entre sus desventajas está que ha demostrado no ser tan preciso comparado con Máquinas de Vectores de Soporte [14] y Árboles de Decisión [30].

2. 9. Máquinas de Vectores de Soporte (SVM)

Este método representa los puntos de muestra (datos) en un espacio, separando las clases de dichos puntos en dos espacios separados mediante un hiperplano construido a partir de vectores de soporte, los cuales son ejemplos en el conjunto de datos con los que se define en el hiperplano de separación como máximo margen entre las distintas clases.

Una de las ventajas de este clasificador es que ha demostrado ser eficiente en memoria [17] y versátil a través de la aplicación de diferentes funciones del núcleo para la función de decisión. Dichas funciones de núcleo o Kernel, pueden ayudar a solucionar la desventaja de sobre ajuste del modelo (overfitting), que sucede cuando el número de características es mayor al de las muestras.

2.9. 1. Funciones de núcleo

En muchos problemas de clasificación con datos reales no es posible separar linealmente de forma perfecta, por lo que no existe un hiperplano de separación y no puede obtenerse un máximo margen. Cuando esto sucede, el interés de las máquinas de vectores de soporte es encontrar el hiperplano con el menor error empírico a través de una función de núcleo. Esta es una medida de similitud genérica, que devuelve el resultado del producto punto entre dos vectores realizado en un nuevo espacio dimensional distinto al espacio original en el que se encuentran los vectores. Existen diversos Kernels, algunos de los más utilizados son: Kernel lineal y Kernel Gaussiano (RBF).

2.9.1.1 Kernel lineal

La función de Kernel lineal descrita en la ecuación 6 es utilizada para problemas binarios de clasificación. Como se observa en la ilustración 2, es recomendable utilizar esta función si la separación lineal de los datos es sencilla.

$$k(X_i, X_j) = X_i^T X_j \tag{6}$$

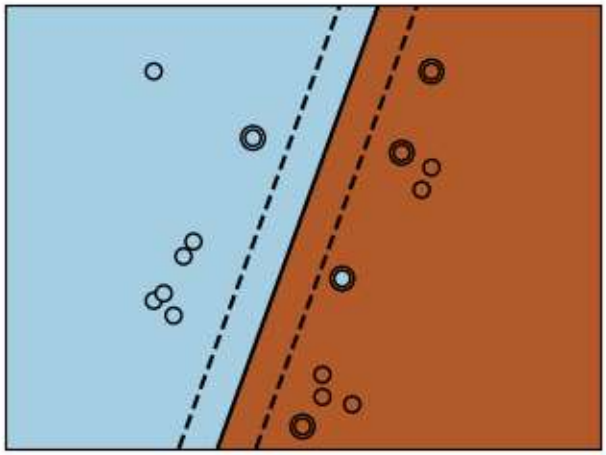

Ilustración 2- ejemplo de hiperplano separado con Kernel lineal [55]

2.9.1.2 Kernel Gaussiano (RBF)

La función de Kernel Gaussiano descrita en la ecuación 7 trata de una función de base radial gaussiana.

$$\mathrm{k}(X_i, X_j) = \exp\left(-\frac{\left|X_i - X_j\right|^2}{2\sigma^2}\right) \tag{7}$$

Siendo $|X_i - X_j|^2$ la distancia euclidiana entre dos vectores y gamma$\gamma = 1/2\sigma^2$un valor escalar que define la influencia de un ejemplo de entrenamiento en el modelo. Lo que lo vuelve muy flexible y puede ir desde un clasificador lineal a uno muy complejo, lo cual se observa en el ejemplo de la ilustración 3. Mientras gamma sea menor existe una mayor

suavidad en la frontera de decisión y mientras sea mayor todos los puntos tienden a ser ortogonales unos a otros y genera sobreajuste.

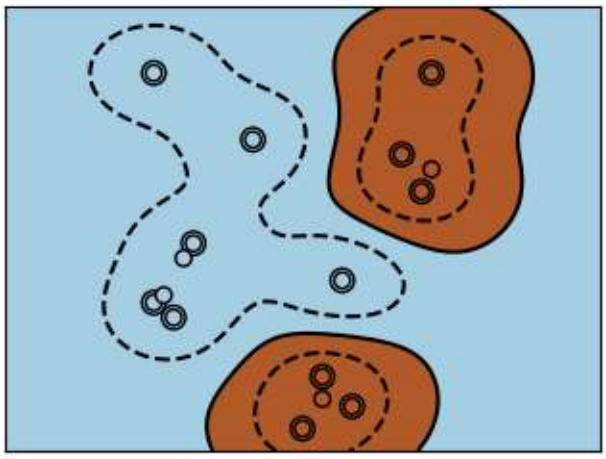

Ilustración 3- Ejemplo de hiperplano separado con Kernel Gaussiano [55]

2. 10. Redes Neuronales Artificiales (RNA)

Tratan de emular el comportamiento del cerebro caracterizado por el aprendizaje a través de la experiencia. [31] las define como, un sistema constituido por un gran número de elementos simples, interconectados, que procesan información por medio de su estado dinámico como respuesta a entradas externas.

Esta red consiste en un conjunto de unidades, llamadas neuronas, conectadas entre sí para transmitir señales. La información de entrada atraviesa la red neuronal sometiéndose a diversas operaciones para producir valores de salida. Se le conoce como topología a la organización de dichas neuronas y está compuesta por el número de capas, la cantidad de neuronas por capa, el grado de conectividad, y el tipo de conexión entre neuronas.

En las redes neuronales, el elemento básico (neurona) es conocida también como el elemento procesador (EP) o perceptrón. Como se observa en la ilustración 4 esta se compone de múltiples entradas x desde 1 hasta n y comúnmente las combina con una suma. La suma de los pesos de las entradas w se modifica por una función de activación f (que puede ser de tipo lineal, umbral o no lineal) y el resultado de esta función se convierte en la salida del elemento procesador y.

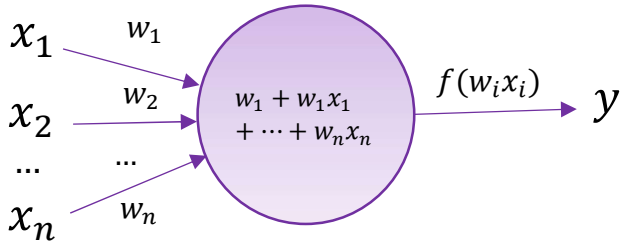

Ilustración 4- Representación de un perceptrón de una Red Neuronal

Al conjunto de neuronas se les conoce como capa, sus entradas provienen de las neuronas de la capa anterior y sus salidas son la entrada de la capa siguiente, a excepción de la primera capa que recibe los datos de entrada y la última capa que produce los valores de salida. Existen redes formadas por una única capa, conocidas como redes mono capa, y las neuronas que la conforman cumplen la función de neuronas de entrada y salida. Cuando la red se compone por dos o más capas se le conoce como red multicapa.

Se le conoce como conexiones laterales o conexiones intra-capa cuando se establece conexión entre dos neuronas de una misma capa. Por otro lado, si la conexión sucede entre neuronas de distintas capas se le denomina conexión intra-capa. Cuando la conexión sucede hacia delante, es decir de la capa de entrada a la de salida se le conoce al modelo como Red Neuronal Prealimentada (Feed-Forward). Al contrario, si la conexión se produce en el sentido inverso al de entrada salida se conoce como Red Neuronal Recurrente (Back-propagation).

Una vez determinada la topología de la red neuronal es necesario entrenarla para que pueda ser utilizada para tareas de clasificación. Durante el entrenamiento, la red es capaz de aprender relaciones mediante el ajuste de los pesos de las conexiones entre neuronas. En este proceso, el modelo se va refinando iterativamente hasta alcanzar un buen nivel de operación para obtener la salida esperada.

2. 11. Validación cruzada

Técnica utilizada para evaluar modelos de aprendizaje de máquina, la cual garantiza que los resultados del modelo son independientes de la división del conjunto de entrenamiento y prueba.

Consiste en dividir el conjunto de datos en K número subconjuntos. Durante K iteraciones se utiliza uno de los subconjuntos como datos de prueba y el resto de los subconjuntos (K-1) se utiliza como un conjunto total para entrenar el modelo de aprendizaje automático. En cada iteración se calculan las medidas de evaluación del modelo. Al finalizar se obtiene la media aritmética de las medidas de evaluación para obtener una métrica única de la cual se puede calcular el margen de error utilizando medidas como: el error cuadrático medio o la desviación estándar.

2. 12. Métricas de evaluación de rendimiento

2.12. 1. Matriz de confusión

Tabla que describe el rendimiento de un modelo supervisado de Aprendizaje de Máquina, se le conoce de esta manera ya que visualiza dónde el modelo está confundiendo las clases. Esta muestra 4 valores.

1. Verdaderos Positivos (TP): Cuando el valor de la clase real era verdadero y la predicción también lo es.
2. Verdaderos Negativos (TN): Cuando el valor de la clase real era falso y la predicción también lo es.
3. Falso Positivo (FP): Cuando el valor de la clase real era falso y el pronosticado es verdadero.
4. Falso Negativo (FN): Cuando el valor de la clase real era verdadero y el pronosticado es falso.

2.12. 2. Sensibilidad

Descrita en la ecuación 8, devuelve la proporción de casos positivos que fueron correctamente identificadas por el algoritmo.

$$TP/(TP + FN) \tag{8}$$

2.12. 3. Precisión

Descrita en la ecuación 9, entrega el número de elementos identificados correctamente como positivo de un total de elementos positivos.

$$TP/(TP + FP) \tag{9}$$

2.12. 4. F1-Score

El valor F1 que se obtiene como resultado en la ecuación 10, se utiliza para combinar las medidas de precisión y sensibilidad en un sólo valor. Se calcula haciendo la media armónica entre la precisión y la exhaustividad.

$$(Sensibilidad * Precisión * 2)/(Sensibilidad + Precisión) \tag{10}$$

Capítulo 3. Estado del Arte

3. 1. Detección de misoginia

A través de la definición de misoginia podemos entender por qué la tarea de detección de este fenómeno se ha vuelto difícil de estudiar. Existen diversos trabajos sobre detección de sexismo que analizan comentarios hacia perfiles de mujeres en plataformas de videojuegos [4] en comentarios de Facebook [5] y otros pocos en Twitter [6]. De la misma manera existen trabajos, en textos cortos, sobre detección de agresividad con corpus de 11 mil tweets divididos entre agresivos y no agresivos [3]. A pesar de los esfuerzos en dichos trabajos, no es posible utilizar dichos conjuntos para detectar misoginia debido a la gran diferencia entre ella, el discurso de odio y agresividad hacia las mujeres y el sexismo.

Recientemente se han llevado a cabo tareas de detección de misoginia en donde se obtuvieron dos grandes conjuntos en español MisoCorpus-2020 [45] con un total de 7682 tweets y un conjunto desarrollado por el Taller de Análisis Semántico en la Sociedad Española para el Procesamiento del Lenguaje Natural en la tarea de Detección Automática de Misoginia [46] que cuenta con 3307 tweets. Lamentablemente ambos conjuntos tienen textos cortos recopilados de diversos países hispanohablantes y no están etiquetados por región lo que impide analizar los prejuicios relacionados a cada país.

3. 2. Preprocesamiento

Existen diversas técnicas utilizadas en preprocesamiento de textos cortos, específicamente tweets, para detección de misoginia en español [49-53], estas son: cambiar las menciones en los tweets por un identificador común (@usuario), transformar todo el texto a minúsculas, eliminar hipervínculos, reducir caracteres duplicados (“agresivooo” se transforma a “agresivo”), corregir errores gramaticales, en la mayoría de dichos trabajos se transformaron los hashtags separándolos por palabras (“#NoMeRepresentan” cambia a “no me representan”) aunque en [49] ignoraron el significado del hashtag reemplazándolos por la palabra HASHTAG y MISO_HASHTAG para aquellos hashtags que se conocían como misóginos, por último, la mayoría de los trabajos busca mapear los emoticones a palabras a través de diversas bibliotecas pero en algunos otros los eliminan [48].

3. 3. Clasificación

Se ha demostrado que no existe una solución única para el aprendizaje de máquina en las tareas de análisis de sentimiento [9, 10] por lo que es necesario analizar los resultados obtenidos a través de la literatura para elegir el que más se adecue a nuestra solución.

A través de la literatura se han propuesto diferentes técnicas y métodos de clasificación, lamentablemente, la mayoría de ellos se enfoca en textos extensos en inglés. A través de ellos se ha demostrado que Maximal Entropy y SVM [12,13,14] obtienen los mejores resultados en corpus extensos y durante los últimos años KNN ha demostrado también ser escalable en problemas de Big Data [15]. Por otro lado, si el corpus a tratar es pequeño, diversos autores [11, 12,16] han demostrado que Naïve Bayes es el más preciso, incluso más que KNN [16] a pesar de ser también un clasificador lineal. De la misma manera, algunas variaciones de SVM han mostrado resultados confiables con corpus pequeños [13,14], así como algunos tipos de Redes Neuronales Artificiales [24-27].

3.3. 1. Multinomial Naïve Bayes

En [19] se utilizó Naïve Bayes como el mejor algoritmo para distinguir entre dos clases alcanzando un 81% de precisión por otro lado también disminuye su precisión al considerar una tercera clase en el conjunto de entrenamiento. En [20] Naïve Bayes alcanzó un 70% de precisión al agregar emoticones al conjunto de datos, también distinguiendo entre dos clases y en [21] comparan Naïve Bayes contra un árbol de decisión C4.5, ambos implementados en Weka, mostrando la diferencia de precisión al agregar emoticones y variar tamaños de corpus. En este trabajo Naïve Bayes obtuvo un 84% de precisión al considerar emoticones y utilizar un corpus pequeño.

Multinomial Naïve Bayes se utilizó en [47] para clasificar misoginia en textos cortos en español, obteniendo 74% de precisión.

3.3. 2. Máquinas de Vectores de Soporte

En [22] se hace una comparación de SVM con corpus en español y en ingles ambos utilizando corpus de más de 1,000 entradas. A pesar de que ambas obtuvieron una precisión de alrededor del 70% el corpus en español obtuvo menor precisión. En [23] se implementó SVM con

Scikit-learn, obteniendo una precisión del 45% hasta 70% entrenando con corpus de temas diversos en inglés. En [45] se alcanzó una precisión de 80% al entrenar un corpus de textos cortos misóginos en español utilizando LS-SVM y una precisión de 81% al entrenar con la variante SMO.

En la literatura se han creado diferentes variantes de SVM, [18] menciona las mejores para clasificación de texto con corpus pequeños: SVMlight, SMO, LASVM, LS-SVM, Proximal SVM y SVM with Uneven Margins.

3.3. 3. Redes Neuronales Artificiales

Existen diferentes tipos de RNA, algunas de ellas han entregado resultados precisos en trabajos de análisis de sentimiento. Una de ellas es Layer-wise Relevance Propagation (LRP) utilizada en [24] con una versión extendida de long short-term memory (LSTM) bidireccional para distinguir sentimiento entre 5 clases en un corpus de críticas de películas, obteniendo una precisión de más del 80%. Otro tipo de RNA es Character to Sentence Convolutional Neural Network (CharSCNN) utilizada en [25] que clasifica el mismo corpus que en [24] con la diferencia que distingue solamente entre 2 clases obteniendo 86% de precisión, este mismo compara los resultados de otros trabajos que han utilizado el mismo corpus con algoritmos como NB y SVM obteniendo una precisión de 82% y 80% respectivamente. Otra RNA es Feedforward Neural Network (FNN) utilizada en [26] entrenando con un corpus de 200 tweets, obteniendo una precisión del 74.15%, se demostró que, al reducir las entradas a 100 tweets la precisión disminuyó a 31%. Por último, la RNA más común en análisis de sentimientos es Dynamic Artificial Neural Network (DAN2), entrenada en [27] con tweets clasificados manualmente entre 4 clases obteniendo una precisión de 65% comparado contra SVM que obtuvo un resultado de 73%. Así mismo, en [28] comparan DAN2 y SVM probando conjuntos de datos de diversos temas y n-grams de distintos tamaños, mencionan la importancia de entrenar un clasificador con clases balanceadas (mismo número de datos). DAN2 obtuvo una precisión de 84% y SVM de 83% en el mejor conjunto de datos y alrededor de 80% en 3 conjuntos restantes. Por último, en [29] utilizan el mismo conjunto de datos que en 6 entrenando también con DAN2 la diferencia está entre el número de clases, probando con 3 y 5 clases no balanceas. Con 3 clases obtiene una precisión de 86% y con 5 clases obtiene 85% a diferencia de SVM que obtiene una precisión de 78% en ambos casos.

La mayoría de los trabajos ha limitado los corpus de entrenamiento a 200 y 300 oraciones, en [26] explican que esto sucedió debido a que era necesaria una gran cantidad de memoria para crear las estructuras de la Red Neuronal. En [27] mencionan que a pesar de tener un corpus de más de 20,000 oraciones la mejor precisión se obtuvo al utilizar 200 oraciones para entrenamiento y en [28] mencionan que es crucial seleccionar los mejores datos y reducir lo más posible el conjunto de entrenamiento. Es importante señalar que todos los trabajos mencionados utilizaron corpus en inglés.

Capítulo 4. Implementación

4. 1. Creación del conjunto de datos

Al generar un corpus de textos cortos misóginos, se busca encontrar textos con agresión verbal directa e indirecta, apología a agresión física o sexual hacia una o varias mujeres y estereotipos que perpetúen la discriminación por género. Los textos cortos misóginos no necesariamente deben incluir palabras agresivas y pueden ser expresados por cualquier persona no importa su género o sexo. La creación del corpus se realizó de manera semi automatizada. Se crearon listas de palabras clave consumidas después por un script, mostrado en la ilustración 5, para hacer búsquedas de tweets que contengan dichas palabras y hayan sido publicados en México. Dichos tweets finalmente fueron clasificados manualmente como misóginos o no misóginos.

```
obtaining tweets from location:  19.436313,-101.991509,300km
keywords:  mujeres AND cocina
obtaining tweets from location:  20.488384,-99.379710,300km
keywords:  mujeres AND cocina
obtaining tweets from location:  20.278533,-98.086248,300km
keywords:  mujeres AND cocina
obtaining tweets from location:  18.306550,-99.703076,300km
keywords:  mujeres AND cocina
obtaining tweets from location:  20.085819,-98.984455,300km
keywords:  mujeres AND cocina
obtaining tweets from location:  19.257448,-100.936631,300km
keywords:  mujeres AND cocina
obtaining tweets from location:  18.327144,-96.668620,300km
keywords:  mujeres AND cocina
obtaining tweets from location:  23.048297,-99.337331,300km
keywords:  mujeres AND cocina
obtaining tweets from location:  24.932877,-105.904010,300km
keywords:  mujeres AND cocina
Saving 3755 tweets at: mujeres_cocina.csv
```

Ilustración 5- Python script con búsquedas de tweets en México por palabras clave

4.1.1. Creación de listas de palabras clave

Se analizaron diversos elementos psicométricos [32] utilizados para evaluar el nivel de machismo, tendencia a la violación, violencia hacia la mujer, comportamiento sexual agresivo, entre otros. A partir de ellos se pudo observar la tendencia en dichas evaluaciones

a utilizar adjetivos agresivos y estereotipos para describir a las mujeres, debido a esto se crearon listas de palabras relacionada a esas dos categorías y se añadió la categoría de refranes mexicanos.

En la tabla 3 se muestran las diferentes listas, la primera se forma de adjetivos y sustantivos femeninos tanto positivos como negativos. La lista creció al modificar adjetivos en superlativo y diminutivo, singular y plural, así como palabras modificadas con errores gramaticales (combinación de v/b, z/s/c) o falta de acentos. En seguida, se creó una lista de palabras clave relacionadas a estereotipos sobre mujeres mexicanas relacionados a diferentes temas tales como: roles de género, sexualidad, belleza, enfermedades mentales, consentimiento, nivel económico, raza, etc. Por último, se creó una lista de conjuntos de palabras relacionadas a refranes mexicanos misóginos; para encontrar una mayor cantidad de textos se redujeron los conjuntos a máximo 3 palabras.

Adjetivos/ sustantivos	Estereotipos	Dichos	Dichos completos
abusada	atención	Aguacates maduran apretón	aguacates y muchachas maduran a puro apretón
abusadora	autismo	arañar negra negro	arañar a una negra, sólo un negro
acomedida	barrer	atole escoba cásate	con la que entienda de atole, escoba y metate, con ella cásate
adaptable	celulitis	bailar vieja burro	bailar con una vieja es lo mismo que andar en burro
adúltera	cerebro	bozo beso sabroso	mujer con bozo, beso sabroso
ágil	chocar	caballo mujer escoger	gallo, caballo y mujer, por la raza has de escoger
agradable	cocina	caballo mujer espuela	al caballo, con la rienda, a la mujer, con la espuela
alborotada	consentimiento	calladita más bonita	calladita te ves más bonita

Tabla 3- Ejemplos de palabras de búsqueda por categoría

4.1.2. Búsqueda de tweets

Con ayuda de la biblioteca Tweepy [50] se desarrolló un programa el cual hace una búsqueda de tweets que contengan una palabra en específico. La búsqueda se repite 32 veces, modificando la latitud y longitud de la búsqueda para cada estado de la república. Se

automatizó la repetición de dicha búsqueda para las aproximadamente 1,500 palabras de las tres listas mencionadas anteriormente. A pesar de tener una extensa cantidad de palabras de búsqueda, Twitter permite al usuario desactivar la ubicación de donde se publican sus tweets, por lo que pocas publicaciones se pueden encontrar a través de la geolocalización. Debido a esto solamente se obtuvieron alrededor de 80,000 tweets sin limpiar. Al eliminar tweets repetidos, con caracteres incorrectos o tweets vacíos, se finalizó con un conjunto limpio de alrededor de 35,000 tweets.

4.1.3. Selección y clasificación de tweets

Al obtener el conjunto de tweets por palabra de búsqueda se seleccionaron máximo 5 tweets misóginos y 5 no misóginos por cada una de ellas, así como también existieron búsquedas que no devolvieron tweets clasificables. Es importante remarcar que al hacer la selección se pudieron diferenciar niveles de misoginia. Se buscó incluir tweets con niveles altos, medios y bajos de misoginia. La selección se realizó de esta manera con el fin de no sesgar la selección hacia una sola palabra o hacia un solo nivel de misoginia. Al finalizar de categorizar se obtuvo un conjunto de 1,100 tweets misóginos y 1,100 no misóginos. Como se observa en la ilustración 6, se almacenó el tweet, lugar de origen y URL único de cada uno de ellos.

	A	B	C	D	E	F	G
1	Tweet	Location	URL				
2	Pobrecita... Me dio pena	Mexico	https://twitter.com/__Jou_/status/1337072287426748418				
3	Al mal caballo espuela, mala mujer palo que d	Mexico	https://twitter.com/_AldaRosas/status/263003417209876480				
4	Y a los 14 la morra toda embarazada XD	Mexico	https://twitter.com/_alex051/status/1381800858724220931				
5	Hay hombres que son peor que viejas mitoter	México	https://twitter.com/_duartecarla/status/1301240441409544194				
6	Pinche morra caca	Chihuahua, México	https://twitter.com/_GrilloCrazy/status/1390757903255195650				
7	Alch ya mátate morra pendeja	Aguascalientes, Ag	https://twitter.com/_Marijose_Puga/status/1199491771190972417				
8	Y luego presumiendo sus patas y manos de ca	Mexico	https://twitter.com/_MazeOfTorment/status/1389653382642737153				
9	@AngysPaolis como sufre la hdsptm.	Monterrey	https://twitter.com/_OsvaldoTrejo/status/1337278330714918912				
10	Quien es? Será que por desnalgada no la reco	México	https://twitter.com/_RickFlag_/status/1396232182750924801				
11	El momento mas critico de algunas mujeres e	Mexico	https://twitter.com/_Szyslak_Moe_/status/1386524614172430341				
12	Ya ven como @i_fernandaa si necesita atenci	Mexico	https://twitter.com/_TioMoe_/status/1392299379629957122				
13	Súper huecas las dos ??	Mexico	https://twitter.com/_uncronopio_/status/1390105552710217740				
14	Entonces nadamas andas de interesada lol qu	Mexico	https://twitter.com/05Jimmyesparza/status/1388750072633036800				
15	alv se ve bien artificial la perra esa	Mexico	https://twitter.com/11_emrodriguez/status/1383234329577332738				

Ilustración 6- Conjunto de datos misógino

4. 2. Preprocesamiento del conjunto de datos

Para realizar el preprocesamiento de datos se tomaron en cuenta diversa técnicas para textos cortos, específicamente para tweets debido a sus características. Se comenzó utilizando la

función TweetTokenizer de la biblioteca NLTK [51] la cual hace una extracción por palabras en cada tweet, la diferencia con otras funciones de extracción es que esta automáticamente identifica características pertenecientes a tweets. Por ejemplo, los hashtags se agrupan como una sola palabra "#YoTeCreo". Funcionando de la misma manera con las menciones, dejando como una sola palabra el identificador @ y el usuario "@JosePerez". Por último, esta función puede identificar también combinaciones de caracteres relacionadas a sentimientos por ejemplo la combinación "<3" está relacionada a amor y "xD" relacionada a humor. Un ejemplo de la extracción con TweetTokenizer es:

"Hola siempre me sentí bien insegura de ser más grande al resto! @ladygaga pero creo que ahora es algo que me gusta mucho de mi <3 :) #AmorPropio"

['Hola', 'siempre', 'me', 'sentí', 'bien', 'insegura', 'de', 'ser', 'mas', 'grande', 'al', 'resto', '!', '@ladygaga', 'pero', 'creo', 'que', 'ahora', 'es', 'algo', 'que', 'me', 'gusta', 'mucho', 'de', 'mi', '<3', ':)','#AmorPropio']

4.2.1. Técnicas por palabra

Al concluir la extracción de palabras se aplicaron diversas técnicas de preprocesamiento por palabra:

a) Eliminar hipervínculos
b) Eliminar signos de puntuación que no estén relacionados a hashtags, usuarios o emociones ("# $ % & ( ) * + - / : ; < = > @ [ ] ^ _ ` { | } ~ . ,)
c) Reducir palabras con caracteres repetidos más de dos veces, a un caracter
d) Convertir todos los caracteres a minúsculas
e) Separar los hashtags por palabras, por ejemplo #BuenDía a Buen Día. Aquellos hashtags sin separación de palabras por mayúsculas se transforman a #hashtag.
f) Transformar nombres de usuarios en menciones a @user
g) Debido a que se hicieron búsquedas con palabras clave con errores ortográficos, se transforman dichas palabras a su forma sin errores.

4.2.2. Palabras vacías

A pesar de que diversos trabajados realizados para detección de misoginia en español [49-53] no realizaron eliminación de palabras vacías, como se observa en la ilustración 7, al analizar las palabras más frecuentes en el corpus destacan en su mayoría pronombres, preposiciones y conjunciones, las cuales son categorías comunes para palabras vacías.

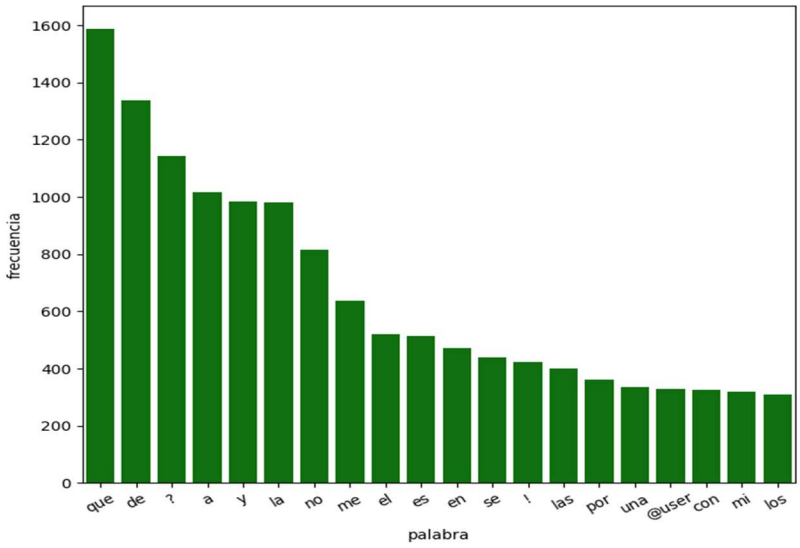

Ilustración 7- Palabras con mayor frecuencia en el corpus

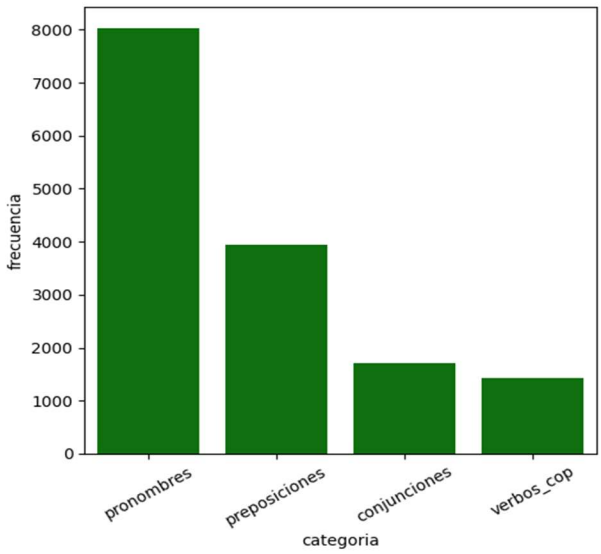

Ilustración 8- Frecuencia de palabras por categoría

Al agruparlas por categoría, se observa en la ilustración 8 una gran diferencia de número de pronombres al resto de las categorías. Una de las razones es porque esta categoría contiene una mayor cantidad de palabras a contar por lo que se categorizó por tipos de pronombres, así como por verbo copulativo.

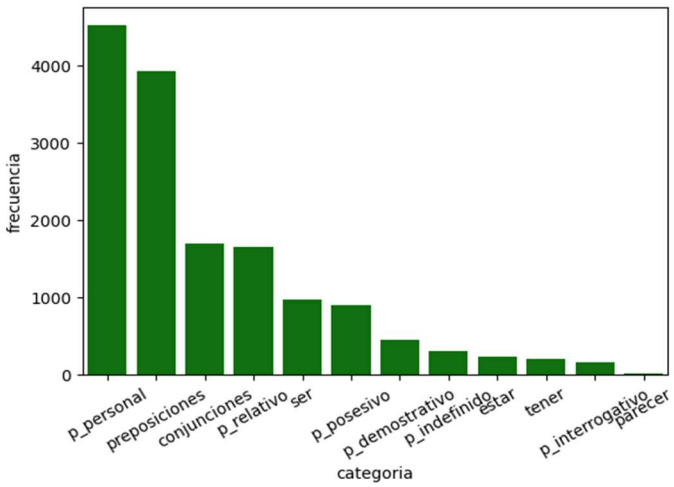

Ilustración 9- Frecuencia de palabras por categoría

A través de este ejercicio en la ilustración 9 se muestra que la categoría más repetida son pronombres personales seguido de preposiciones.

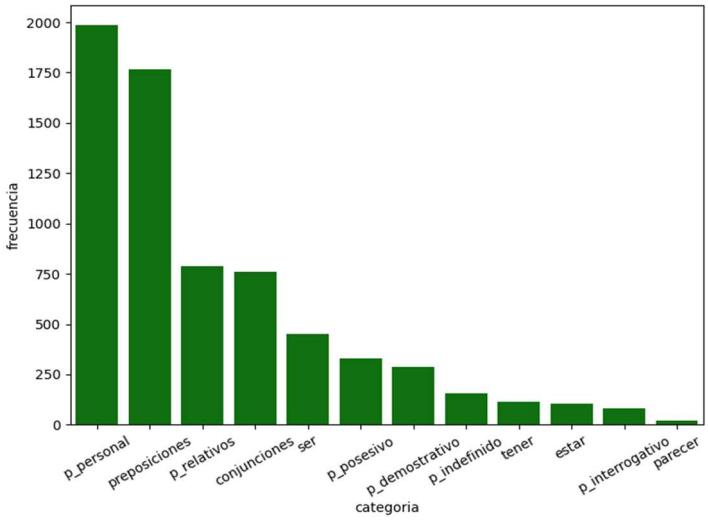

Ilustración 10- Frecuencia de palabras por categoría en corpus misógino

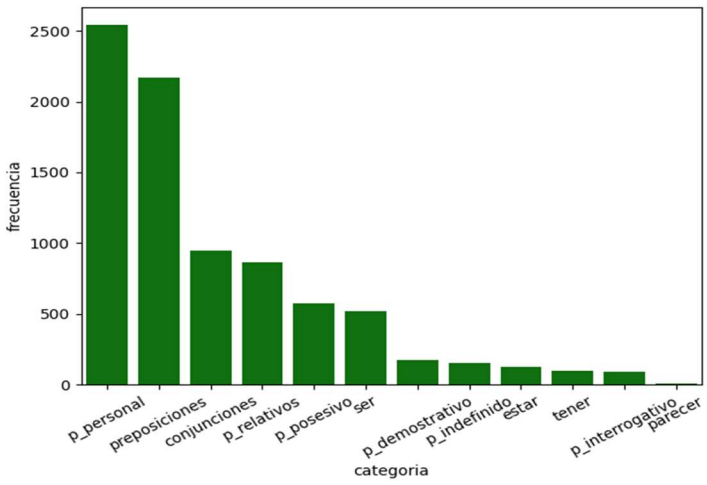

Ilustración 11- Frecuencia de palabras por categoría en corpus no misógino

Al observar las mismas frecuencias en el corpus dividido entre misógino en la ilustración 10 y no misógino en la ilustración 11, se observan tendencias similares, al tener mayor frecuencia en pronombres personales, preposiciones, pronombres relativos y conjunciones. Por ende, se analizaron, las palabras con mayor frecuencia de esas categorías.

Palabras por categoría	**Frecuencia Misógino**	**Frecuencia No Misógino**
Pronombres personales		
La/las	587	692
Me	483	153
El	295	225
Lo/los	354	234
Le	147	122
Conjunciones		
y	551	432
Pronombres relativos		
que	839	750
Preposiciones		
de	735	603
a	572	443
por	177	185
en	275	196
con	173	151

Tabla 4- Palabras más frecuentes por categoría

A pesar de que en la tabla 4 se demuestra que “la”, “las”, “el”, “los”, “me” fueron de los pronombres con más repeticiones, la cantidad de repeticiones entre conjuntos varía y estos están relacionados al género del sujeto por lo que podrían dar valor al clasificador. Al contrario, se”, “que”, “lo”, “de”, “y”, “le”, “a”, “por”, “en”, “con” tuvieron una cantidad parecida de repeticiones y son neutros por lo que podrían ser consideradas palabras vacías.

A través de este análisis se crearon 3 tipos de corpus basados en técnicas de preprocesamiento para tweets y eliminación de palabras vacías:

a) Corpus preprocesado con eliminación total de pronombres, preposiciones, conjunciones, verbos copulativos y palabras que, estadísticamente, son muy comunes a través de la lista de palabras vacías de NLTK [51]
b) Corpus preprocesado con eliminación de pronombres, preposiciones y conjunciones frecuentes en nuestro corpus que no estén relacionadas al género del sujeto
c) Corpus preprocesado sin eliminación de palabras vacías

4.2.3. Lematizar

En Python existen diversas bibliotecas que contienen conjuntos de lemas, lamentablemente sólo dos contienen lemas en español Spacy [52] y Freeling [53]. Se descubrió que Spacy puede cometer múltiples errores con los lemas, menciona tener un 95% de precisión en inglés [54], pero no el porcentaje de precisión en español. Al lematizar todas las formas diversos verbos, se encontró que lematiza de manera incorrecta los verbos flexionados en plural. Por ejemplo, corremos se lematizó a corremo, corríamo a corriamo, correremos a correremo. correríamos lo lematizó a correriar y correrían a correrir. Así como flexiones que terminan con la letra A las tranforma a la letra O como el verbo pegaría lo transforma a pegario. Por otro lado, Freeling, mostró una mejor precisión al observar los mismos ejemplos lematizados de manera correcta y demostró aumentar la precisión de 76% a 79% para clasificación de misoginia en español [48], por lo que se crearon dos corpus lematizados con cada una de las herramientas antes mencionadas.

4.2.4. Radicalizar

A pesar de que esta técnica ha demostrado entregar mejores resultados en textos cortos en inglés que en español [48] creó un corpus con esta técnica para analizar sus resultados. Para radicalizar se utilizó el truncamiento de “Snowball” de la técnica de Porter Stemmer de la biblioteca NLTK [51].

4. 3. Modelos de representación de palabra

4.3.1. Bolsa de Palabras con frecuencia

Se creó un modelo de bolsa de palabras utilizando la función CountVectorizer de la biblioteca SKLearn la cual extrae las palabras de cada texto y crea una colección de palabras a partir del conjunto, a través de esta función se puede obtener una matriz donde las columnas son las palabras de la colección y las filas el conjunto de valores enteros que representan la frecuencia de aparición de las palabras pertenecientes a cada tweet.

4.3.2. Bolsa de Palabras con TF-IDF

Se generó una matriz similar a la bolsa de palabras, sin embargo, se obtuvo el TF-IDF de cada una de las palabras del corpus, a través de la función TfidfVectorizer de la biblioteca sklearn, dicha función realiza los mismos pasos que CountVectorizer pero sustituye el valor entero de frecuencia con el resultado del TF-IDF

4.3.3. Representación de bigramas

Las funciones mencionadas anteriormente TfidfVectorizer y CountVectorizer permiten enviar como parámetro un par de valores a través de ngram_range, este recibe un mínimo a y máximo b de valores para los generar n-gramas, si no se envía ese parámetro el valor default de ngram_range es (1,1) por lo que únicamente se consideran unigramas. Se implementó un valor mínimo de 1 y máximo de 2 por lo que el conjunto se vectorizó entre unigramas y bigramas, permitiéndolo ser flexible para textos cortos.

4. 4. Separación del conjunto de entrenamiento y de prueba

Para la separación del conjunto de datos se utilizó la función train_test_split de la biblioteca sklearn para crear los dos conjuntos de manera aleatoria. Dicha función permite mandar como parámetro un entero como semilla (random_state) para aleatorizar de tal manera que la separación de los subconjuntos de entrenamiento y prueba pueda ser reproducida para todos los modelos. Así mismo, la función permite enviar como parámetro un valor booleano si se desea estratificar la división de subconjuntos, lo cual nos regresa subconjuntos balanceados en clases. Por último, a través de un flotante se puede definir el porcentaje de tamaño deseado para el subconjunto de prueba, con base en el estado del arte se mantuvo una relación de 80% del conjunto para entrenamiento y 20% para el conjunto de prueba.

4. 5. Entrenamiento de los clasificadores

4.5.1. Implementación con SKLearn

Para la implementación de los algoritmos de Multinomial Naïve Bayes y Máquinas de vectores de Soporte se utilizaron los modelos implementados de la biblioteca SKLearn, conocidos como “MultinomialNB” y “SVC”. El modelo SVC recibe como parámetro el nombre del Kernel a utilizar, de los cuales se implementó “linear” y “rbf”.

Después de crear ambos modelos se utiliza la función “fit” para realizar el entramiento del modelo, la cual recibe como parámetro el conjunto con los datos de entrenamiento y el conjunto de etiquetas de clase del conjunto anterior. Por último, al tener un modelo entrenado se puede utilizar la función “predict” que recibe el conjunto de datos a predecir y devuelve un arreglo con las etiquetas de las predicciones realizadas.

4.5.2. Implementación con Keras

Para la implementación de la Red Neuronal Artificial (FFNN) se utilizó el modelo de la biblioteca Keras a través de la función de “Sequential”. Este modelo está definido como una secuencia de capas las cuales se agregan a través de la función “Add” que añade un objeto “Dense” el cual recibe como parámetros las características de la capa tales como número de

nodos por capa, y la función de activación. La primera capa también recibe el número de datos de entrada y el número de palabras del conjunto total.

Se realizaron diversas pruebas con la arquitectura del nodo para conocer los hiper parámetros óptimos para el conjunto de datos. Se encontró que al aumentar el número de capas se reducía la precisión por lo que se diseñó el modelo con 3 capas una de entrada, una capa oculta y una de salida. Del mismo modo, los nodos demostraron una mejor precisión al reducir su número a 32 nodos en la capa de entrada 16 en la oculta y un nodo para la salida, todas con una función de activación sigmoide. El número de iteraciones en las que el aprendizaje recorrerá por completo al conjunto se le conoce como épocas; se evaluó la precisión del modelo al entrenar con diferentes valores para esta. Al asignar números mayores a 1000 el modelo devolvía una precisión menor a 50% y al acercarse a 200 épocas devolvía un porcentaje mayor a 70% en precisión.

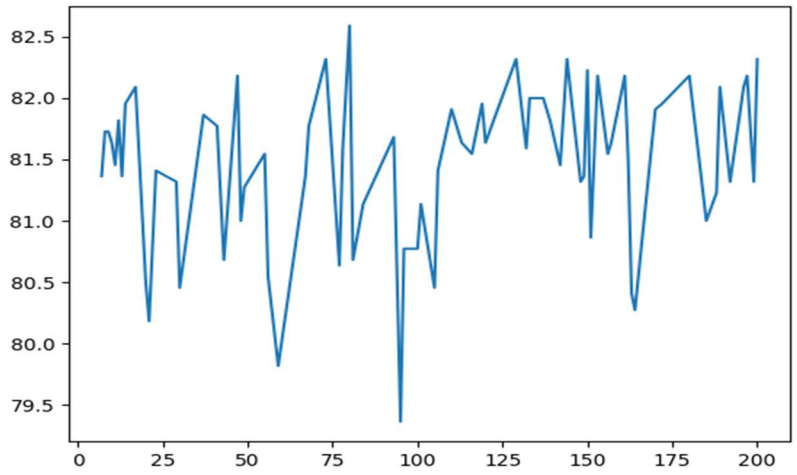

Ilustración 12- Precisión de la RNA entrenada con diferentes épocas

A través de los resultados de este experimento que se observan en la ilustración 12 se demostró que no existe una tendencia en precisión al aumentar el número de épocas. Los valores que dieron los mejores resultados fueron 80, 129, 144, 73 y 200 épocas. Por lo que se seleccionó el valor de 80 para entrenar la red como se muestra en la ilustración 13.

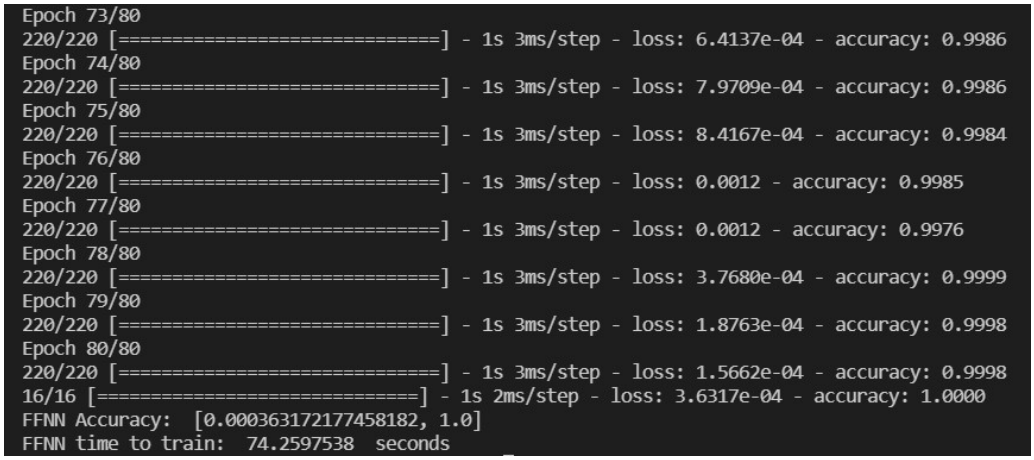

Ilustración 13- Python script de RNA entrenada con 80 épocas

Una vez definida la arquitectura es necesario establecer diversas de métricas para el modelo a través de la función “Compile”. las cuales son la función de pérdida, optimización y las métricas a evaluar del modelo.Para la función de pérdida se utilizó el estimador de error cuadrático medio, su función es calcular el margen de error que un modelo debe minimizar durante el entrenamiento. La función de optimización busca cambiar los atributos del modelo para reducir las pérdidas. Se utilizó el modelo ADAM (Adaptive Moment Estimation) el cual es característico, por utilizar pocos recursos computacionales y funcionar para conjuntos de datos con una gran cantidad de parámetros, descripción que se ajusta a nuestro modelo. Por último, la métrica que se definió para evaluar el modelo fueron las de precisión y pérdida.

A continuación, la función “Fit” se utiliza para entrenar el modelo. Esta recibe el conjunto de datos de entrenamiento y sus etiquetas, así como los hiper parámetros tales como el número de épocas y “batch_size” el cuál es el número de veces que el conjunto se va a dividir durante el entrenamiento, este debe ser mayor a cero y menor que el número de muestras en el conjunto y fue asignado a 10.

4.5.3. Evaluación de los modelos

Se utilizó validación cruzada con K iteraciones para evaluar la precisión de los tres modelos. Para la separación de los subconjuntos se utilizó la función “StratifiedKFold” de la biblioteca SKLearn con un parámetro de K = 10 (ya que demostró una desviación estándar menor a la de K = 5). Este devuelve la separación de los conjuntos estratificados de prueba y

entrenamiento para las 10 iteraciones. Esta técnica fue implementada en MNB y SVM a través de la función "cross_val_score" de la misma biblioteca, la cual devuelve el promedio de la precisión y su desviación estándar. Para la RNA se realizó una función para repetir el entrenamiento de la red y regresar la precisión que devuelve, se entrenó con cada uno de los subconjuntos que devolvió StratifiedKFold y se obtuvo la precisión promedio y desviación estándar de los modelos.

Otras de las métricas a obtener fue f1-score, sensibilidad y precisión, así como la matriz de confusión para analizar las predicciones de los modelos MNB y SVM. Se implementaron a través de las funciones "metrics.classification_report" y "metrics.confusion_matrix" de la biblioteca SKLearn.

Por último, se calculó el tiempo de la transformación de los conjuntos, así como el tiempo de entrenamiento y predicción con los modelos, a través de la función "perf_counter_ns" de la biblioteca "time".

Capítulo 5. Resultados

Se entrenaron 4 modelos con 21 corpus diferentes, generados con la combinación de técnicas de preprocesamiento mencionadas en el capítulo 4.

Al analizar los resultados, 20 de los conjuntos devuelven una precisión mayor a 75% y una desviación estándar menor a 2.90% al realizar validación cruzada estratificada con 10 iteraciones. En la tabla 5, se muestran la precisión P y desviación estándar DE como resultado de 9 conjuntos que al entrenar alcanzan un porcentaje de precisión mayor a 80%.

Corpus	**MNB**		**SVM Linear**		**SVM RBF**		**RNA**	
	P	DE	P	DE	P	DE	P	DE
Bigramas	81.90	1.60	80.90	1.90	79.20	2.80	**82.59**	**1.50**
Unigramas	81.50	1.40	81.50	2.00	80.70	2.10	78.10	1.70
Bigramas, PV	81.50	1.50	81.50	2.50	80.10	2.50	81.50	1.40
Unigramas, PV	81.00	1.10	81.30	1.00	80.60	1.60	78.30	1.50
Bigramas, radicalizar, PV	80.90	1.60	79.50	1.90	78.90	2.20	79.80	1.00
Bigramas, radicalizar	80.50	1.80	80.10	1.00	79.00	2.00	79.80	0.90
Bigramas, lematizar FL, PV	80.00	1.70	79.40	1.40	77.80	2.60	80.50	1.50
Unigramas, PV NLTK	79.30	1.70	80.50	2.20	79.40	1.70	76.50	2.00
Unigramas, radicalizar	79.40	1.50	80.30	1.70	80.00	2.00	74.30	0.70

Tabla 5- Precisión y desviación estándar de modelos entrenados

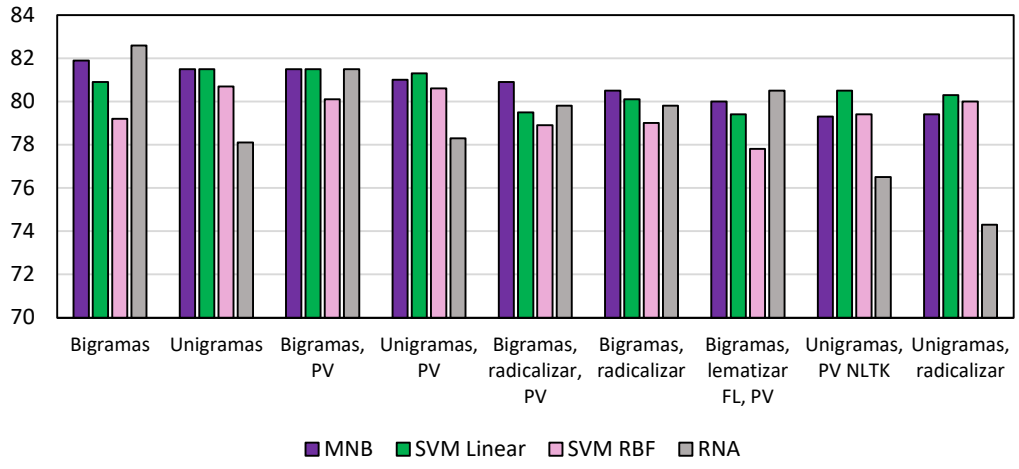

Ilustración 14- Porcentaje de precisión de modelos

A través de la ilustración 14 se observa que la diferencia entre la precisión de los modelos es mínima. RNA devolvió la mejor precisión utilizando bigramas, técnica que demostró aumentar la precisión con cualquier otra combinación de técnicas. A pesar de ser RNA el modelo que devolvió mayor precisión con esa técnica, entrenando con el resto de los corpus no devolvió una precisión mejor que MNB o SVM con Kernel lineal. Aunado a esto RNA tarda aproximadamente 70 segundos en entrenar, 10 veces más que el modelo de MNB que tarda 6 segundos, por lo que podríamos analizar qué tan eficaz es implementar una RNA por la poca diferencia de precisión que devuelve.

Sólo un corpus devolvió una precisión menor al promedio obteniendo entre 54.90% y 56.80% al entrenar con los 4 modelos y una desviación estándar mayor a 18%. Este corpus fue lematizado con la herramienta Spaceling y devolvió dichos resultados al no eliminar palabras vacías mostrado en la Tabla 6.

Corpus	**MNB**		**SVM Linear**		**SVM RBF**		**RNA**	
	P	**DE**	**P**	**DE**	**P**	**DE**	**P**	**DE**
Unigramas, lematizar Spaceling	59.70	18.80	56.80	18.70	58.60	**19.90**	**54.90**	15.50
Unigramas, lematizar Spaceling, Palabras vacías	78.30	1.30	78.30	2.10	78.10	2.40	77.20	2.50
Unigramas, lematizar Spaceling, Palabras vacías NLTK	77.20	2.60	76.50	2.20	77.80	1.30	74.20	1.90

Tabla 6- Resultados de Corpus lematizados con Spaceling

Como se observa en la ilustración 15, a diferencia de los corpus lematizados con la misma herramienta, pero sin palabras vacías tanto del conjunto fijo como de NLTK aumentaron su precisión a más de 76%. A través de esto podemos demostrar la falta de efectividad de esta herramienta para lematizar corpus en español con palabras vacías. En cambio, utilizando otras herramientas se puede observar que se obtuvo buena precisión tanto al implementar radicalización como lematización, con una diferencia mínima al utilizar ambas técnicas.

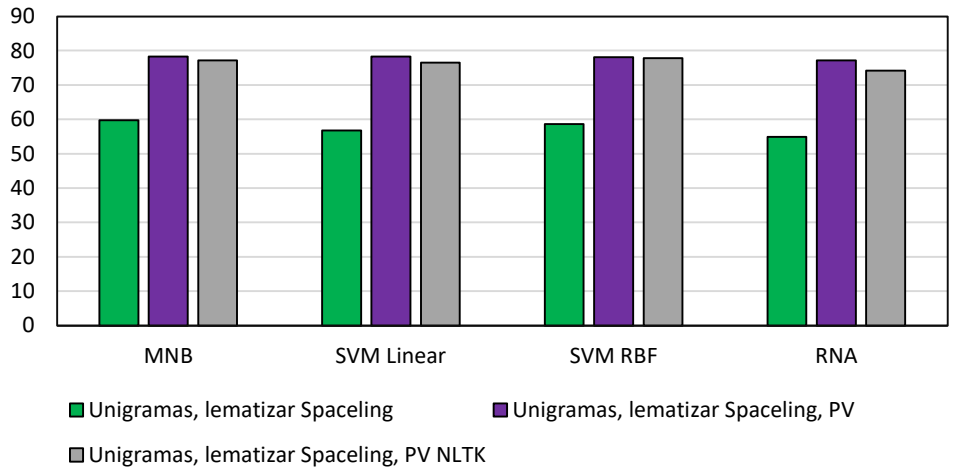

Ilustración 15- Precisión de modelos

Por otro lado, a través de las matrices de confusión, en la ilustración 16 se puede observar que SVM devuelve un f1-score mayor en la clase de no misóginos que los misóginos a diferencia de MNB en la ilustración 17.

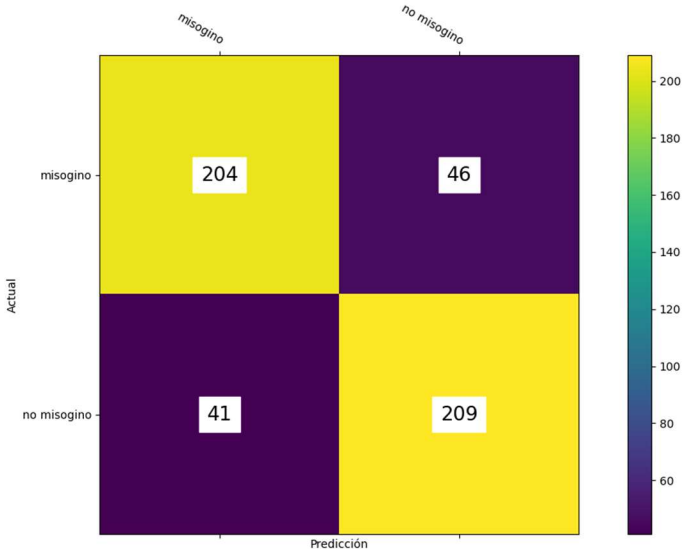

Ilustración 16- Matriz de confusión SVM con corpus unigramas

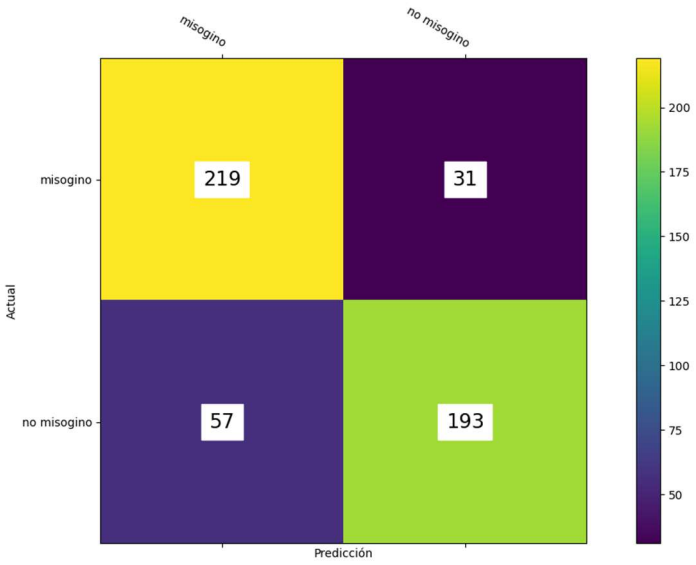

Ilustración 17- Matriz de confusión MNB con corpus unigramas

Se utilizó la herramienta Voyant-tools [56] para analizar las palabras más frecuentes de ambos corpus sin palabras vacías, con este análisis se puede observar en la ilustración 18 una mayor cantidad de palabras agresivas en el corpus misógino. De la misma manera dichas palabras tienen una frecuencia mayor, a diferencia del corpus no misógino en la ilustración 19 que contiene una mayor cantidad de palabras en general y una menor frecuencia de palabras agresivas.

Ilustración 18- Nube de palabras más frecuentes de corpus misógino

Ilustración 19- Nube de palabras más frecuentes de corpus no misógino

En el corpus misógino, la palabra “pinche” se repite 107 veces y dentro de los bigramas se encuentra frecuentemente junto a las palabras “vieja”, “no”, “prieta”, “puta” y “tu”. A diferencia del corpus no misógino donde la misma palabra se repite 61 veces y está relacionada a “me”, “no”, “encanta” y “@user”. Otro ejemplo es la palabra “vieja” en el corpus misógino se relaciona a “no”, “puta”, “me”, “te” y “pendeja” y en el corpus no misógino dicha palabra es menos frecuente y se acompaña de las palabras “gusta”, “me”, “agárrenme” y “jalo”. Teniendo en cuenta que el modelo está basado en la probabilidad de repetición de palabras se puede entender porque MNB es capaz de detectar mejores verdaderos positivos cuando el corpus a predecir contiene una gran cantidad de palabras agresivas repetidas, a diferencia de los ejemplos que se muestran en la tabla 7:

debes ser la típica niña chismosa que nadie quería en el salón y se refugia en los libros porque no tienes amigos
feminista y se auto cosifica vendiendo sus packs en only fans es neta
enseñando las nalgas lo dicen como que saliera desnuda y lo bueno es que nunca he tenido que enseñar mi cuerpo para llamar la atención de alguien pq soy más que eso
jajaja no mames nicole porque les encanta hacer famosas a esas naquitas que no tuvieron una figura paternal sólida

Tabla 7- Verdadero Positivo SVM y Falso Negativo en MNB

Podemos observar que MNB devuelve falsos negativos en textos que no tienen o tienen poca cantidad de palabras explícitamente agresivas, al contrario de SVM que es capaz de aprender más allá de la repetición de palabras.

por qué no eres como las demás niñas ? cómo me saco esas palabras de la cabeza ?¡
pinche semana pesada
calladita no te ves más bonita
pónganse a barrer y a trapear bien crudos y van a ver cómo se las pela cualquier mamador que hace crossfit
jajajaja ! una refinería tonta ? ? ? si y con deuda ! ! infórmense ! por eso les ven la cara !
qué le van a dar a sus mamás de regalo díganme ya

Tabla 8- Falso positivo SVM y MNB

Por otro lado, en la tabla 8 se observa que los casos de falso positivo tanto en SVM como en MNB pueden ser causados por incluir textos con palabras que existen en el corpus del aprendizaje. Al tener "calladita te ves más bonita" el clasificador no fue capaz de discernir como no misógino el texto "calladita no te ves más bonita". Lo contrario se observa en la tabla 9 con falsos negativos en ambos modelos cuando el texto no contiene suficientes palabras conocidas como misóginas en el corpus de entrenamiento.

creo no me explique bien yo no la sigo no me gustan los circos ella como muchas mujeres no se dan a respetar
creo que las mujeres son mucho más hipócritas que los hombres en temas como ese no se si estoy equivocado pero realmente lo pienso
andan denunciando a todo el mundo de violador mientras cantan tú quiere que sea chivirika putona papi ven y detona que tenga un chapón te encanta este culo ya estoy dando para papi deja tu emoción ? ? tantita madre

Tabla 9- Falso negativo SVM y MNB

Capítulo 6. Conclusiones y trabajo futuro

En este trabajo se entrenaron diversos clasificadores supervisados para textos cortos en español capaces de discernir tweets misóginos de no misóginos provenientes de usuarios ubicados en México. Se comenzó realizando un análisis del estado del arte sobre clasificación de sentimientos en español e inglés específicamente en tareas de detección de agresión y misoginia. Se puede observar una gran diferencia en la cantidad de corpus en inglés contra otros idiomas como el español, tanto en temas generales como en temas específicamente misóginos, agresivos o sexistas. Actualmente solo existen dos trabajos de detección de misoginia con conjuntos etiquetados en español. Sin embargo, los dos conjuntos contienen textos de múltiples países hispano hablantes, no definen si están balanceados entre países de origen y tampoco están disponibles para trabajar con ellos. Por lo que se creó un conjunto de datos construido a partir de textos cortos misóginos en español de México.

La creación semi automatizada de dicho corpus generó múltiples problemas. Desde la obtención de tweets, ya que la mayoría de los usuarios de Twitter en México no muestran la ubicación donde se generó el contenido, obteniendo pocos textos al delimitar la búsqueda por ubicación. Otro problema fue el tiempo de limpieza de los textos cortos, ya que muchos eran casi idénticos y fue necesario distinguirlos manualmente. Por último, el etiquetado de manera manual realizado por una sola persona puede generar sesgo. A pesar de que dicho etiquetado se basó en elementos psicométricos utilizados para evaluar el nivel de machismo, tendencia a la violación, violencia hacia la mujer, comportamiento sexual agresivo y otros, se vuelve complicado discernir misoginia en textos que no son explícitamente agresivos.

Al realizar las tareas pertenecientes a la detección de misoginia con dicho corpus, se encontraron diferentes problemáticas en los trabajos con corpus de textos en español de distintos países. Comenzando con el preprocesamiento y modelado de palabras, dado que la diversidad lingüística puede provocar que la semántica del texto sea muy diferente en palabras similares; lo que lleva a tener una mayor cantidad de palabras y menor frecuencia de ellas, por lo tanto, puede disminuir la precisión del clasificador. Otra desventaja sucede con las herramientas de preprocesamiento. A pesar de que existen algunas que implementan técnicas en español, estas tampoco distinguen palabras entre países hispanohablantes

teniendo muy pocas palabras específicas a un país, disminuyendo la eficacia de sus técnicas. Finalmente, la diversidad entre países vuelve muy difícil el análisis de sentimiento, especialmente la clasificación de misoginia, sentimiento que está relacionado a los roles sociales del lugar donde se expresa. Dichas desventajas conllevan a la falta de resultados de clasificadores, con los cuales comparar la clasificación con corpus en español específicamente de México.

Por último, se clasificó el conjunto de tweets con ubicación en México para realizar un análisis del desempeño obtenido a partir de cada técnica de preprocesamiento, obteniendo la mejor precisión de 82.59% utilizando Redes Neuronales Artificiales con técnica de bigramas. Esta métrica supera la mejor precisión de 79.44% en detección de misoginia en español en [48] y 77.06% en [45]. A pesar de que el modelo de Red Neuronal Artificial entregó la mejor precisión, en este documento se demuestra que Multinomial Naïve Bayes es un modelo de bajo costo que devuelve resultados ligeramente mejores que dicho modelo y que Maquinas de Vectores de Soporte al tratar de clasificar conjuntos con palabras explícitamente agresivas. Por otro lado, se encontró que SVM es mejor para detectar contextos misóginos, no solo agresión explícita, por lo que entrenar un clasificador con SVM y un conjunto mucho más grande con textos misóginos sin palabras agresivas podría mejorar la precisión clasificador.

En cuanto a las técnicas de preprocesamiento, los bigramas demostraron aumentar la precisión sobre todos los corpus con unigramas lo que demuestra la importancia de utilizar modelos de palabras que resuelvan el problema de relación de palabras en tareas de detección de misoginia. Esto también está demostrado en trabajos como [45] donde la precisión mejora al utilizar dichos modelos como incrustaciones de palabras. Por último, se demuestra la falta de herramientas adecuadas para preprocesamiento de textos en español y el impacto que esto genera en la precisión de los clasificadores.

A pesar de que la tarea de detección automática de misoginia es reciente y parece una tarea difícil de realizar por una máquina, en este trabajo se demostró que, al evaluar patrones de misoginia en el lenguaje tomando en cuenta el contexto social de un país en específico, un clasificador supervisado puede devolver un valor alto de precisión. Crear sistemas que ayuden a visibilizar la violencia en espacios virtuales es el siguiente paso para construir espacios seguros e inclusivos para diversas comunidades.

Bibliografía

[1] Instituto Nacional de Estadística y Geografía. (2019, noviembre). *Estadísticas a propósito del día internacional de la eliminación de la violencia contra la mujer*. Recuperado el 20 de enero de 2021 de https://www.inegi.org.mx/contenidos/saladeprensa/aproposito/2019/Violencia2019_Nal.pdf

[2] Sistema Nacional de Seguridad Pública. (2020, agosto). *Información sobre violencia contra las mujeres Incidencia delictiva y llamadas de emergencia 9-1-1.* Recuperado el 20 de noviembre de 2020 de https://drive.google.com/file/d/1IAN68eTY8ZoXuoJ0-W4uTpyZwbiS913H/view

[3] Álvarez-Carmona, M. Á., Guzmán-Falcón, E., Montes-y-Gómez, M., Escalante, H. J., Villasenor-Pineda, L., Reyes-Meza, V., & Rico-Sulayes, A. (2018, septiembre). Overview of MEX-A3T at IberEval 2018: Authorship and aggressiveness analysis in Mexican Spanish tweets. *Notebook Papers of 3rd SEPLN Workshop on Evaluation of Human Language Technologies for Iberian Languages (IBEREVAL),* Seville, Spain, 2150, 74–96. http://ceur-ws.org/Vol-2150/overview-mex-a3t.pdf

[4] Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., & Vakali, A. (2017, abril). Measuring# gamergate: A tale of hate, sexism, and bullying. In *Proceedings of the 26th international conference on world wide web companion*, 1285–1290. https://doi.org/10.1145/3041021.3053890

[5] Druzhynin, O. O., Nekhai, V. V., & Prila, O. A. (2019). Facebook text posts classification with tensorflor. In *Mathematical Machines and Systems*, 3, 47–54. https://doi.org/10.34121/1028-9763-2019-3-47-54

[6] Nozza, D., Volpetti, C., & Fersini, E. (2019, octubre). Unintended bias in misogyny detection. In *IEEE/WIC/ACM International Conference on Web Intelligence*, 149–155.

[7] Brooke, J., Tofiloski, M., & Taboada, M. (2009, septiembre). Cross-linguistic sentiment analysis: From English to Spanish. In *Proceedings of the international conference RANLP-2009*, 50–54.

[8] Enríquez, F., Cruz, F. L., Ortega, F. J., Vallejo, C. G., & Troyano, J. A. (2013). A comparative study of classifier combination applied to NLP tasks. In *Information Fusion*, 14(3), 255-267.

[9] Gamallo, P., Garcia, M., & Fernández-Lanza, S. (2013, September). TASS: A Naive-Bayes strategy for sentiment analysis on Spanish tweets. In *Workshop on Sentiment Analysis at SEPLN (TASS2013)*, 126–132.

[10] Anta, A. F., Chiroque, L. N., Morere, P., & Santos, A. (2013). Sentiment analysis and topic detection of Spanish tweets: A comparative study of of NLP techniques. Procesamiento del lenguaje natural, 50, 45–52

[11] Catal, C., & Diri, B. (2009). Investigating the effect of dataset size, metrics sets, and feature selection techniques on software fault prediction problem. In Information Sciences, 179(8), 1040–1058.

[12] Manning, C. D., Schütze, H., & Raghavan, P. (2008). *Introduction to information retrieval*. Cambridge university press.

[13] Li, Y., Bontcheva, K., & Cunningham, H. (2004, September). SVM based learning system for information extraction. In *International Workshop on Deterministic and Statistical Methods in Machine Learning*, Berlin, Heidelberg, 319–339.

[14] Forman, G., & Cohen, I. (2004, September). Learning from little: Comparison of classifiers given little training. In *European Conference on Principles of Data Mining and Knowledge Discovery*, Berlin, Heidelberg, 161–172.

[15] Deng, Z., Zhu, X., Cheng, D., Zong, M., & Zhang, S. (2016). Efficient kNN classification algorithm for big data. *Neurocomputing*, *195*, 143–148.

[16] Sharma, A., & Dey, S. (2012, October). A comparative study of feature selection and machine learning techniques for sentiment analysis. In *Proceedings of the 2012 ACM research in applied computation symposium*, 1–7.

[17] Bhavsar, H., & Ganatra, A. (2012). Variations of support vector machine classification technique: a survey. In *International Journal of Advanced Computer Research*, *2*(6), 230–236.

[18] Kumar, M. A., & Gopal, M. (2010, febrero). An investigation on linear SVM and its variants for text categorization. In *2010 Second International Conference on Machine Learning and Computing*, 27–31. doi: 10.1109/ICMLC.2010.64

[19] Go, A., Huang, L., & Bhayani, R. (2009). Twitter sentiment analysis. *Entropy*,17, 252.

[20] Read, J. (2005, June). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL student research workshop*, 43–48.

[21] Wakade, S., Shekar, C., Liszka, K. J., & Chan, C. C. (2012). Text mining for sentiment analysis of Twitter data. In *Proceedings of the International Conference on Information and Knowledge Engineering (IKE),*1–6.

[22] Brooke, J., Tofiloski, M., & Taboada, M. (2009, septiembre). Cross-linguistic sentiment analysis: From English to Spanish. In *Proceedings of the international conference RANLP-2009*, 50–54.

[23] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O. & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *In Journal of Machine Learning research*, 12, 2825-2830.

[24] Arras, L., Montavon, G., K, M., & Samek, W. (2017, septiembre). Explaining Recurrent Neural Network Predictions in Sentiment Analysis. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 159–168.

[25] Dos Santos, C., & Gatti, M. (2014, August). Deep Convolutional Neural Networks for Sentiment Analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 69–78.

[26] Duncan, B., & Zhang, Y. (2015, julio). Neural Networks for Sentiment Analysis on Twitter. In *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, 275–278.

[27] Ghiassi, M., Skinner, J., & Zimbra, D. (2013). Twitter brand Sentiment Analysis: A Hybrid System using N-gram Analysis and Dynamic Artificial Neural Network. In *Expert Systems with applications*, *40*(16), 6266–6282.

[28] Ghiassi, M., & Lee, S. (2018). A Domain Transferable Lexicon set for Twitter Sentiment Analysis using a Supervised Machine Learning Approach. In *Expert Systems with Applications*, *106*, 197–216.

[29] Zimbra, D., Ghiassi, M., & Lee, S. (2016, enero). Brand-related Twitter Sentiment Analysis using Feature Engineering and the Dynamic Architecture for Artificial Neural Networks. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 1930–1938.

[30] Gupte, A., Joshi, S., Gadgul, P., Kadam, A., & Gupte, A. (2014). Comparative study of Classification Algorithms used in Sentiment Analysis. In *International Journal of Computer Science and Information Technologies*, *5*(5), 6261–6264.

[31] Pang, B., Lee, L., & Vaithyanathan, S. (2002, julio). Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, 79–86. https://doi.org/10.3115/1118693.1118704

[32] Fisher, T. D., Davis, C. M., & Yarber, W. L. (2011). Handbook of Sexuality related Measures. *Routledge*.

[33] Vijayarani, S., Ilamathi, M. J., & Nithya, M. (2015). Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, *5*(1), 7-16.

[34] Sharma, D., & Cse, M. (2012). Stemming algorithms: a comparative study and their analysis. *International Journal of Applied Information Systems*, *4*(3), 7-12.

[35] Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, *14*(3), 130-137.

[36] Paice, C. D. (1990, noviembre). Another Stemmer. In *ACM Sigir Forum*, 24(3), 56-61.

[37] Melucci, M., & Orio, N. (2003, noviembre). A novel method for Stemmer generation based on Hidden Markov Models. In *Proceedings of the twelfth international conference on Information and knowledge management*,131-138.

[38] Krovetz, R. (2000). Viewing Morphology as an Inference Process. In *Artificial intelligence*, *118*(1-2), 277-294.

[39] Hernández-Figueroa, Z., Carreras-Riudavets, F. J., & Rodríguez-Rodríguez, G. (2013). Automatic syllabification for Spanish using Lemmatization and Derivation to solve the prefix's prominence issue. *Expert systems with applications*, *40*(17), 7122-7131.

[40] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using Machine Learning techniques. *arXiv preprint cs/0205070*.

[41] Dave, K., Lawrence, S., & Pennock, D. M. (2003, mayo). Mining the peanut gallery: Opinion Extraction and Semantic Classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web* (pp. 519-528).

[42] Manne, K. (2017). Down Girl: The Logic of Misogyny. Oxford University Press.

[43] Johnson, A. G. (2000). *The Blackwell Dictionary of Sociology*. Wiley.

[44] Flood, Michael (18 de julio de 2007). International encyclopedia of men and masculinities. ISBN 978-0-415-33343-6.

[45] García-Díaz, J. A., Cánovas-García, M., Colomo-Palacios, R., & Valencia-García, R. (2021). Detecting misogyny in Spanish tweets. An approach based on Linguistics Features and Word Embeddings. *Future Generation Computer Systems*, *114*, 506-518.

[46] Rosso, P., Gonzalo, J., Martínez, R., Soto, M., & Carrillo-de-Albornoz, J. (Eds.). (2018). *IberEval Evaluation of Human Language Technologies for Iberian Languages Workshop 2018,* 2150. http://ceur-ws.org/Vol-2150/

[47] Molina-González, M. D., del Arco, F. M. P., Martín-Valdivia, M. T., & López, L. A. U. (2019). Ensemble Learning to Detect Aggressiveness in Mexican Spanish Tweets. In *IberLEF@ SEPLN*, 495-501.

[48] Frenda, S., & Bilal, G. (2018). Exploration of Misogyny in Spanish and English tweets. In *Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018)*, *2150*, 260-267. Ceur Workshop Proceedings.

[49] Shushkevich, E., & Cardiff, J. (2019). Automatic misogyny detection in social media: A survey. *Computación y Sistemas*, *23*(4).

[50] Roesslein, J. (2020). *Tweepy* (4.0) [Twitter for Python]. https://github.com/Tweepy/Tweepy

[51] Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O' Reilly Media, Inc."

[52] Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*.

[53] Padró, L., Stanilovsky E. (mayo, 2012). FreeLing 3.0: Towards Wider Multilinguality. *Proceedings of the Language Resources and Evaluation Conference*.

[54] *Facts & Figures · spaCy Usage Documentation*. (s. f.). SpaCy Usage Documentation.

Recuperado 24 de julio de 2021, de https://spacy.io/usage/facts-figures/

[55] Varoquaux, G. & Scikit Learn. (2007). *Kernels* [Illustration]. https://scikitlearn.org/stable/auto_examples/svm/plot_svm_kernels.html

[56] Sinclair, S., & Rockwell, G. (2021). *Voyant-Tools* (2.4) [Web-based reading and analysis environment for digital texts]. https://voyant-tools.org/