Detección de sexismo

Alvarado A. Morán Óscar
OscarAlvarado@ciencias.unam.mx

Dante Bermúdez Marbán
bermudezmarbandante@gmail.com

Resumen

En el presente reporte se muestra la metodología y resultados de detección de sexismo en redes sociales mediante algoritmos de aprendizaje máquina y de aprendizaje profundo para los idiomas inglés y español indiscriminadamente. Los algoritmos que presentan mejores resultados son la regresión logística y el bosque aleatorio, con una exactitud de hasta 0.71 para cada uno de éstos.

1 Introducción

El diccionario de Oxford define sexismo como un prejuicio, estereotipo o discriminación basado en el sexo, típicamente en contra de las mujeres (of Oxford, 2021).

El sexismo nunca ha dejado de existir, y más aún, con el uso de redes sociales se ha propagado el fácil uso de este, situación que afecta a las mujeres en diferentes facetas de su vida y que puede llevar a consecuencias tan graves como la violencia física o hasta la muerte. Es por esto que a principios del año 2021 se lanzó *sEXism Identification in Social netTworks* (abreviado como EXIST), una competencia en donde se tienen que identificar si algunos tweets o posts de una red social llamada *GAB* son sexistas (IberLEF-2021, 2021).

Para la competencia se liberaron datos de entrenamiento y prueba (que se explican más a detalle a continuación), sin embargo, sólo se tenía el etiquetado de los datos de entrenamiento ya que con los datos de prueba se obtenían resultados que se tenían que subir directamente a la página para su evaluación. En este reporte se muestra la implementación de algunos modelos de aprendizaje máquina y aprendizaje profundo para la clasificación de los datos entre sexistas y no sexistas.

1.1 Descripción del corpus

En la página de la competencia se describe que el objetivo del conjunto de datos era cubrir el fenómeno del sexismo en un sentido amplio, por lo que se recolectaron varias expresiones y términos populares, tanto en inglés como en español, que se suelen usar para subestimar el rol de la mujer en la sociedad. Estas expresiones y términos se extrajeron de cuentas de Twitter que recolectan dichas frases que las mujeres reciben diario. El contenido recolectado fue analizado y filtrado por dos personas expertas en desigualdad de género.

Cada tweet fue etiquetado por 5 anotadores de crowdsourcing, siguiendo los lineamientos desarrollados por las dos personas expertas mencionadas previamente. Las etiquetas era elegidas por mayoría de votos entre los anotadores, aunque si había un caso de 3vs2, un hombre y una mujer, ambos expertos en analizar contenido sexista en redes sociales, decidían.

El conjunto de datos consiste de 11,345 textos, cada uno asociado a una fuente, lenguaje y las respectivas etiquetas para las dos tareas.

En la figura 1 se puede apreciar que el conjunto de datos se encuentra aproximadamente balanceado en términos de las etiquetas. Se tienen 3600 textos no sexistas y 3377 textos sexistas.

En la figura 2, podemos ver las fuentes de donde provienen los textos para cada conjunto. Vemos que en el de entrenamiento, los 6977 son tweets, mientras que en el de prueba, son 3386 tweets y 492 gabs. La idea de agregar textos gabs es medir la diferencia entre una red social que controla el contenido (Twitter) y otra que no (Gab).

Como se mencionó, los textos vienen tanto en inglés como en español. En la figura 3 se puede ver que el conjunto de datos también se encuentra balanceado en términos de idioma.

2 Metodología

Se dividió el conjunto de textos etiquetados, de tal manera que 80% era para entrenamiento y 20%

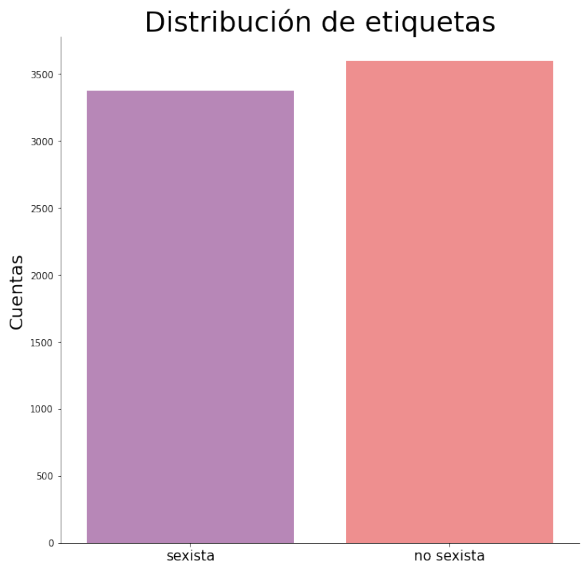

Figura 1: Distribución de las etiquetas para la tarea 1

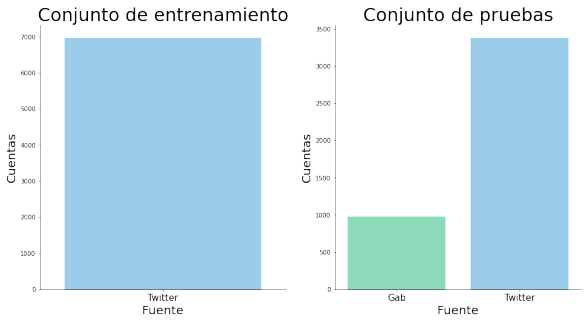

Figura 2: Distribución de las fuentes de origen del texto

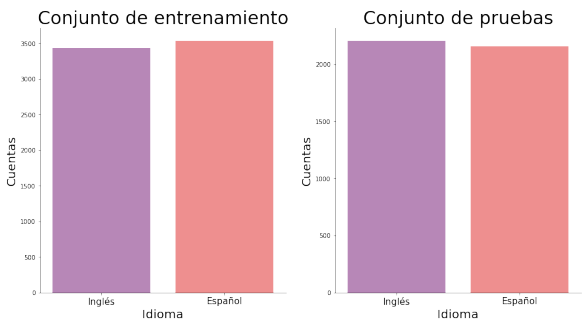

Figura 3: Distribución de los idiomas del texto

para pruebas.

2.1 Preprocesamiento

Antes de aplicar algún modelo, se realizó un procesamiento de los tweets con la finalidad de reducir el vocabulario.

1. Todo el texto se pasó a minúsculas.
2. Los usuarios fueron convertidos a `@usuario`.
3. Todos los enlaces fueron sustituidos por `<link>`.
4. Los números y fracciones como "1/2kilo" fueron sustituidos por `<número>`.
5. Se utilizó un tokenizador de tweets del módulo NLTK, el cual tiene la peculiaridad de reducir secuencias largas de un mismo símbolo, a una secuencia de tamaño tres, por ejemplo, "hooooooooolaaaaa" se convierte en "hooolaaa". Con base en esto, se colapsó esta repetición de tres elementos a un solo símbolo. Si continuamos el ejemplo, el resultado final sería "hola".
6. Finalmente, se quitaron algunos símbolos no deseados, principalmente de puntuación.

2.2 Modelos de aprendizaje máquina

Se probaron modelos de aprendizaje máquina clásicos, por lo que era necesario encontrar características en la matriz término-documento. Para este caso, se utilizó un tf-idf convencional, y se descartaron palabras que no aparecieran más de una vez, es decir, la frecuencia de documento mínimo de una palabra tenía que ser mayor a uno para ser considerada como característica.

Para cada modelo de aprendizaje máquina clásico, se realizó una búsqueda de hiperparámetros por medio de búsqueda aleatoria, esto es probar una cantidad de combinaciones de hiperparámetros, donde los valores de éstos provienen de distribuciones (en el caso más sencillo, uniformes).

- Regresión logística. 200 iteraciones
  - norma usada en regularización: l1, l2
  - $C \sim U(1, 99)$
- Máquina de soporte vectorial. 50 iteraciones
  - kernel: rbf, sigmoide y polinomial

  - coeficiente del kernel γ: escalado y automático
  - grado del polinomio: 2, 3 y 4
  - $C \sim U(0.1, 99.9)$

- Árbol de decisión. 50 iteraciones
  - Criterio: gini y entropía
  - Profundidad máxima $\sim \mathcal{U}[50, 201]$
  - Número máximo de características[1]: n, $\sqrt{n}$ y $\log_2 n$, donde n es el número de características.
  - Número de muestras mínimo para dividir un nodo $\sim \mathcal{U}[2, 11]$

- Bosques aleatorios. 50 iteraciones
  - número de árboles $\sim \mathcal{U}[80, 201]$
  - Criterio: gini y entropía
  - Profundidad máxima $\sim \mathcal{U}[50, 201]$
  - Número máximo de características: n, $\sqrt{n}$ y $\log_2 n$, donde n es el número de características.
  - Número de muestras mínimo para dividir un nodo $\sim \mathcal{U}[2, 11]$

2.3 Modelos de Aprendizaje profundo

Para esta sección se realizó el mismo preprocesamiento de los datos que se indica en la sección 2.1, y adicional a esto se obtuvo el vocabulario de todo el corpus de entrenamiento, con lo que se realizó un etiquetado de palabras de cada texto conforme al índice de dicha palabra en el vocabulario, agregando además dos índices extras, el del 0 para lo que se le conoce como *padding*, que representa el relleno de la lista de índices de tal modo que todas las oraciones tuvieran el mismo número de índices y así poderle pasar al modelo. Además se agrega el índice 1 que representa palabras desconocidas, que formarán parte del conjunto de validación, conformado por el 20 por ciento de los datos de entrenamiento. Es de importancia mencionar que para la partición de los datos de entrenamiento se usó como semilla el número 42 para la repetición de resultados.

Se implementó primero una red neuronal convolucional con una estructura básica que consta de una capa de *embeddings*, una un bloque convolucional que consta de una capa convolucional, la función de activación ReLU y un *max pooling*, finalmente se añade a la arquitectura un aplanado y la capa de clasificación. A esta arquitectura básica se le fueron agregando diferentes características tales como un *dropout*, una normalización por lotes y más bloques convolucionales, todo esto con la finalidad de intentar mejorar los resultados obtenidos. La arquitectura de este modelo se puede apreciar mejor en la figura 4.

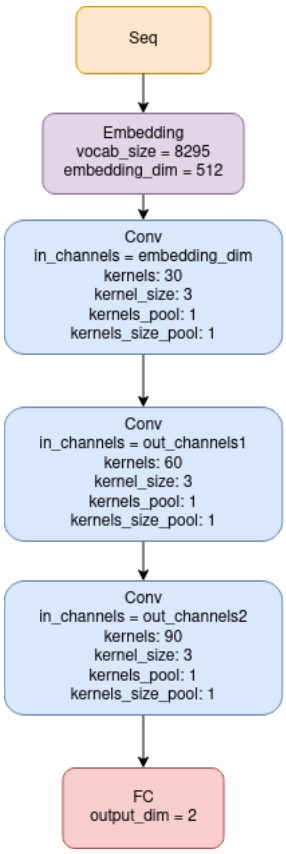

Figura 4: Arquitectura del modelo convolucional.

Por otro lado, se implementó una red neuronal recurrente GRU. Consiste de una capa de embeddings, cuyo vocabulario tiene un tamaño de 22888 palabras (incluyendo los tokens de *padding* y de token desconocido para palabras fuera del vocabulario aprendido), luego una serie de capas GRU, en las cuales se incluyó *dropout* con una probabilidad de 0.2 entre capas, y finalmente una capa completamente conectada cuya salida es una neurona, que es la que realiza la clasificación. La arquitectura de este modelo se puede apreciar mejor en la figura 5.

3 Resultados y análisis.

Los mejores hiperparámetros encontrados para cada modelo de aprendizaje máquina fueron

- Regresión logística: regularización $\ell 2$, $C = 2.789$

[1] a la hora de buscar la mejor división

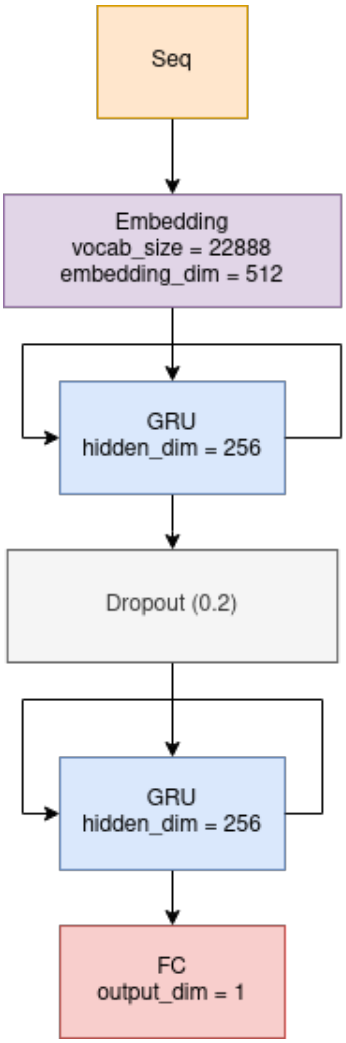

Figura 5: Arquitectura del modelo recurrente.

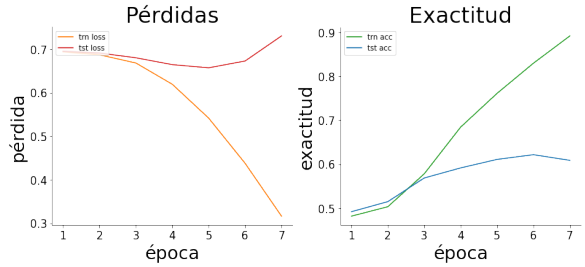

Figura 6: Resultados para la red convolucional con los mejores resultados.

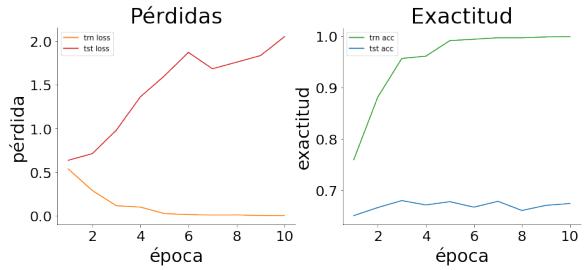

Figura 7: Resultados para la red recurrente con los mejores resultados.

- Máquina de soporte vectorial: kernel polinomial de grado 2 cuya gamma era igual a *scale*. $C = 52.523$
- Árbol de decisión: criterio gini, 64 niveles de profundidad máxima, utilizando todas las características para encontrar la mejor división. El número mínimo de muestras para dividir un nodo era de 2.
- Bosque aleatorios: 195 árboles usando el criterio de gini, hasta 178 niveles de profunidad, utilizando todas las características para encontrar la mejor división. 10 ejemplos eran necesarios para dividir un nodo.
- CNN: El modelo básico fue el que presentó mejores resultados. Se intentó con normalización por lotes, *dropout* y al menos otro bloque convolucional. Con todos se obtuvieron iguales o peores resultados. Se lograron los mejores resultados con un número de *embeddings* de 500 y 300 filtros para la capa convolucional y con un tamaño de 3, 1 para la capa de *max pooling* con tamaño de 1, sin relleno y con un paso de 1. Las gráficas de pérdida y de exactitud para los mejores resultados se muestran en la figura 6.

En la tabla 1 se pueden ver los desempeños de los modelos. Notamos que tanto la regresión logística

como el bosque aleatorio son los que presentan mejores resultados.

	Exactitud
Regresión logística	0.71
SVC	0.63
Árbol de decisión	0.66
Bosque aleatorio	0.71
CNN	0.64
RNN	0.67

Tabla 1: Resultados de los modelos.

4 Conclusiones

Como ya pudimos observar, los algoritmos de aprendizaje máquina son los que presentan mejores resultados, que no están tan lejanos a los puntajes oficiales de la competencia. Los algoritmos de aprendizaje profundo presentaron menores resultados y además tardaron más como es común, además de que son más difíciles de implementar, por lo que se recomienda seguir con la metodología de aprendizaje máquina clásica, o en todo caso, poder generar más datos para un mejor entrenamiento de redes neuronales profundas.Como trabajo a futuro también se podría pensar en una traducción hacia alguno de los dos idiomas (de preferencia, traducir al inglés) para tratar de mejorar resultados teniendo vectores preentrenados tales como word2vec, sin embargo, cada idioma presenta sus diferentes formas de presentar sexismo, por lo que habría que investigar y experimentar más al respecto.

Los datos, modelos y resultados se encuentran en el repositorio (Alvarado and Bermudez, 2021).

Referencias

Ó Alvarado and D Bermudez. 2021. Oscaralvaradom/sexism-identification-in-social-networks.

IberLEF-2021. 2021. Exist: sexism identification in social networks.

University of Oxford. 2021. Oxford learner's dictionaries.