Detección de lenguaje ofensivo con redes neuronales profundas

Edson Raul Cepeda Marquez
Facultad de Ingeniería Mecánica y Eléctrica
raulcepedac@hotmail.com

1. Introducción

Uno de los retos actuales de internet es el de mantener las plataformas digitales libres de agresiones, mensajes de odio, discriminacion y promover un ambiente sano para los usuarios. Las redes sociales son un punto un punto clave para esto puesto que generan una cantidad inmensa de datos que pueden ser utilizados para el estudio e implementación de inteligencia artificial. El objetivo de este proyecto es identificar correctamente texto con lenguaje ofensivo utilizando técnicas de procesamiento de lenguaje natural y redes neuronales profundas. El enlace al repositorio que contiene el código del proyecto se encuentra en la sección de referencias.

2. Herramientas

Se desarrolla código escrito en Python para la resolución de la problemática y se utiliza el entorno de programación Jupyter Notebook.

Para el desarrollo de los algoritmos de aprendizage de maquina se hace uso de las librerías Gensim y Keras. Para el control de version se utilza git y repositorios remotos de github.

La visualización de datos se realiza con la ayuda de la librería Matplotlib y para la manipulación de los conjuntos de datos Pandas.
Para la documentación se hace uso del sistema de composición de textos LaTex y el editor Overleaf.

3. Metodología

Se utiliza el conjunto de datos para la identificación de lenguaje ofensivo OLID. Este conjunto de datos contiene 14,100 tweets en ingles etiquetados para entrenamiento y pruebas.

id	tweet	a	b	c
86426	@USER She should ask a few...	OFF	UNT	NaN
90194	@USER @USER Go home you're...	OFF	TIN	NaN
16820	Amazon is investigating...	NOT	Nan	Nan

Tabla 1: Conjunto de datos OLID.

Limpieza de datos: Se realiza una limpieza al conjunto de datos de manera que no exista información innecesaria para el entrenamiento de un algoritmo de aprendizaje automático. Se remueven las etiquetas de id, subtask_b y subtask_c. Del texto se eliminan los signos de puntuación, caracteres innecesarios, emoticones y se convierte en minúscula.

Análisis Exploratorio: Con la limpieza hecha, se crea uno de los formatos estándar para el análisis, el corpus de texto. Se realiza un conteo de las palabras más frecuentes y se generan nubes de palabras para corroborar que el conjunto de datos tiene sentido y es factible para ser utilizado en un algoritmo.

Figura 1: Nube de palabras de texto ofensivo y no ofensivo etiquetado en el conjunto de datos.

Entrenamiento del algoritmo: Se entrena un algoritmo de word embeddings para la representación del texto en vectores y se diseña la arquitectura de una red neuronal profunda para la clasificación.

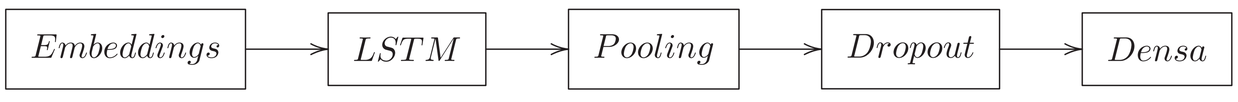

Figura 2: Arquitectura de la red neuronal.

4. Resultados

Al probar el modelo de word embeddings generado con el corpus de texto es posible buscar similitud de contexto para distintas palabras y realizar operaciones con los vectores de palabras.

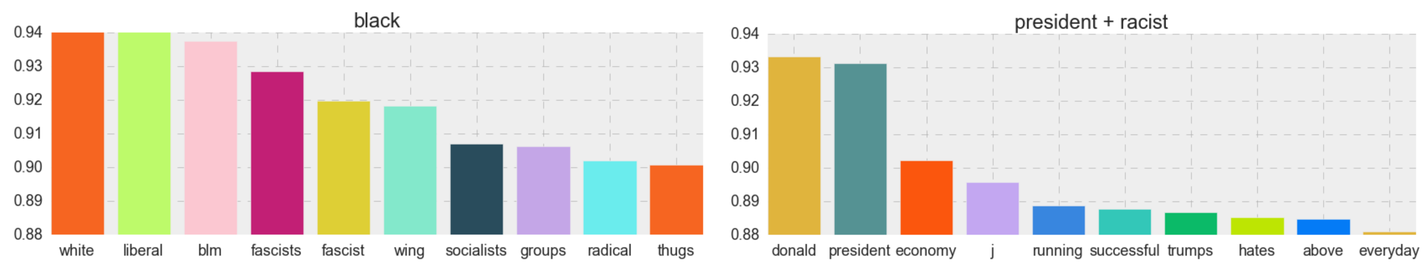

Figura 3. Prueba de similaridad en los vectores de palabras.

Con el word embedding resultante se entrena la red neuronal, se visualiza la variación de la precisión y la perdida y se utiliza la red para realizar predicciones sobre el conjunto de pruebas. Si el valor es mayor a 0.5 se considera ofensivo de otra manera se considera no ofensivo. Parte de las predicciones son las siguientes:

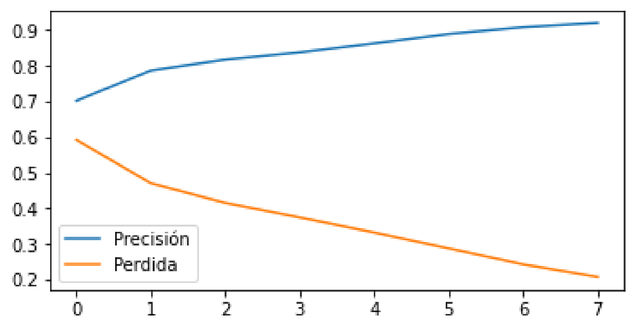

Figura 4. Variación en la precisión y la perdida de la red neuronal.

Texto	Predicción	Esperado
democrats support antifa muslim brotherhood...	[0.996477]	OFF
is revered by conservatives hated by progressives...	[0.029128]	NOT
first it reduces the ca	[0.082428]	NOT
getting the news that she is still up for parole...	[0.000400]	NOT
unity demo to oppose the farright in – — enough is enough	[0.961980]	OFF

Tabla 2. Predicciones finales de la red.

5. Conclusiones

Como se puede observar, combinar técnicas de procesamiento de lenguaje natural como word embeddings da un resultado muy positivo combinado con redes neuronales. Especificamente las redes neuronales recurrentes son efectivas para procesar texto puesto que para el análisis contextual es necesario que entradas anteriores afecten a las entradas siguientes, justo lo que sucede en las redes recurrentes LSTM (Long short-term memory). Además los word embeddings no solo sirven para ser la capa de entrada de una red neuronal si no que también permiten crear modelado de temas y modelos de predicciones de palabras.

6. Trabajo futuro

Se espera poder trabajar con conjunto de datos más grandes, mejor etiquetados y con una cantidad de características mayor para poder abarcar un mejor vocabulario. Con respecto a los word embeddings se debe utilizar modelos previamente entrenados con conjuntos de datos grandes que dan un mayor rendimiento cuando se usan con redes neuronales.

Se propone tambíen probar distintas arquitecturas de redes profundas, no solo con redes recurrentes si no también con redes convolucionales.

7. Referencias

[1] Zampieri, Marcos and Malmasi, Shervin and Nakov, Preslav and Rosenthal, Sara and Farra, Noura and Kumar, Ritesh.(2019) Predicting the Type and Target of Offensive Posts in Social Media. Proceedings of NAACL.

[2] Bongo. (2020). Do Pretrained Embeddings Give You The Extra Edge?
Recuperado de: *https://www.kaggle.com/sbongo/do-pretrained-embeddings-give-you-the-extra-edge*

[3] Edson Cepeda. (2020). Offensive Language Detection.
github.com/OrbitalCardinal/OffensiveLanguageDetection