

# **Intro. al Procesamiento de Lenguaje Natural Curso 2020**

## **Laboratorio – Detección de Discurso de Odio**

El objetivo de este laboratorio es implementar un clasificador en el lenguaje de programación Python para detectar discurso de odio en redes sociales. En particular nos enfocaremos en la red social Twitter.

Se pretende que el estudiante investigue y utilice clasificadores y funcionalidades de la librería sklearn y/o modelos de redes neuronales mediante la librería keras o pytorch para la construcción de los clasificadores.

Al finalizar el laboratorio, se evaluarán las entregas realizadas en un conjunto no conocido de tweets y se publicarán los resultados de cada grupo.

**Los estudiantes que estén en el podio (primeros 3 lugares) recibirán una bonificación de 5 puntos sobre el total (100) del curso.**

**Nota: Es importante destacar que los datos utilizados en este laboratorio son reales y pueden contener insultos, lenguaje agresivo y ofensivo.**

### **Descripción**

Se pretende que el estudiante construya un clasificador de discurso de odio en textos con los insumos impartidos. Se dispondrá de un corpus anotado y un repertorio de vectores de palabras con amplia cobertura sobre dicho corpus.

Además cada grupo podrá utilizar recursos adicionales siempre y cuando estos estén debidamente referenciados y justificados, preguntando previamente en el foro para obtener autorización por parte de los docentes y dar conocimiento al resto de los grupos.

El discurso de odio abordado en este trabajo refiere a la discriminación o agresión a un grupo de individuos por una característica ideológica o física (ej. racismo). No debe confundirse con una agresión a una persona particular, que en el marco de este trabajo no es considerado discurso de odio. El corpus impartido fue construido con esta consideración.

### **Corpus**

El corpus consiste en una colección de textos extraídos de Twitter, donde cada texto tiene una etiqueta binaria: 0 (sin discurso de odio) o 1 (contiene discurso de odio). Este corpus fue originado con anotación manual colaborativa mediante la web <https://odioelodio.com/>, en el trabajo realizado como proyecto de grado por Manuel Saravia y Lucas Kunc. El corpus es impartido en formato csv y dividido en subconjuntos de entrenamiento, validación y evaluación.

# Introducción al Procesamiento de Lenguaje Natural – Laboratorio

A continuación se presentan unos ejemplos extraídos del corpus:

Cascada chorro del indio hermoso	0
Significa que estás muy sonsa y necesitas ayuda	0
Callate la jeta guebon, o mejor muerete chavista de mierda!!!	1
Que pueden esperar de un negro Así, igual que el hermano, lo echaron de Boca por chorro.	1

Se recomienda a cada grupo utilizar el corpus de entrenamiento para entrenar y el corpus de validación para la búsqueda de hiperparámetros y monitorear el sobreajuste del modelo. Luego utilizar el corpus de evaluación para evaluar el resultado final y seleccionar el modelo a entregar. **La implementación entregada será evaluada en un conjunto de datos nuevo, desconocido para el sistema.**

## Interfaz

El clasificador debe ser entrenado respetando exactamente la siguiente interfaz:

➤ `python3 es_odio.py <data_path> test_file1.csv ... test_fileN.csv`

Donde **<data\_path>** tiene la ruta donde se encuentran los datos impartidos:

- train.csv, val.csv, test.csv
- fasttext.es.300.txt

y **test\_file1.csv ... test\_fileN.csv** son un conjunto de archivos de test (pueden tener salida o no).

El programa debe:

1. Entrenar un clasificador (**hiperparámetros encontrados previamente**) utilizando los conjuntos de **train** y **val** (**debe demorar menos de 10 minutos en una CPU intel i7**)
2. Por cada archivo **test\_file1.csv ... test\_fileN.csv** el programa debe:
  1. aplicar el modelo previamente entrenado
  2. generar un archivo de salida **test\_file1.out ... test\_fileN.out** con las salidas obtenidas a cada archivo de test. El archivo de salida debe tener las salidas (0 o 1) en orden y separados por un fin de línea. (Ej. 1\n0\n0\n1...\n0)

## Insumos y librerías

Se utilizará el lenguaje de programación Python 3 con las librerías numpy, sklearn y keras en versiones actuales. Se recomienda utilizar la versión de python 3.8 de Anaconda [2]. El repaso de python [3] del curso de Stanford cs224n es una buena introducción rápida y en general otras partes del curso puede resultarles un material útil de consulta.

Para los experimentos y realizar la búsqueda de hiperparámetros puede resultarles útil Jupyter Notebook [4]. En caso de que utilice un notebook para la búsqueda de hiperparámetros puede incluirlo en su entrega.

## Entrega

La entrega de este laboratorio es el **lunes 30 de noviembre a las 23:59** y se realizará mediante un formulario en el EVA del curso que estará disponible próximo a la fecha de entrega.

Se debe entregar:

- **es\_odio.py** con el programa pedido
- **otros archivos .py** con funcionalidades auxiliares
- **readme.txt** describiendo lo realizado incluyendo las versiones utilizadas
- **grupo.txt** con el número de grupo y las cédulas de los integrantes del grupo (una por línea) sin puntos ni dígito de verificación.  
(Ej. 7\n1111111\n2222222\n3333333\n4444444)
- **test.out** con la salida obtenida al ejecutar la versión entregada de su programa sobre el conjunto humor\_test.csv

Es un requisito para la aprobación del laboratorio que la entrega realizada ejecute correctamente, respete la interfaz detallada y despliegue las salidas en el formato pedido.

El Laboratorio representa un 20% de la aprobación total del curso.

## Referencias

- [1] <https://www.python.org>
- [2] <https://www.anaconda.com/products/individual>
- [3] <http://web.stanford.edu/class/cs224n/readings/python-review.pdf>
- [4] <https://jupyter.org/>