# Laboratorio de Introducción al Procesamiento de Lenguaje Natural

# Tarea 2

El objetivo de este laboratorio es realizar diferentes experimentos para representar y clasificar textos. Para esto se trabajará con un corpus para análisis de sentimiento, creado para la competencia TASS 2020 (IberLEF - SEPLN).

## Entrega

Deberán entregar un archivo *.ipynb* con su solución, que incluya código, comentarios y respuestas a las preguntas que se incluyen al final de este notebook.

El plazo de entrega de la tarea 2 cierra el **20 de junio a las 23:59 horas**.

## Plataforma sugerida

Sugerimos que utilicen la plataforma Google colab, que permite trabajar colaborativamente con un *notebook* de python. Al finalizar pueden descargar ese *notebook* en un archivo .ipynb, incluyendo las salidas ya ejecutadas, con la opción `File -> Download -> Download .ipynb`.

## Aprobación del laboratorio

Para aprobar el laboratorio se exige como mínimo:

- Probar dos enfoques diferentes para la representación de tweets (uno basado en BoW y otro en word embeddings)
- Probar al menos dos modelos de aprendizaje automático con cada representación
- Comparar los resultados con los obtenidos por el modelo de pysentimiento. El preprocesamiento, las pruebas con otras formas de representación de los tweets, los experimentos con otros modelos de aprendizaje automático, incluyendo aprendizaje profundo, entre otros posibles experimentos, no son requisito para aprobar el laboratorio, aunque aportan a la nota final.

# Parte 1 - Carga y preprocesamiento del corpus

Para trabajar en este notebook deben cargar los tres archivos disponbiles en eva: train.csv, devel.csv y test.csv.

La aplicación de una etapa de preprocesamiento similar a la implementada en la tarea 1 es opcional. Es interesante hacer experimentos con y sin la etapa de preprocesamiento, de modo

de comparar resultados (sobre el corpus de desarrollo, devel.csv) y definir si se incluye o no en la solución final.

```
# Carga de los datasets
```

```
# Preprocesamiento de los tweets
```

## Parte 2 - Representación de los tweets

Para representar los tweets se pide que experimenten con modelos basados en Bag of Words (BoW) y con Word Embeddings.

Para los dos enfoques podrán elegir entre diferentes opciones:

**Bag of Words**

- BOW estándar: se recomienda trabajar con la clase CountVectorizer de sklearn, en particular, fit_transform y transform.
- BOW filtrando stop-words: tienen disponible en eva una lista de stop-words para el español, adaptada para análisis de sentimiento (no se filtran palabras relevantes para determinar la polaridad, como "no", "pero", etc.).
- BoW usando lemas: pueden usar herramientas de spacy.
- BOW seleccionando las features más relevantes: se recomienda usar la clase SelectKBest y probar con diferentes valores de k (por ejemplo, 10, 50, 200, 1000).
- BOW combinado con TF-IDF: se recomienda usar la clase TfidfVectorizer

**Word Embeddings**

- A partir de los word embeddings, representar cada tweet como el vector promedio (mean vector) de los vectores de las palabras que lo componen.
- A partir de los word embeddings, representar cada tweet como la concatenación de los vectores de las palabras que lo componen (llevando el vector total a un largo fijo).

Se recomienda trabajar con alguna de las colecciones de word embeddings disponibles en https://github.com/dccuchile/spanish-word-embeddings. El repositorio incluye links a ejemplos y tutoriales.

Se pide que prueben al menos una opción basada en BoW y una basada en word embeddings.

```
# Representación de los tweets usando BoW
```

```
# Representación de los tweets usando word embeddings
```

## Parte 3 - Clasificación de los tweets

Para la clasificación de los tweets es posible trabajar con dos enfoques diferentes:

- Aprendizaje Automático basado en atributos: se pide probar al menos dos modelos diferentes, por ejemplo, Multi Layer Perceptron (MLP) y Support Vector Machines (SVM), y usar al menos dos formas de representación de tweets (una basada en BoW y otra basada en word embeddings). Se publicó en eva un léxico de palabras positivas y negativas que puede ser utilizado para generar atributos.
- Aprendizaje Profundo: se recomienda experimentar con alguna red recurrente como LSTM. En este caso deben representar los tweets an base a word embeddings.

Deberán usar el corpus de desarrollo (devel.csv) para comparar resultados de diferentes experimentos, variando los valores de los hiperparámetros, la forma de representación de los tweets, el preprocesamiento, los modelos de AA, etc.

Tanto para la evaluación sobre desarrollo como para la evaluación final sobre test se usará la medida Macro-F1 (promedio de la medida F1 de cada clase).

```
# Experimentos con Aprendizaje Atuomático y BoW
```

```
# Experimentos con Aprendizaje Atuomático y word embeddings
```

```
# Experimentos con Aprendizaje Profundo
```

## Parte 4: Evaluación sobre test

Deben probar los mejores modelos obtenidos en la parte anterior sobre el corpus de test.

También deben comparar sus resultados con un modelo pre-entrenado para análisis de sentimientos de la biblioteca pysentimiento (deben aplicarlo sobre el corpus de test).

```
# Evaluación sobre test
```

# Preguntas finales

Responda las siguientes preguntas:

1) ¿Qué modelos probaron para la representación de los tweets?

2) ¿Aplicaron algún tipo de preprocesamiento de los textos?

3) ¿Qué modelos de aprendizaje automático probaron?

4) ¿Qué atributos utilizaron para estos modelos?

5) ¿Probaron algún enfoque de aprendizaje profundo?

6) ¿Probaron diferentes configuraciones de hiperparámetros?

7) ¿Qué enfoque (preprocesamiento + representación de tweets + modelo + atributos/parámetros) obtuvo la mejor Macro-F1?

8) ¿Qué clase es la mejor clasificada por este enfoque? ¿Cuál es la peor? ¿Por qué piensan que sucede esto?

9) ¿Cómo son sus resultados en comparación con los de pysentimiento? ¿Por qué piensan que sucede esto?