# Introducción al Procesamiento de Lenguaje Natural - Laboratorio 2015

El objetivo del laboratorio 2015 del curso Introducción al Procesamiento de Lenguaje Natural (PLN) es aplicar técnicas de PLN para la resolución de una tarea de clasificación de textos escritos. Para esto, el estudiante deberá familiarizarse con diferentes herramientas de PLN de acceso libre e incorporarlas en un entorno unificado de programación (en este caso, sobre la plataforma Python).

La tarea a resolver consiste en clasificar comentarios sobre películas como positivos, negativos o neutros. Para esto será necesario el análisis del texto, la extracción de atributos para la clasificación, la selección de los atributos  y del modelo de aprendizaje automático, y la evaluación de los resultados.

## Corpus

El corpus a utilizar son comentarios de usuarios del sitio web http://cartelera.com.uy en los años 2014 y 2015[1]. Está compuesto por 1447 comentarios sobre películas, y una calificación de de 1 a 5 (además de información extra sobre la película como su nombre, año de estreno, reseña y fecha y hora del comentario). A los efectos de este laboratorio, se considerará que un comentario es positivo si tiene calificación 4 o 5,  neutro si tiene 3, y negativo si tiene 1 o 2.

## Procedimiento de trabajo

Al comienzo del laboratorio, se dejará disponible el corpus, así como un notebook de IPython con los comandos para importarlo y separar conjuntos de entrenamiento y testeo. Estos conjuntos deberán utilizarse durante todo el laboratorio. Además se proveerá, como recurso auxiliar, una lista de términos positivos y negativos.

Con este corpus, los estudiantes deberán utilizar el método de clasificación basado en modelos de Entropía Máxima provisto por la biblioteca NLTK de Python, para construir un clasificador que, dado un nuevo comentario, lo clasifique en una de las tres categorías de orientación semántica mencionadas.

### Herramientas

- Se utilizará la plataforma Python para el desarrollo, en su versión 3.3 o superior. En particular, los resultados se probarán sobre una distribución Anaconda (https://store.continuum.io/cshop/anaconda/), por lo que sugerimos instalarla
- Las biblioteca para procesamiento de lenguaje natural sobre Python a utilizar es NLTK (http://www.nltk.org/).

[1] Agradecemos a la empresa Montevideo.com por cedernos gentilmente el corpus para este trabajo

- Para la tokenización, tagging y cualquier tarea para el idioma español, se utilizará la herramienta FreeLing (http://nlp.lsi.upc.edu/freeling/), la que deberá integrarse al ambiente
- Podrán utilizarse bibliotecas adicionales de Python, tales como: NumPy y SciPy (http://www.scipy.org/), pandas (http://pandas.pydata.org/) o scikit-learn (http://scikit-learn.org)
- Podrán utilizarse herramientas y bibliotecas adicionales, siempre y cuando se integren al entorno Python de trabajo

**Recursos**

Además de la lista de palabras positivas y negativas, podrán utilizarse recursos como listas de palabras o bases de datos léxicas como WordNet (solamente por mencionar ejemplos), siempre y cuando estén disponibles libremente en Internet, o sean construidas por los estudiantes.

**Formato de entrega**

La entrega deberá realizarse utilizando un notebook IPython donde se incluirá tanto la documentación, como el código Python a ejecutar.

Las tareas a realizar y documentar en el notebook son las siguientes:

- Importación del corpus y separación en corpus de entrenamiento y testeo (provisto por los docentes).
- De ser necesario, depuración de los textos quitando/sustituyendo marcas HTML o similares.
- Breve análisis del corpus (tamaño, cantidad de instancias en cada clase, etc).
- Tokenización de los textos utilizando NLTK y conversión a un formato susceptible de ser utilizado para clasificación. En una primera instancia, se utilizarán como features las n palabras más frecuentes. Para ajustar n, deberán realizarse pruebas con diferentes valores, utilizando cross-validation sobre el corpus de entrenamiento.
- Incorporar como features la pertenencia a la lista de palabras proporcionadas y repetir el proceso. Comparar resultados.
- Repetir el proceso anterior utilizando Freeling para tokenizar. Comparar resultados.
- Repetir el proceso anterior incorporando información de POS-tagging.
- (Opcional) mejorar el modelo incorporando otras features y repetir el proceso anterior.
- Si es necesario en algún caso ajustar algún parámetro del modelo, siempre utilizar cross-validation.

Está permitido, buscando mejorar los resultados, incorporar otras tareas o recursos; esto deberá documentarse claramente en el informe.

Cada una de estas tareas deberá estar documentada, incluyendo al menos:

- las decisiones de diseño tomadas para cada paso, y los resultados obtenidos.
- una descripción de la metodología utilizada (corpus de entrenamiento y testeo, método para validación cruzada).
- valores intermedios y finales de las medidas de evaluación

Adicionalmente, deberán incluirse los resultados de la aplicación de los mejores modelos obtenidos durante el entrenamiento, sobre el corpus de evaluación, en la forma precision/recall/f-score sobre cada categoría (en un formato one-versus-all), y presentando la matriz de confusión. El informe deberá incluir un análisis de los resultados.

**Evaluación**

Para la evaluación del laboratorio, se tendrá en cuenta:

- El cumplimiento de las tareas pedidas, incluyendo el código solicitado
- La claridad de la documentación, en particular la justificación de las decisiones tomadas, así como el análisis de los resultados obtenidos

*Deberá presentarse en la entrega toda la información de configuración necesaria para poder reproducir el proceso de entrenamiento y evaluación. Esto será condición necesaria para la aprobación del curso.*

**Insumos**

Se proveerán los siguientes archivos para la realización del laboratorio:

- **`comentarios_peliculas.csv`**: corpus con los comentarios de películas.
- **`elementos_subjetivos.txt`**: lista de palabras con valoración positiva o negativa.
- **`Laboratorio1_IntroPLN.ipynb`**: notebook IPython con una implementación inicial que carga el corpus y lo separa en un conjunto de entrenamiento y uno de test.

**Referencias**


- Bird, Klein, Lopper: “Natural Language Processing with Python” (esp. capítulo 6) - http://www.nltk.org/book/
- Referencia de NLTK - http://www.nltk.org/api/nltk.html
- The IPython notebook - http://ipython.org/notebook.html
- Freeling 3.1, sitio oficial - http://nlp.lsi.upc.edu/freeling/