

## Proyecto: Clasificación de Textos en Lenguaje Natural

**Objetivo:** Construir un sistema para la clasificar automáticamente tweets de usuarios sobre el COVID en Positivo y Negativo (Análisis de sentimiento).

**Contenidos:**

### *Parte 1 Estimación de probabilidades en el modelo del lenguaje*

En esta parte se estimarán las probabilidades del modelo del lenguaje para las clases Positivo y Negativo. Utiliza el fichero `COV_train.xlsx` en el campus virtual. Tienes 33444 tweets clasificados en cada una de las categorías con el formato:

`<texto>,<descripción>`

#### **1.1 Creación del vocabulario**

Halla el vocabulario del problema. Para ello examina el fichero `COV_train.xlsx`, obtén qué palabras están presentes en los tweets (preprocesamiento y tokenización) y pon las palabras en el fichero `vocabulario.txt`. Si una palabra se repite ponla sólo una vez. Las palabras del fichero de vocabulario deben estar ordenadas alfabéticamente.

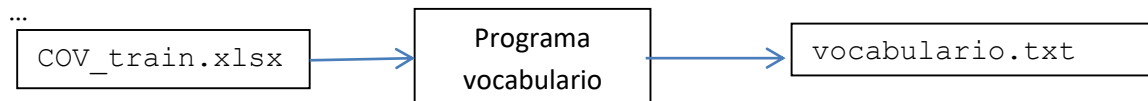
El fichero `vocabulario.txt` tendrá el formato:

Numero de palabras:<Número entero>

<palabra>

<palabra>

...



**Entregable**

**En el Campus Virtual**

– **Programas:**

○ Vocabulario

– **Ficheros:**

`vocabulario.txt`

– **Nota:** Práctica individual, lenguaje de programación libre, utilización de librerías libre. Se penalizará con un 50% no entregar los ficheros en el formato pedido.

## Preprocesamiento

### Tareas típicas:

- Pasar a minúsculas.
- Eliminación de signos de puntuación.
- Eliminación de palabras reservadas (stopwords).
- Eliminación de emojis y emoticonos o su conversión a palabras.
- Eliminación de URLs, etiquetas HTML, hashtags.
- Corrección ortográfica.
- Truncamiento: Reducir una palabra a su raíz (grito, grita, gritos, gritas ->grit).
- Lematización: Reducir una palabra a su forma canónica (dije,diré,dijéramos->decir).

### Algunas stopwords en inglés:

a, able, about, across, after, all, almost, also, am, among, an, and, any, are, as, at, be, because, been, but, by, can, cannot, could, dear, did, do, does, either, else, ever, every, for, from, get, got, had, has, have, he, her, hers, him, his, how, however, i, if, in, into, is, it, its, just, least, let, like, likely, may, me, might, most, must, my, neither, no, nor, not, of, off, often, on, only, or, other, our, own, rather, said, say, says, she, should, since, so, some, than, that, the, their, them, then, there, these, they, this, tis, to, too, twas, us, wants, was, we, were, what, when, where, which, while, who, whom, why, will, with, would, yet, you, your

## 1.2 Estimación de probabilidades

La estimación de las probabilidades para los corpus correspondiente a las clases P o N. Se escribirá en un fichero de texto llamado `modelo_lenguaje_<P o N>.txt`. En el fichero de texto debe aparecer:

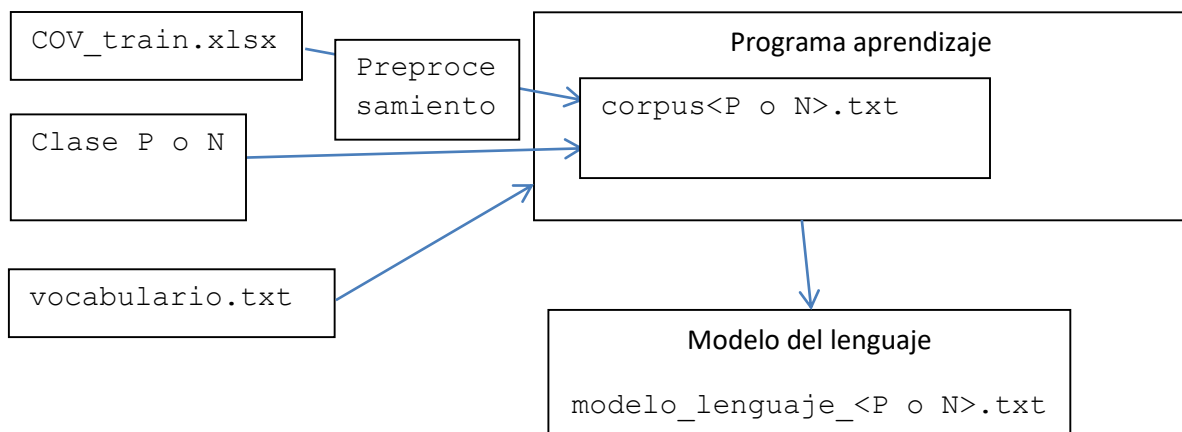
Cabecera:

Numero de documentos (tweets) del corpus :<número entero>

Número de palabras del corpus:<número entero>

Por cada palabra de `vocabulario.txt`, su frecuencia absoluta en el corpus y una estimación del logaritmo de su probabilidad mediante suavizado laplaciano con tratamiento de palabras desconocidas. Las palabras en los ficheros de aprendizaje estarán ordenadas alfabéticamente.

Palabra:<cadena> Frec:<número entero> LogProb:<número real>



## Entregable

### En el Campus Virtual

- **Programas:**
  - o Aprendizaje
- **Ficheros:**
  - `modelo_lenguaje_<P o N>.txt`.
- **Nota:** Proyecto individual, lenguaje de programación libre. Se penalizará con un 50% no entregar los ficheros en el formato pedido.

## Parte 2 Clasificación

En esta parte se clasificarán las descripciones presentes como Positivo (P) o Negativo (N).

Escribe un programa que tome como entrada las estimaciones de probabilidad de cada palabra en `modelo_lenguaje_<P o N>.txt` y pida un corpus con tweets a clasificar: `COV_test.xlsx` (con el formato de corpus de `COV_train.xlsx` sin la clase).

El programa debe clasificar todos los tweets de `COV_test.xlsx` y devolver los tweets clasificados en dos ficheros:

- `clasificacion_alu<numero de alu>.txt` donde cada línea del fichero de salida tiene el formato:

`<primeros 10 caracteres del tweet>,<lP en P>,<lP en N>,<P o N>`

`lP` : logaritmo neperiano de la probabilidad del tweet, con 2 decimales.

`<P o N>` la clase en la que se clasifica el tweet.

- `resumen_alu<numero de alu>.txt` donde cada línea del fichero de salida tiene el formato:

`<P o N>` : clase en la que se clasifica el tweet.

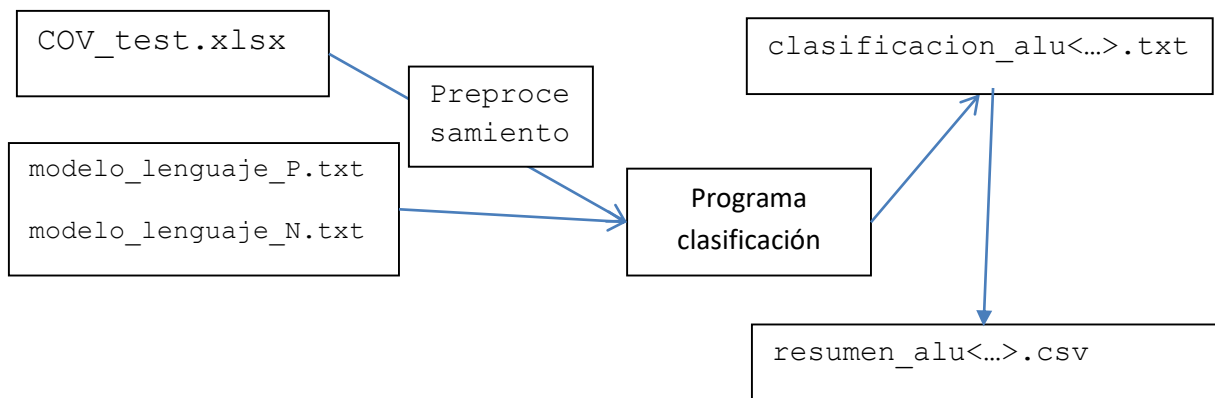
### Notas:

En los ficheros de salida no deben aparecer corchetes `<,>`

Los tweets clasificados deben estar en el mismo orden de entrada.

Antes de subirlo, probar el programa con `COV_train.xlsx` quitando la clase y estimar el error de clasificación. Este error debe escribirse en el informe de la implementación.

Se penalizará con un 50% de la evaluación no ajustarse al nombre del fichero o al formato pedido.



### ***Evaluación del Proyecto***

- Entregables: Breve informe con la implementación: Preprocesamiento, librerías utilizadas, implementación de los programas, error sobre corpus de entrenamiento. Programas y ficheros pedidos (1/10)
- Rendimiento del programa sobre el corpus que proporcionará el profesor (7/10):
  - 98-100% del porcentaje de acierto del mejor programa 7 puntos
  - 95-98% del porcentaje de acierto del mejor programa 6 puntos
  - 93-98% del porcentaje de acierto del mejor programa 5 puntos
  - 85-93% del porcentaje de acierto del mejor programa 4 puntos
  - 75-85% del porcentaje de acierto del mejor programa 3 puntos
  - 65-75% del porcentaje de acierto del mejor programa 2 puntos
  - Menos del 65% del porcentaje de acierto del mejor programa 0 puntos