Proyecto: Análisis de Sentimiento en Lenguaje Natural

Objetivo: Construir un Sistema de Análisis de Sentimiento para Lenguaje Natural.

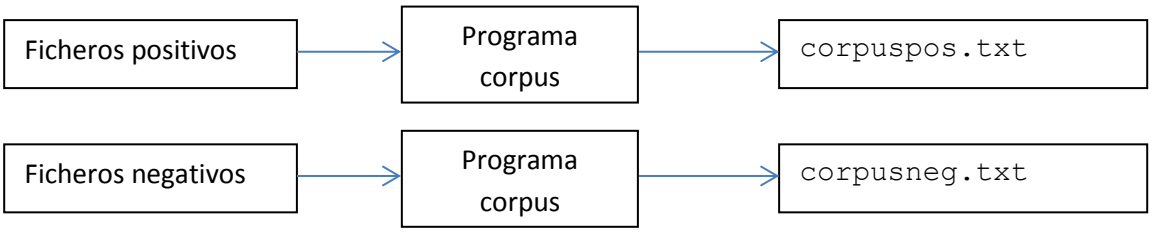
Parte 1 Estimación de probabilidades en el modelo del lenguaje

En esta parte se estimarán las probabilidades del modelo del lenguaje para las clases positivo y negativo

1.1 Creación de los corpus

Utiliza los documentos sobre valoraciones de películas proporcionado en el campus virtual. Tienes 12500 documentos con valoración positiva y 12500 con valoración negativa.
Ejemplo de opinion positiva: Very good drama…
Ejemplo de opinion negativa: Sometimes a movie is so comprehensively awful…
Crea un corpus con nombre `corpus<pos o neg>.txt` que una todos los documentos con opinión positiva y otro corpus con todos los documentos con opinión negativa. Cada línea del fichero de salida en el corpus debe tener la siguiente estructura:

```
Texto:<cadena con texto del fichero>
```



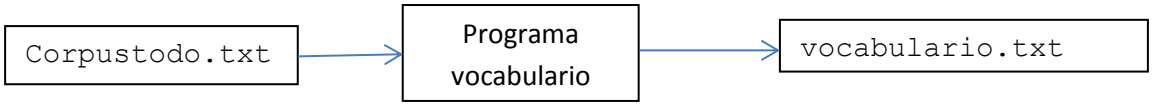
Crea también el fichero `corpustodo.txt` concatenando `corpuspos.txt` y `corpusneg.txt`

1.2 Creación del vocabulario

Halla el vocabulario del problema. Para ello examina el fichero `corpustodo.txt` y obtén las palabras del vocabulario a partir del texto.

(Ayuda:http://rosettacode.org/wiki/Tokenize_a_string). Debes generar un fichero de salida `vocabulario.txt` con cabecera

```
Numero de palabras:<Número entero>
Palabra:<cadena>
```



Las palabras de `vocabulario.txt` estarán ordenadas alfabéticamente.

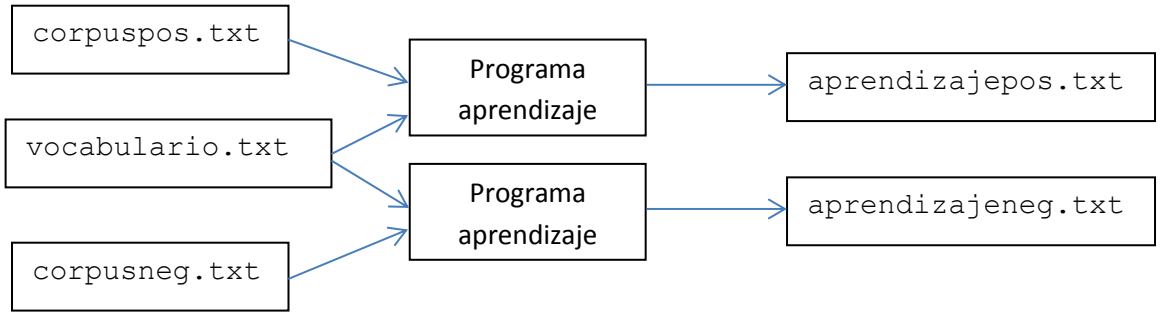
1.3 Estimación de probabilidades

La estimación de las probabilidades se escribirá en un fichero de texto llamado `aprendizaje<pos o neg>.txt`. En el fichero de texto debe aparecer:
Cabecera:

```
Numero de documentos del corpus :<número entero>
Número de palabras del corpus:<número entero>
```

Por cada palabra de `vocabulario.txt`, su frecuencia 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>
```

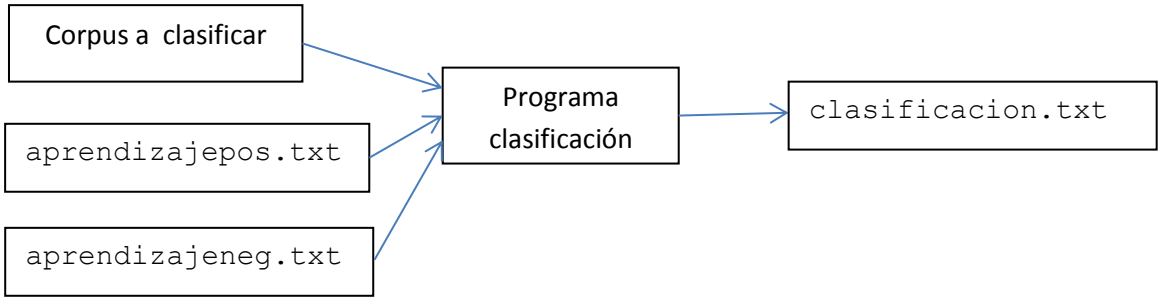


Parte 2 Clasificación

En esta parte se clasificarán los documentos presentes en un corpus.

Escribe un programa que tome como entrada las estimaciones de probabilidad de cada palabra y un corpus con documentos a clasificar y devuelva los documentos clasificados en un fichero `clasificación.txt` donde cada línea del fichero de salida con el corpus tenga la siguiente estructura:

```
Clase:<pos o neg> Texto:<cadena con texto del fichero>
```

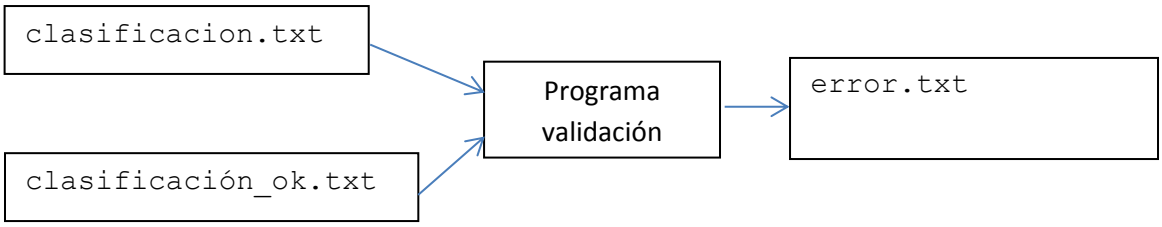


Deberás clasificar `corpustodo.txt` generando el fichero `clasificación.txt`

Parte 3 Validación

En esta parte se estimará el rendimiento del sistema.

Escribe un programa que estime el rendimiento de sistema. Deberá tomar el fichero `clasificación.txt` y otro fichero con el mismo formato llamado `clasificación_ok.txt` y escribir el porcentaje de errores cometidos en la clasificación (con qué porcentaje la clase para `clasificación.txt` no coincide con `clasificación_ok.txt`) en el fichero `error.txt` con formato `Porcentaje de error:<número real>`



Deberás generar el fichero `clasificación_ok.txt` para `corpustodo.txt` y el fichero `error.txt`

Nota: La descripción anterior del proyecto constituye un conjunto de requerimientos que es obligatorio cumplir.

Entregable

En el Campus Virtual

- **Programas:**
  - Corpus, Aprendizaje, Clasificación, Validación (fuentes)
- **Ficheros:**
  - `vocabulario.txt, aprendizajepos.txt, aprendizajeneg.txt, clasificacion.txt, clasificación_ok.txt, error.txt`
- **Documentación:**
  - Descripción de la implementación incluyendo cual ha sido la participación de cada alumno

Presentación

Debes pasar por el despacho del profesor para presentar el proyecto y ejecutar el programa sobre un conjunto de testeo en formato corpus que deberás leer de un pen-drive.

Calificación

- Documentación 2 puntos
- Adecuación de la implementación a los requerimientos 2 puntos
- Complejidad computacional de los programas 2 puntos
- Resultados sobre el conjunto de testeo 4 puntos

Notas adicionales:

- **Máximo 2 alumnos por proyecto**. No puedes repetir con quien ya hayas trabajado salvo causa justificada que deberá aprobar el profesor.

- El proyecto es tutorizado.
- Lenguaje de programación libre.
- Fecha límite: 6 de Junio, pero es altamente recomendable que lo entregues antes del comienzo de los exámenes.
- Presentación al profesor durante los horarios de tutoría.
- Se disponen de horas de prácticas (virtuales) para hacer el proyecto. Programación orientativa (no es obligatoria)
    - Parte 1 Horas de prácticas de la Semana del 28 de Abril
    - Parte 2 Horas de prácticas de la Semana del 5 de Mayo
    - Parte 3 y Entregables Horas de prácticas de la Semana del 12 de Mayo
- Se comprobará que no se ha copiado el proyecto y se penalizará severamente la nota final de la asignatura en su caso.