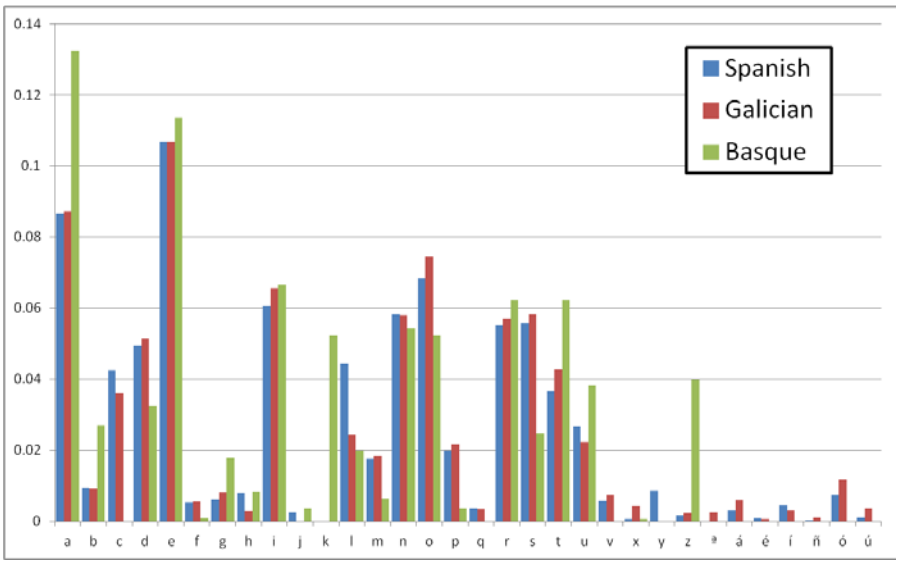
# TAP Labo 25/11/2015

## La torre de Babel

La finalidad de la presente práctica será poder relacionar diferentes lenguas entre sí, estableciendo una medida de semejanza en función de su forma escrita.

Supongamos que tenemos un mismo texto traducido a dos lenguas diferentes. Es de esperar que si estas dos lenguas son *parecidas*, los respectivos textos también lo sean. Obviamente, esto no ocurrirá siempre, ya que pudieran existir dos lenguas semejantes que utilicen una escritura totalmente distinta. Pero, como veremos en esta práctica, este sencillo criterio será suficiente para la gran mayoría de las lenguas de nuestro planeta. Por tanto, definiremos la semejanza entre dos lenguas como la semejanza entre un mismo texto traducido a ambas lenguas.

De nuevo, definir la semejanza entre dos textos es una tarea compleja. Una de las maneras más sencilla (y aún así eficaz) es comparar la frecuencia de aparición de los caracteres en cada texto. La siguiente figura, denominada *histograma* (http://es.wikipedia.org/wiki/Histograma), muestra la frecuencia de aparición de algunos (no todos) de los caracteres que pudiéramos encontrar en textos de las tres lenguas: español, gallego y vasco. Como puede observarse en la figura, el español y el gallego comparten cierta semejanza en cuanto a frecuencia de aparición de caracteres, mientras que el vasco se diferencia de ellos.



Es posible definir una sencilla función que calcule la semejanza o grado de solapamiento de dos histogramas *h1* y *h2*:

$$Similarity(h1,h2) = \sum_{c \in h1 \cup h2} \min(h1(c), h2(c))$$

Es decir, la similitud es la suma del mínimo de frecuencias para todos los caracteres de ambos histogramas. Dos lenguas que no compartan ningún carácter (tengan una escritura totalmente diferente) tendrán una semejanza 0, mientras que la semejanza entre una lengua y ella misma será 1 (la suma de todas las frecuencias de un histograma es igual a 1).

## Un ejemplo práctico

Si tomamos la Declaración Universal de los Derechos Humanos (http://www.un.org/es/documents/udhr/law.shtml), que está traducida a prácticamente todas las lenguas del planeta, y analizamos cuál es la semejanza entre las cuatro lenguas oficiales de España (español-spn, gallego-gln, catalán-cln y vasco-bsq), obtendríamos la siguiente tabla:

| | spn | gln | cln | bsq |
|---|---|---|---|---|
| spn | 1 | 0.946 | 0.909 | 0.757 |
| gln | 0.946 | 1 | 0.899 | 0.765 |
| cln | 0.909 | 0.899 | 1 | 0.761 |
| bsq | 0.757 | 0.765 | 0.761 | 1 |

## Tarea 1

Descargar el archivo http://gtts.ehu.es/German/Docencia/TAP/Labo/udhr.zip que contiene la Declaración Universal de los Derechos Humanos traducida a 281 lenguas en formato de texto (codificación UTF-8) y descomprimirlo.

Crear una clase `Language` que represente a una lengua y pueda instanciarse mediante un fichero de texto. El constructor se parecerá a:

```
public Language(String id, String descr, String fileName){…}
```

Donde `id` será un identificador de dicha lengua, `descr` una breve descripción y `fileName` será el nombre de un fichero de texto en formato UTF-8 que contendrá texto en dicha lengua. La clase `Language` deberá almacenar internamente el histograma que se obtenga a partir del fichero de texto. Para ello, puede utilizarse un mapa que relacione cada carácter visto con su frecuencia de aparición: `Map<Character,Double>`. La clase también deberá implementar los siguientes métodos:

```
public String getId() {…}

public String getDescr(){…}
```

```
public String toString(){…}

public static double similarity(Language a, Language b){…}

public static void main(String[] args){…}
```

No hace falta comentar los tres primeros métodos, ya que resulta obvio qué pueden devolver. El método `similarity` (nótese el modificador `static`) deberá tomar dos lenguas y devolver su semejanza. Por último, el método `main` deberá obtener la semejanza entre las cuatro lenguas oficiales de España en base a los textos de la Declaración Universal de los Derechos Humanos. Es decir, deberán obtenerse los valores de la tabla anterior.

# Tarea 2

Crear una clase `Babel` que represente a un conjunto de lenguas. El constructor de la clase permitirá cargar un conjunto de lenguas a partir de un directorio que deberá contener un fichero de texto denominado `languages.txt`. Este fichero deberá contener información sobre el conjunto de lenguas, una por fila. El primer campo (separado por un espacio) será el identificador de la lengua, y el resto de la línea será una descripción breve:

```
bsq Basque (Euskara)
chn Chinese (Mandarin)
ger Deutsch (German)
gls Gàidhlig Albanach (Scottish Gaelic)
…
```

En ese mismo directorio existirá un subdirectorio llamado `txt`, que deberá contener tantos ficheros de texto como lenguas contenga el fichero índice, y cuyos nombres coincidirán con los identificadores de cada lengua. Es decir, deberán existir los ficheros (los nombres son relativos a la carpeta que contiene el fichero `languages.txt`):

```
txt\bsq.txt
txt\chn.txt
txt\ger.txt
txt\gls.txt
…
```

Puede comprobar que esta estructura es exactamente la de lacarpeta que contiene la Declaración Universal de los Derechos Humanos en 281 lenguas (descargada en la Tarea 1). El constructor de la clase Babel deberá parecerse a:

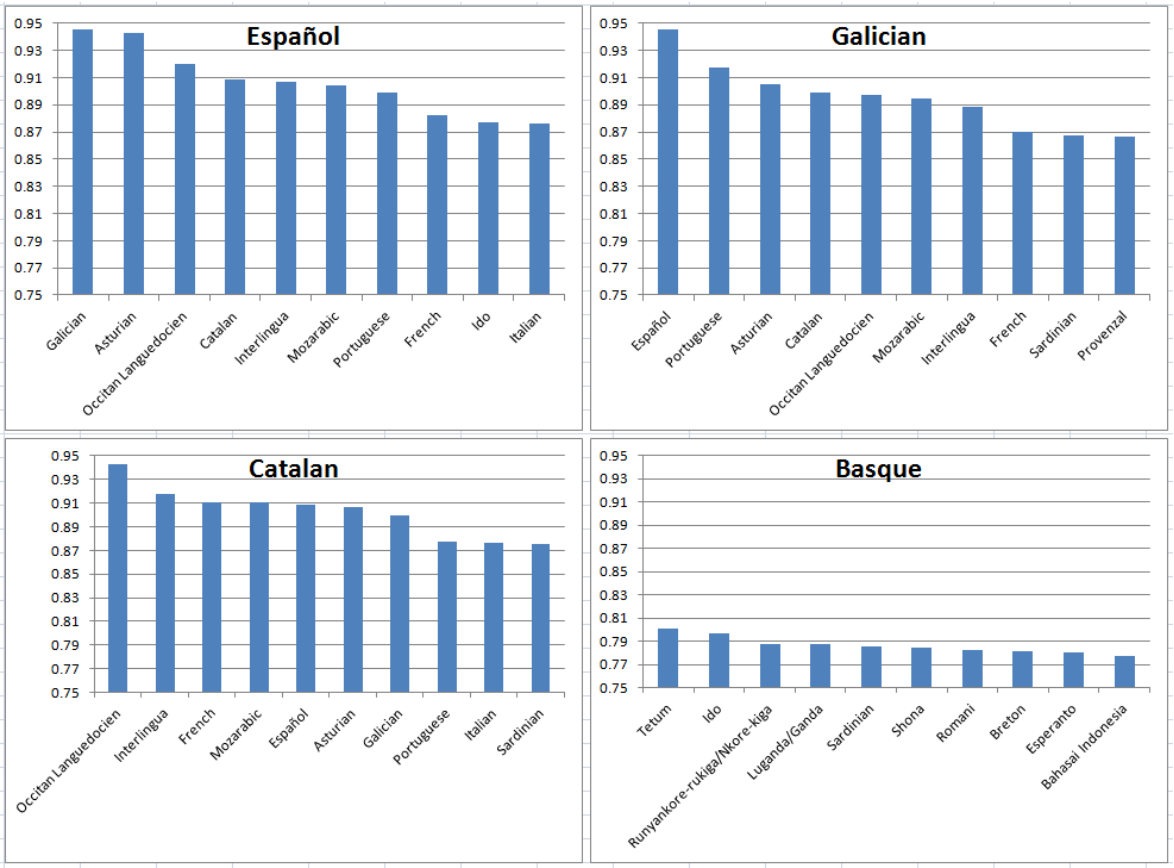
```
public Babel(String dirName){…}
```

Además, esta clase deberá implementar los siguientes métodos:

```
public Language[] findMostSimilars(String id,int n){…}

public static void main(String[] args){…}
```

El método `findMostSimilars` deberá devolver, dado el identificador `id` de una lengua, las `n` lenguas más semejantes a dicha lengua (en orden descendente de semejanza). Por último, el método `main` deberá devolver las 10 lenguas más semejantes a las 4 lenguas oficiales de España. A continuación se muestra una imagen que muestra de manera gráfica el resultado del método:




**Nota**: el método estático

```
java.util.Arrays.sort(T[] a, Comparator<? super T> c)
```

ordena el array **`T`** de manera ascendente, haciendo uso del comparador **`c`** suministrado.