1. Introducción

En esta práctica hallaremos el código de *Huffman* binario de las variables aleatorias $S_{English}$ y $S_{Español}$ con la distribución de probabilidad respecto a los textos almacenados en "auxiliar_en_pract2.txt" y "auxiliar_es_pract2.txt" respectivamente. También utilizaremos este código para codificar y decodificar palabras.

2. Material empleado

Para esta práctica teníamos un código previo que dado un texto extraía las probabilidades de aparición de cada carácter y generaba el árbol del algoritmo de *Huffman* (como un array de diccionarios de dos elementos que representan a los hijos de cada nodo).

2.1. Hallar el código binario de las dos variables, sus longitudes medias y comprobar que satisfacen el Primer Teorema de Shannon

Para hallar el código de cada carácter dado el árbol, lo que hacemos en la función *extract_code* es recorrer el array en el que está guardado el árbol desde el final (el nodo raiz) hasta el principio. En cada iteración extraemos las dos claves del nodo y añadimos a cada carácter de cada clave un 0 o un 1 dependiendo de a qué nodo pertenezcan.

Este algoritmo extrae correctamente el código binario de *Huffman* del árbol porque para cada carácter lo que estamos haciendo es ir guardando el camino por el cual se bajaría en el árbol hasta llegar a su nodo hoja correspondiente.

Para calcular la longitud media simplemente aplicamos la definición
$L(C) := \frac{1}{W}\sum_{i=1}^{N} w_i|c_i|$ donde $W = \sum_{i=1}^{N} w_i$ es la suma de las frecuencias de cada carácter (en este caso es 1 porque son frecuencias relativas) y $|c_i|$ es la longitud de la cadena que representa a cada carácter.

El cálculo de la entropía también lo hacemos aplicando la definición
$H := -\sum_{j=1}^{N} P_j \log_2 P_j$ donde P_j es la probabilidad de cada carácter.

2.2. Codificar la palabra "fractal" y comprobar la eficiencia de longitud con el código binario usual

Codificar una palabra es muy sencillo ya que teniendo el código de cada carácter lo único que tenemos que hacer es ir añadiendo el código de cada letra.

Para calcular la longitud con el código binario usual necesitamos saber la longitud de cada carácter y esta es $l := \lceil \log_2 N \rceil$ donde N es el número de carácteres del lenguaje. Después multiplicamos por el número de carácteres de la palabra y ya tendríamos la longitud de la palabra.

2.3. Decodificar la palabra "10101000011111011111100" del inglés

Este apartado es similar al anterior, lo único que tenemos que hacer es dar la vuelta al diccionario anterior y decodificar a partir de este nuevo. Este diccionario se puede crear porque cada clave también era única (dos letras no podían tener la misma codificación). Para decodificar, como no sabemos la longitud de cada letra tenemos que ir recorriendo la palabra hasta que una parte coincida con un carácter en el diccionario. Este método funciona porque la clave de un carácter nunca es subclave de otro porque es equivalente a llegar a un nodo hoja (que no tiene hijos).

2.4. Calcular el índice de Gini y la diversidad ${}^2D$ de Hill de $S_{English}$

3. Resultados

4. Conclusión

5. Código