Trabajo Integrador 2

Informe

Por:
Barriga Nahuel,
Bedini Tomas,
Hernandez Julieta.

Fecha de entrega: 17/10/2023

Introducción

El presente informe tiene como objetivo detallar el trabajo práctico realizado, donde se utilizaron diversas herramientas para analizar secuencias de código proporcionadas en archivos de texto. A lo largo del informe, se describirán las diferentes etapas del procesamiento del código, desde el análisis de palabras hasta la evaluación de la codificación resultante. Asimismo, se aplicarán conceptos teóricos sobre codificación y análisis de información para comprender mejor los fundamentos de la teoría de la información y su aplicación práctica en la codificación de datos.

Análisis de la fuente de informacion

Probabilidad de aparición de las palabras - Alfabeto código

Inicialmente se buscó la forma de parsear el archivo de texto con el objetivo de obtener todos las palabras dentro del código. Esto se realizó abriendo el archivo y leyéndolo en una lista, luego aplicando la función np.split(" ") , la cual divide la lista en diferentes elementos según la aparición de espacios.

Para generar un diccionario se realizó una función la cual iteraba a través de la lista de palabras, a medida que el programa iba detectando palabras nuevas, las agregaba a un DataFrame y en el caso que no sean nuevas se aumentaba su cantidad de apariciones, también llevaba un contador de palabras a medida que se recorría el archivo.

Una vez finalizado el recorrido y obtenido el DataFrame se generó una tercera columna dentro del mismo con las probabilidades de aparición de cada palabra.

Luego se creó una función que reiteraba las palabras dentro del DataFrame y las descomponía en letras, generando así el alfabeto código**.**

Entropía de la fuente - Longitud media del código

Se solicitó hallar la entropía (E) del código, la entropía es una medida clave que cuantifica la incertidumbre asociada a un sistema de información. Se calcula utilizando la fórmula de Shannon:

$$E = \sum(P_i * \ log_l(\frac{1}{P_i}))$$

Donde Pi es la probabilidad de ocurrencia de cada símbolo en el alfabeto y l el largo del alfabeto.

La entropía es máxima cuando todos los símbolos son igualmente probables, lo que significa que no hay información predecible en el sistema. En contraste, la entropía es mínima cuando un símbolo es completamente predecible, es decir, ocurre con probabilidad 1.

Luego, se calcula la longitud media del código, esto se refiere a la cantidad de bits necesarios para representar una palabra codificada en el código, esto es esencial para determinar la eficiencia de la codificación. Esta se calcula sumando el producto de la longitud de cada palabra por su probabilidad de aparición, tal que:

$$L = \sum(p_i * l_i)$$

Para ello bastó con iterar el *DataFrame* calculando el largo de cada palabra y multiplicándolo por la probabilidad de aparición de la palabra.

Inecuaciones de Kraft y McMillan

Se buscó analizar si la codificación cumplía con las inecuaciones de Kraft y de McMillan. La ecuación de Kraft es una condición necesaria para que un código sea instantáneo. Esta desigualdad establece que la suma de las longitudes de código inverso para todos los símbolos debe ser menor o igual a 1.

$$\sum_{i=1}^{q}(r^{-l_i}) \leq 1$$

Siendo r el número de símbolos diferentes que constituyen el alfabeto código. La desigualdad de Kraft garantiza que las palabras codificadas no se superpongan y que el código sea único y decodable sin ambigüedad.

En el caso que se determine que el código cumple con la inecuación de Kraft, se buscará determinar si el código es instantáneo, es decir, si ninguna palabra codificada es un prefijo de otra. Esto se logra iterando las palabras dentro del *DataFrame* y comparándolas con las del principio del resto de las palabras.

Código instantáneo y/o compacto

Se busca analizar si el código es compacto. Un código se considera compacto cuando la longitud de la palabra codificada para cada símbolo es menor o igual al logaritmo en base r del inverso de su probabilidad, donde r es la base del código. Tal que: 4

$$l_i \leq \log_r(\frac{1}{p_i})$$

Conclusiones

En general, este trabajo práctico nos permitió aplicar conceptos teóricos sobre codificación y análisis de información a un caso práctico. A través de la implementación de las diferentes etapas del procesamiento de un código, logramos comprender mejor los fundamentos de la teoría de la información y su aplicación en la codificación de datos.