

Análisis de la eficiencia del programa codificador

Mateo Restrepo 201226523

October 13, 2015

1 Análisis basado en los primeros 10 Capítulos del Principe de Maquiavelo

Se codificó con el script *compresor.py* los primeros diez capítulos del Principe, 16382 caracteres, con los siguientes resultados

1.1 Código plano

La longitud de l de la palabra de acuerdo al alfabeto \mathcal{A} fue de,

$$\lceil \log_2 |\mathcal{A}| \rceil = 7.0$$

La cantidad de **bits** que resultaron de comprimir el archivo en la modalidad binaria plana fueron,

$$\#\mathbf{bits}_{\text{plain}} = 114675$$

1.2 Código de Huffman

La entropía de la distribución de frecuencias,

$$H(f_i) = \sum_i f_i \log_2 \frac{1}{f_i} = 4.387248326$$

La cantidad de **bits** que resultaron de comprimir el archivo en la modalidad huffman fueron,

$$\#\mathbf{bits}_{\text{huffman}} = 72056$$

1.3 Comparación

Comparando tenemos las siguientes razones,

$$\frac{H(f_i)}{\lceil \log_2 |\mathcal{A}| \rceil} = 0.626749714$$

$$\frac{\#\mathbf{bits}_{\text{huffman}}}{\#\mathbf{bits}_{\text{plain}}} = 0.628349683$$

Luego de estas razones podemos deducir que el codificador es muy bueno ya que las razones difieren por aproximadamente 2%, pero como era de esperarse aunque el algoritmo de Huffman es extremadamente eficiente no es perfecto en todos los casos, ya que existe la posibilidad que se agregue un bit de más en la codificación haciendo el mensaje más largo.