CMPE-365 - Assignment 3

Robert Moore
10179621

Due Date, November 2, 11:59 PM

Part 2:

In Table 1 is a comparison of the three canonical collections and the effectiveness of their respective Huffman codes. Codes were constructed based on the frequency of characters of all files in each canonical collection. Each code was then used so encode all elements of Data.zip and the compressed size divided by 8 was compared to the original size.

Collection #	size of collection (kB)	size of encoded Data (Bytes)	simulated size of encoding (Bytes)	compression ratio (original size=2296000 Bytes)
1	396	29548000	3693500	1.608667247
2	44	12116000	1514500	0.659625436
3	1548	10808000	1351000	0.588414634

Table 1: Comparison of Compression Rates for Collections 1, 2, and 3.

We see that collection 1 does very poorly, increasing the size of Data.zip by over 60%. we see that collection 2 does well, reducing the size of Data.zip to around 66% of its original size. Collection 3 does the best achieving a reduction to less than 59% of the collections original size.

This difference is explained by examining the contents of the three collections as well as Data.zip. Data.zip is a collection of long texts in well structured English. This is most closely matched by collection 3 where we see another collection of long texts. We can assume that these two groups will have a similar frequency distribution across all printable ascii characters. This is supported by the high efficiency of the Huffman code generated from the collection and used on Data.zip.

The next closest to Data.zip is collection 2 which is a collection of short texts. Although this is still English it is structured quite differently to Data.zip and collection 1 and therefor has a slightly different frequency distribution amongst all printable ascii chars. Again, this is supported by the still decent but not optimal compression ratio of ~0.65.

Lastly, Collection 1 is simply a collection of English words separated by new lines. The new line characters alone throw off the frequency distribution significantly. This in addition to the lack of spaces and filler words such as ‘the’ and ‘I’ further push the frequency distribution away from Data.zip. This is why the code generated from collection 1 does the worst and even increases the size of the files.