

Lab #7

Peter Naumoff

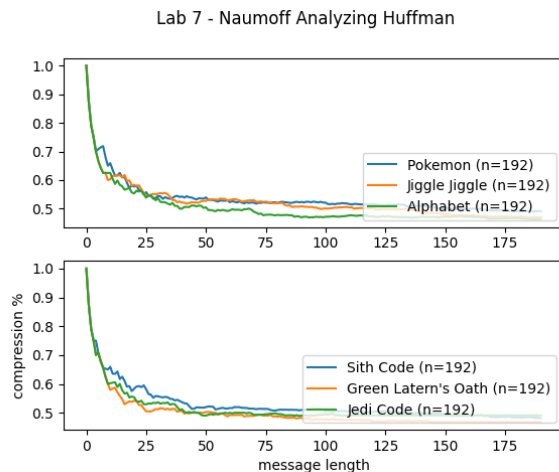
Department Of Computer Science, California Polytechnic State University

CSC 202, Data Structures

Professor Nick Stapleton

June 7, 2023

For your (i) songs and (ii) mantras, how can you describe compression performance from Huffman?



(i) Songs

The average length of the songs are much longer. It also seems that there are more unique vocabulary words, but also some repetition between some of the words. This repetition leads to a higher frequencies of the same letters in the songs and thus leads to a higher compression rate and overall effectiveness of the Huffman algorithm. This is reflected in my graph as there is a high amount of compression even as the song length gets long. 50% compression is very satisfactory and the compression is even better for shorter lengths of songs. One interesting thing to note is that the compression gets slightly better around the mid length of the song. My theory on this is that because songs are usually structured in a way that there is a chorus in the beginning, a part in the middle, and then the chorus following the compression ends up being better because the first and last parts of the song are basically the same. This means there are more characters that are similar and thus the compression percentage goes up slightly. This benefit is lost as the song length goes on as the compression performance dips from the increased frequency of unique vocabulary words and a higher overall spread of letters.

(ii) Mantras

Mantras are different in the most obvious way in that they have more repeated words which means there is less of a spread between the frequencies of the characters that appear. Consequently, this means that the Huffman algorithm performs better and this is reflected in the graph where it is shown that the performance is higher around the mid-early message length range. The added benefit of the repeated words and characters in the Mantras is that searching is way more simple since the height of the Huffman tree is smaller and there are less nodes to parse through and search over.