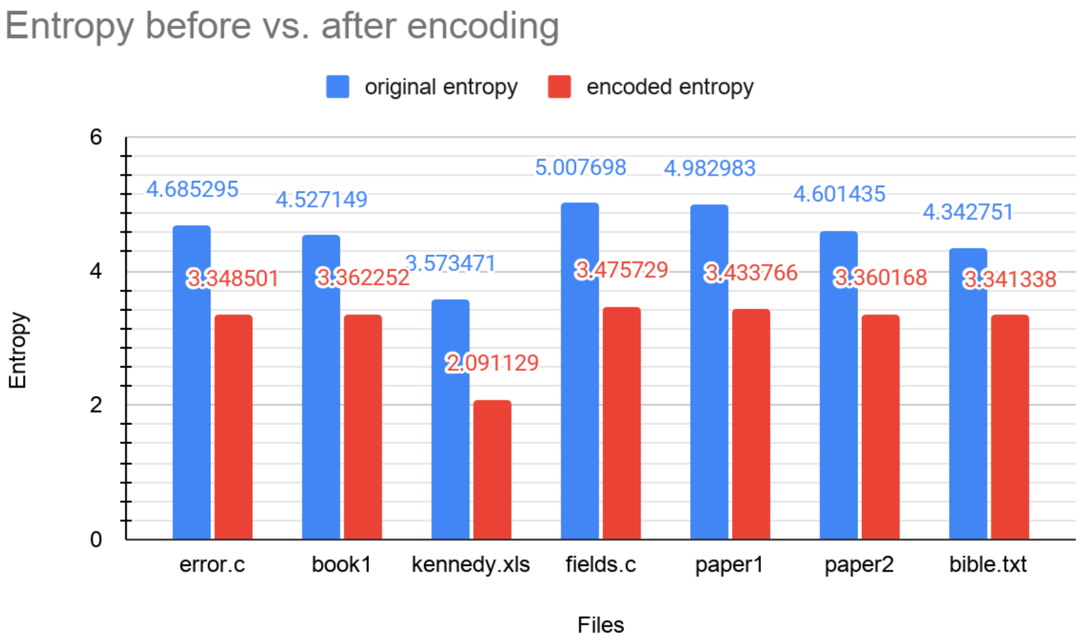
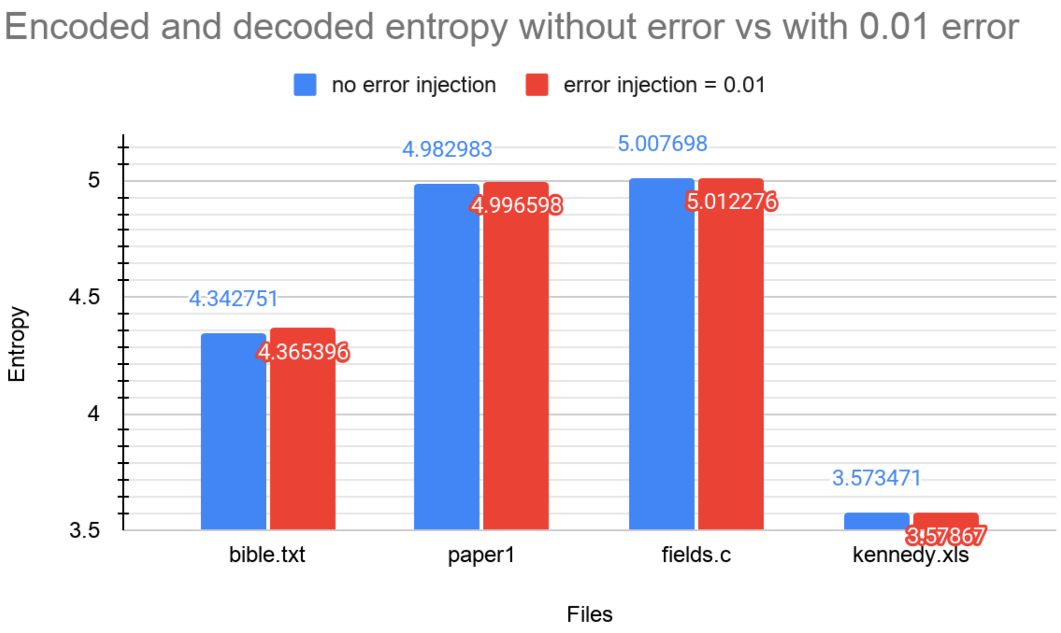
Ravjodh Heer

Assignment 5 - Hamming Codes

Entropy before vs after encoding graph:



Encoded and decoded entropy before vs after error injection graph:



Analysis:

Entropy is the amount of certainty that we can have of a file's contents. For example, a file containing aaaaaaa… would have a very low entropy whereas a file with more unique and random contents would have more entropy. In the first graph, I compare how the entropy of various different files compares with each other, however I also compare each file's original entropy to it's entropy after encoding. After encoding, we see a huge decrease in file entropy as shown on the graph since there is more certainty of what characters we would encounter when it comes to encoded message bits which are much less randomized than truly random words. In order to be more statistically accurate, I compared various different files of different sizes and types in order to spread out our sample size as broad as possible. This way we will be able to observe the effects that encoding has on the entropy of different file types and sizes. One of the largest file types, bible.txt, had a surprisingly average entropy compared to other files being tested and similar to them, the entropy was decreased by nearly 25% after encoding. Papers and books were also included to get a sample of short texts and long texts as well. These papers and books it seems, similar to bible.txt, had an average entropy under 5 with ranges going from 4.3 to 4.9. After encoding, however, all of these entropies dropped to about 3.36 with the differences in entropy between them also decreasing heavily. The entropy difference ranged from 4.3 to 4.9 (a range of 0.6) before encoding but after encoding, that range went from 3.34 to 3.43 (a range of 0.09). My hypothesis for why this would be is that when it comes to books and other texts, there is a ceiling of predictability of the words in the text equal to the complexity of the grammar and vocabulary used. With more archaic texts such as bible.txt, we see a lower entropy as there is most likely very little new vocabulary being introduced after so many pages.

Paper1 on the other hand contains various new jargons of programming and new names that are being introduced almost all the time. This constant introduction of new terms and names is what increases the entropy of this to 4.98 while bible.txt stays at a lower 4.34. After encoding however, these drop significantly because the encoded bits are much more easy to predict since the various unique characters and words get translated to message bits that are all related to one another in some way or another which increases predictability. The test between the C files was very interesting as well since the C files seemed to have much more entropy than the other files. A reason for this could possibly be that the C files contain various punctuation and new function declarations and such which are much less predictable than the flow of a sentence or paragraph. These too had their entropy drop by nearly 30% after encoding. The xls file was the most surprising of them all since it had such a low entropy already. The possible reason for this could be that xls files are already encoded and compressed as they are which could explain why the original entropy was about as low as the rest of the files' encoded entropy. After encoding however, this still managed to heavily decrease in entropy. This comparison taught me that entropy can be heavily decreased if you are encoding since it makes the predictability of the file's contents much easier due to similarity in the message bits. The second graph was a test created by me where I encoded and decoded a file and got the entropy of that and recorded the value. This value was then compared to the entropy of an encoded and decoded file with an added noise/error of 0.01. I hypothesized that the added error of 0.01 would increase the noise in the file and cause an increase in entropy since noise equates to unpredictability. My hypothesis ended up being correct since the increase in entropy ended up being about 1% of the original entropy. This value is extremely small and relatively hard to see on the graph but the 1% increase in entropy is directly equal to the 0.01 = 1% error rate. The greater the error rate, the greater the added entropy, therefore, my hypothesis on the increase in entropy relative to the increase of error was correct.