Asgn5 Writeup

ENTROPY VS. ERROR RATE

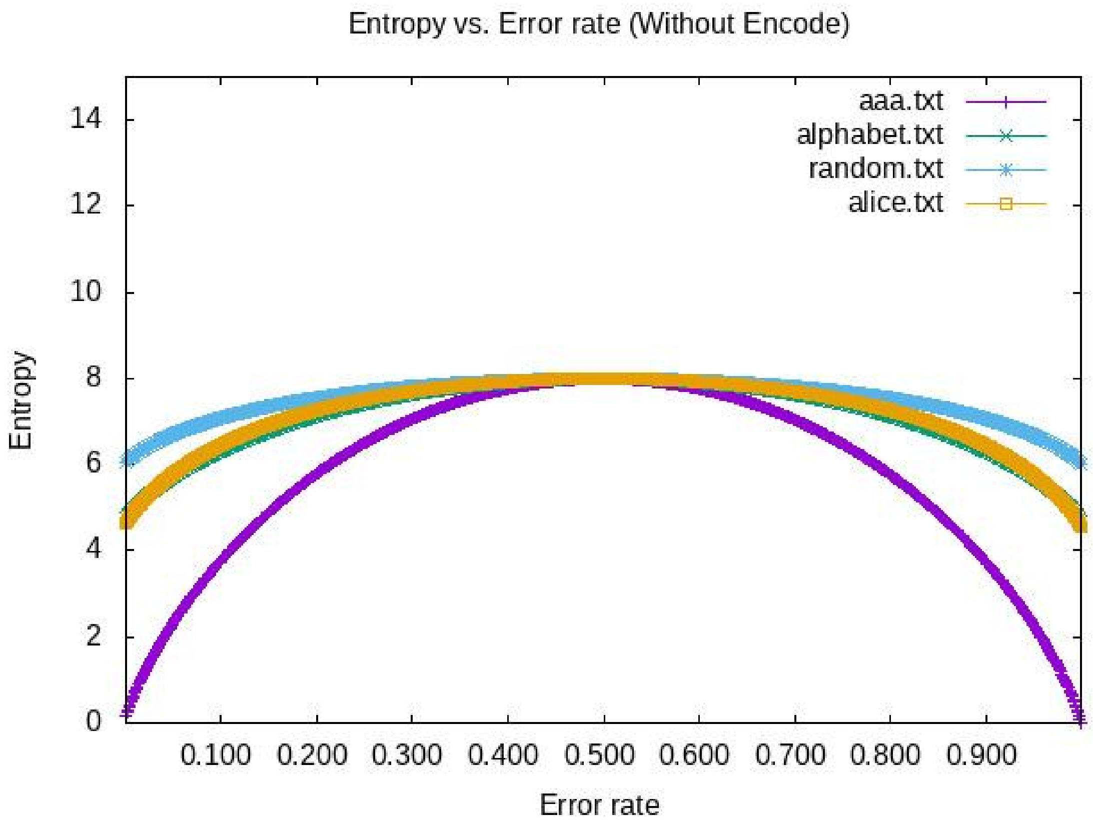

Figure 1.1 Entropy without Encoding (Entropy vs. Error Syndrome)

Definition Entropy: The measurement for the lack of order or predictability; gradual decline into disorder

Figure 1.1 is a representation of the error syndrome (noise) of a file vs. the Entropy of the text **without encoding**. In this graph, the error rate starts from 0.001 and ends at 1.0 with iterations of 0.001. Here, the data that is represented has not been encoded.

With this Entropy vs. Error Syndrome graph (Figure 1.1 and Figure 1.2), we can see four different files. The aaa.txt file contains only lower case As, alphabet.txt has the alphabet (abc), random.txt contains random text including numbers and symbols, then alice.txt is just a regular text file containing the book Alice in wonderland.

Thus, with the definition of Entropy, these different files show us how random they are. As a result, in **Figure 1.1**, aaa.txt has the lowest Entropy since files with all A's are pretty predictable and stable. Furthermore, alphabet.txt and alice.txt fall around the same Entropy (higher than aaa.txt) because they contain primarily alphabetical words, which is not as predictable as aaa.txt. But we can see that alice.txt has a little more entropy than alphabet.txt because alice.txt does not have a sequential alphabetical pattern that alphabet.txt has. Finally, random.txt has the most because it does not have a sequential pattern and has both symbols and numbers, which significantly lowers the predictability of the text.

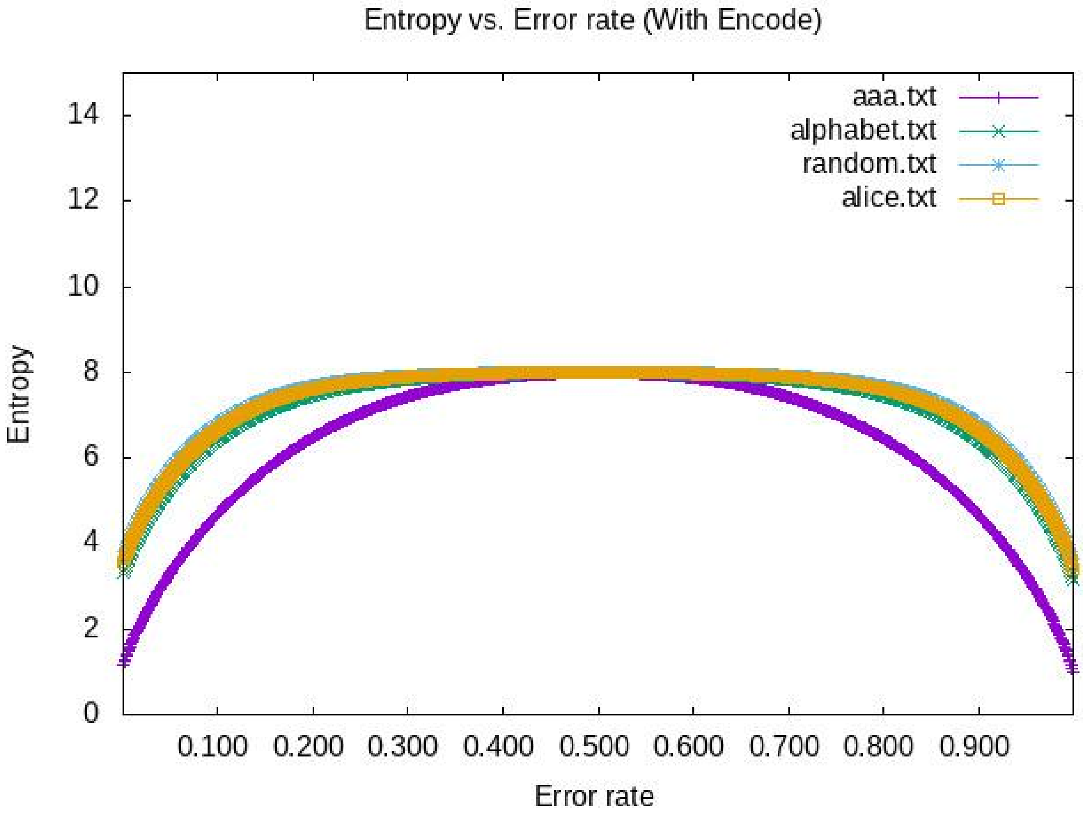

Figure 1.2 Entropy with Encoding (Entropy vs. Error Syndrome)

Figure 1.2 represents the error syndrome (noise) of a file vs. the Entropy of the text **with encoding**. In this graph, the error rate starts from 0.001 and ends at 1.0 with iterations of 0.001. Here, the data that is represented has been encoded.

Unlike Figure 1.1, we can see that the entropy for all files are generally the same except for aaa.txt. This is because the encoding process creates two bytes to represent one character. These two bytes are usually the same for all characters, with just a few differences. As a result, if we look at the structure of a ascii table, the upper nibble is the same for each of the characters in the row.

For example, there are 1/4 possibilities for each character to get the same upper nibble. This lets the encoding process create the same entropy for all files. Also, we can see that representing a character with two bytes makes the entropy increase.

Why is there a bell curve?

The Entropy vs. Error Syndrome graphs shows us a fascinating bell curve. One would think that since the error syndrome increases, the entropy would gradually increase and not create a bell curve.

But, these graphs (Figure 1.1 and Figure 1.2) show us that on the left side of 0.5, there is more of the original text compared to the noise injected into the text. As the error syndrome on the left side increases until 0.5, this median point is when both the noise and the original text are half and half mixed in together, causing predictability to decrease significantly.

Therefore, we can see the pattern:

Data from Entropy without Encoding (Figure 1.1):

75% original text: 25% noise ≈ 2 entropy

50% original text: 50% noise ≈ 8 entropy

25% original text: 75% noise ≈ 2 entropy

Thus, we can see that Entropy can decrease if it interacts with a system that increases somewhere else by at least as much.

Therefore, since the original text transforms into noise, the noise will eventually become its pattern representing the original text. As a result, we can say that between two systems, if one system represents both systems, then there is no entropy. But if both systems represent the same file, there is entropy, and it gets higher when both systems represent the characters in the file evenly.