Consider the noisy typewriter channel, mapping
\[ A_Z = \{A, B, C, \ldots, Y, Z, \} \]
with \(|A_Z| = 27\), into \(A_Y = A_Z\), where each letter is mapped with equal probabilities into the preceding, the following or the same letter (p. 41-44 of the notes). Design an efficient code by which to reliably send symbols from \(A_X = A_Z\) through the channel (i.e., you should be able to send and retrieve a text using the 27 symbols with no error).

**Solution:** We can base an efficient and reliable code by using a subset of non-confusable inputs, e.g.,
\[ |S| = \{B, E, H, K, N, Q, T, W, Z\} \]
with \(|S| = 9\). Since there are 81 sequences of 2 symbols from \(|S|\),
\[ |S^{(2)}| = 81 \]
and \(81 > 27\), we can just select any subset of 27 pairs of symbols from \(S\), i.e.,
\[ S' \subset S^{(2)} \]
with \(|S'| = 27\), to reliably encode the input. For instance, we can divide the 27 letters into 3 groups of 9 letters,
\[ g_1 = \{A, B, \ldots, I\}, \quad g_2 = \{J, K, \ldots, R\}, \quad g_3 = \{S, T, \ldots, Z\} \]
and use the first letter of the code to encode for the group, e.g.,
\[ g_1 \mapsto B, \quad g_2 \mapsto E, \quad g_3 \mapsto H \]
and the second letter of the code to encode for the specific letter within the group,
\[ \{A, B, \ldots, I\} \mapsto \{B, E, \ldots, Z\} \]
\[ \{J, K, \ldots, R\} \mapsto \{B, E, \ldots, Z\} \]
\[ \{S, T, \ldots, Z\} \mapsto \{B, E, \ldots, Z\} \]
arriving at the two-letter code \(E : A_X \rightarrow S^{(2)}\):
\[
\begin{align*}
A & \mapsto BB, \quad B \mapsto BE, \quad C \mapsto BH, \quad D \mapsto BK, \quad E \mapsto BN, \quad F \mapsto BQ, \quad G \mapsto BT, \quad H \mapsto BW, \quad I \mapsto BZ, \\
J & \mapsto EB, \quad K \mapsto EE, \quad L \mapsto EH, \quad M \mapsto EK, \quad N \mapsto EN, \quad O \mapsto EQ, \quad P \mapsto ET, \quad Q \mapsto EW, \quad R \mapsto EZ,
\end{align*}
\]
S↦HB, T↦HE, U↦HH, V↦HK, W↦HN, X↦HQ, Y↦HT, Z↦HW, −↦HZ,

The average length of this code is 2, irrespective of the input distribution. The rate is

$$R = \frac{K}{n} = \frac{\log 27}{2} = \frac{3 \log 3}{2}$$

We know that the channel capacity is

$$C = \max_Z I[Z : Y] = 2 \log 3 > R = \frac{3}{2} \log 3$$

which is more than the channel capacity. When using the channel at its capacity, we should use 1.5 letters to encode a single letter (not 2). We know that we can achieve the capacity by encoding sequences of messages $A_X^{(m)}$ into non-confusable sequences of inputs strings $A_Z^{(n)}$, and in the limit $n \to \infty$, we should communicate information at the channel capacity. In fact, for this simple case it is sufficient to consider $m = 2$, $n = 3$. For $n = 3$, we can consider the set of sequences of 3 non-confusable input symbols,

$$Z^{(3)} \in S^{(3)}$$

with $|S^{(3)}| = 9^3 = 729$. These sequences are sufficient to reliably encode $27^2 = 729$ symbols $x^{(2)} \in A_X^{(2)}$. In other words, we can map a pair of letters $X^{(2)} \in A_X$ from the 27-letter alphabet into a triple of letters from $S^{(3)}$, e.g.

AA↦BBB, AB↦BBE, AC↦BBH, ... −U↦ZZT, −Z↦ZZW, −−↦ZZZ

This is $(n, K)$ block code with $n = 3$ and $K = \log 729 = 6 \log 3$. Hence the rate is

$$R = \frac{K}{n} = 2 \log 3 = C$$

achieving the channel capacity. In this code, we use only 1.5 letters to encode a letter.