Many fields such as the first name and last name occasionally contain minor typographical variations or errors. Even with high-quality lists, such as the 1990 US Census and its PES, there were areas of the United States in which 30% of the first names and 25% of the last names of individuals who were in fact matches did not agree exactly on a character-by-character basis. If we attempt to match two such records and at least one of the first names on the two records has a typographical variation, then we may fail to match two records that may indeed be matches. Using the terminology of computer science, we can consider these records or their components to be strings— that is, strings of alphanumeric characters. We need a practical method for dealing with such situations.

As the name indicates, string comparator metrics are used to compare two strings. In particular, they are used to determine how much alike the two strings are to each other. Common practice is to restrict the values of the metrics to the interval from zero to one; here, one indicates perfect agreement (the two strings are identical) and zero indicates that they are highly dissimilar, the extreme case being that they have no characters in common. These values are needed to adjust the likelihood ratios of the Fellegi–Sunter scheme to account for this partial agreement. In this work we focus on the string comparator metric introduced by Jaro and enhanced by Winkler. Current research in this area is described in Cohen, Ravikumar, and Fienberg [2003a, b].

13.1. Jaro String Comparator Metric for Typographical Error

Jaro [1972, see also 1989] introduced a string comparator metric that gives values of partial disagreement between two strings. This metric accounts for the lengths of the two strings and partially accounts for the types of errors— insertions, omissions, or transpositions— that human beings typically make when constructing alphanumeric strings. By transposition we mean that a character from one string is in a different position on the other string. For example, in comparing “sieve” to “seive,” we note that “i” and “e” are transposed from one
string to the other. The string comparator metric also accounts for the number of characters the two strings have in common. The definition of common requires that the agreeing characters must be within half of the length of the shorter string. For example, “spike” and “pikes” would only have four characters in common because the “s’s” are too far apart.

Specifically, let \( s_1 \) denote the first string, \( s_2 \) denote the second string, and let \( c \) denote the number of characters that these two strings have in common. Then, if \( c > 0 \), the Jaro string comparator metric is

\[
\Phi_j(s_1, s_2) = W_1 \cdot \frac{c}{L_1} + W_2 \cdot \frac{c}{L_2} + W_t \cdot \frac{(c - \tau)}{c}
\]

where

- \( W_1 \) is the weight assigned to the first string,
- \( W_2 \) is the weight assigned to the second string,
- \( W_t \) is the weight assigned to the transpositions,
- \( c \) is the number of characters that the two strings have in common,
- \( L_1 \) is the length of the first string,
- \( L_2 \) is the length of the second string, and
- \( \tau \) is the number of characters that are transposed.

We require that the weights sum to 1: \( W_1 + W_2 + W_t = 1 \).

Finally, if \( c = 0 \), then \( \Phi_j(s_1, s_2) = 0 \).

Example 13.1: Using the Jaro string comparator metric on Higvee versus Higbee

Let the first string, \( s_1 \), be “Higbee” and the second string, \( s_2 \), be “Higvee,” and let all of the weights be equal to 1/3. Find the value of the Jaro string comparator metric for these two strings.

Solution

Because the two have five of six letters each in common, we have

\[
L_1 = L_2 = 6, \quad c = 5, \quad \text{and} \quad \tau = 0.
\]

Hence,

\[
\Phi_j(s_1, s_2) = W_1 \cdot \frac{c}{L_1} + W_2 \cdot \frac{c}{L_2} + W_t \cdot \frac{(c - \tau)}{c} = \left( \frac{1}{3} \right) \cdot \left( \frac{5}{6} \right) + \left( \frac{1}{3} \right) \cdot \left( \frac{5}{6} \right) + \left( \frac{1}{3} \right) \cdot \left( \frac{5 - 0}{5} \right) = \frac{8}{9}.
\]

Example 13.2: Using the Jaro string comparator metric on Shackleford versus Shackelford

Let the first string, \( s_1 \), be “Shackleford” and the second string, \( s_2 \), be “Shackelford” and let all of the weights be equal to 1/3. Find the value of the Jaro string comparator metric for these two strings.