

The Damerau-Levenshtein distance function is a modified version of the Levenshtein distance[3], which is a widely used to compute the edit-distance of two discrete sequences[1, 4]. The most important applications are within the Natural Language Processing (NLP) discipline and the Biomedical Sciences. Within these areas, we often use the Levenshtein distance to compute the edit-distance between two words, two sentences or two DNA sequences. Note, that the elements of these sequences are often atomic symbols instead of multidimensional vectors. Generally, the distance accounts for inserts, deletions and substitutions of elements between the two sequences. Damerau modified the distance function to allow for transposition operations. For Process Mining, transpositions are important as one event can transition into two events that are processed in parallel and may have varying processing times. In Figure 1, we schematically show two sequences and their distance.

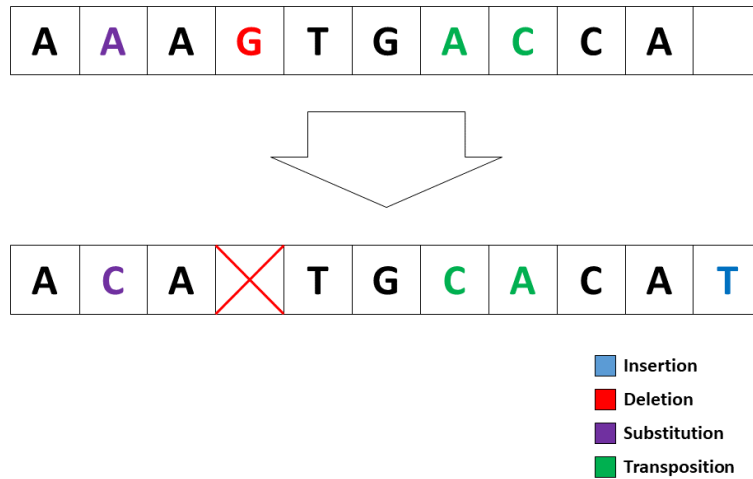


Figure 1: Shows two sequences. The edit distance is the sum of multiple operations. Blue shows an insert, red a deletion, purple a substitution and green a transposition. Therefore the edit distance is 4.

Equation 1 depicts the recursive formulation of the distance. The distance computes the costs of transforming the sequence  $a$  to  $b$ , by computing the

minimum of five separate terms.

$$d_{a,b}(i, j) = \min \begin{cases} d_{a,b}(i-1, j) + 1 & \text{if } i > 0 \\ d_{a,b}(i, j-1) + 1 & \text{if } j > 0 \\ d_{a,b}(i-1, j-1) + 1 & \text{if } i, j > 0 \\ d_{a,b}(i-2, j-2) + 1 & \text{if } i, j > 1 \wedge a_i = b_{j-1} \wedge a_{i-1} = b_j \\ 0 & \text{if } i = j = 0 \end{cases} \quad (1)$$

The recursive form  $d_{a,b}(i, j)$  for sequences  $a$  and  $b$  with respective elements  $i$  and  $j$  takes the minimum of each of each allowed edit operation. In particular, no change, deletion, insertion, substitution and transposition. For each operation, the algorithm adds an edit cost of 1.