3.1.1 Mathematics of Similarity Score

Similarity Score is used to measure the difference between two versions of UI segmentation annotation on the same text. It is represented as a percentage, the larger the number means that the two versions of annotation are more similar. The formular for the Similarity score is:

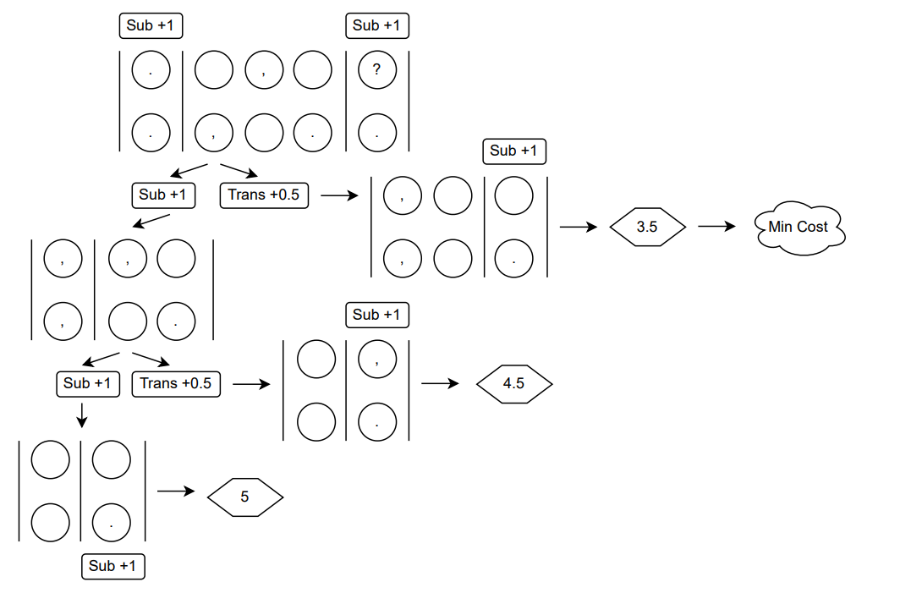
}.

There is a potential UI boundary between each two units after tokenization. There are three kinds of potential boundaries: no boundary (empty space), punctuations (",", ".", "?", "--"), boundaries with no punctuation (";"). The total number of the three types of potential boundaries is the Number of Potential Boundaries in the formular.

Minimum Cost of Transformation represents the minimum number of actions to change one of the annotations same to the other one. There are generally two types of defined actions in our case: Substitution and Transition. Substitution means changing one boundary to a different one at the same place, and it is defined to have a cost of 1. Transition means interchange the boundary with the boundary one space before or after, and it is defined to have 0.5 cost. Using these two types of actions, our function tries to find the minimum cost to change one annotation to the other one.

3.1.2 R Algorithm

In our function, we calculate the cost for each speaker separately and calculate the sum of them. For each speaker, we separate the long list into some small segmentations as well. If two none empty boundaries appear at the same place, no matter they are same or not, we separate at this place, and we only consider substitution action at this place. Therefore, we have several shorter lists between those none empty boundaries at the same places.

 For each of these shorter lists, we run a recursive algorithm to find the minimum cost the list. This is basically a binary process, for each place in the shorter lists, we try two actions to get a new list: directly substitute boundary of one annotation with the boundary in the other annotation, or interchange the boundary with the next boundary in the same annotation. Then we try to compare the new list with the list of second annotation to get the smallest cost for each step.