A Appendix

Algorithm 1: Python code of heuristic weight function

```
steps_overlap = 20
steps_agreement = 11

def getHeuristicWeight(overlap , agreement):
    # create empty matrix
    matrix = numpy.zeros(shape=(steps_overlap , steps_agreement))

    length_row = len(matrix)
    length_column = len(matrix[0])

    # define boundary points
    matrix[length_row -1][0] = 0
    matrix[length_row -1][length_row -1] = 1
    matrix[0][0] = 0.2
    matrix[0][length_column -1] = 0.6

    # fill first and last row of matrix
    for i in range(1,length_column -1):
        matrix[0][i] = matrix[0][0] + (matrix[0][length_column -1]
                            - matrix[0][0])*i/(length_column -1)
        matrix[length_row -1][i] = matrix[length_row -1][0] +
                            (matrix[length_row -1][length_column -1]
                            -matrix[length_row -1][0])*i/(length_column -1)

    # fill rest of the matrix
    for r in range(1,length_row -1):
        for c in range(0,length_column):
            matrix[r][c] = matrix[0][c] + (matrix[length_row -1][c]
                            - matrix[0][c])*r/(length_row -1)

    # read the weight for the given overlap and agreement rate
    pos_overlap = int(overlap)-1
    pos_agreement = int(round(agreement*10))
    return matrix[pos_overlap][pos_agreement]
```