CSE 654_484 NATURAL LANGUAGE PROCESSING

HW 1

REPORT 1

Merve Horuz

1801042651

Edit distance method and Smith Waterman algorithm are used to find similar text sections between documents.

The S-W Algorithm implements a technique called dynamic programming, which takes alignments of any length, at any location, in any sequence, and determines whether an optimal alignment can be found. Based on these calculations, scores or weights are assigned to each character-to-character comparison: positive for exact matches/substitutions, negative for insertions/deletions. In weight matrices, scores are added together and the highest scoring alignment is reported.

In essence, we are converting one string (sequence of letters) into another string by performing certain operations on the individual characters that make up that string. We can insert a character or delete a character from the first string, or we can substitute a character in the first string with a character from the second string. Thus, an insertion into one string results in the simultaneous deletion from the second string. What we call an edit distance, is just the minimum number of operations we perform to convert one string into another. So the similarity of two strings is simply the value of the alignment between the two strings that maximizes the total alignment value (optimal alignment value), or the highest score given. Global alignment is obtained by first inserting chosen spaces (or dashes) either into or at the ends of the two strings, and then placing the two resulting strings one above the other so that every character or space in either string is opposite a unique character or a unique space in the other string.

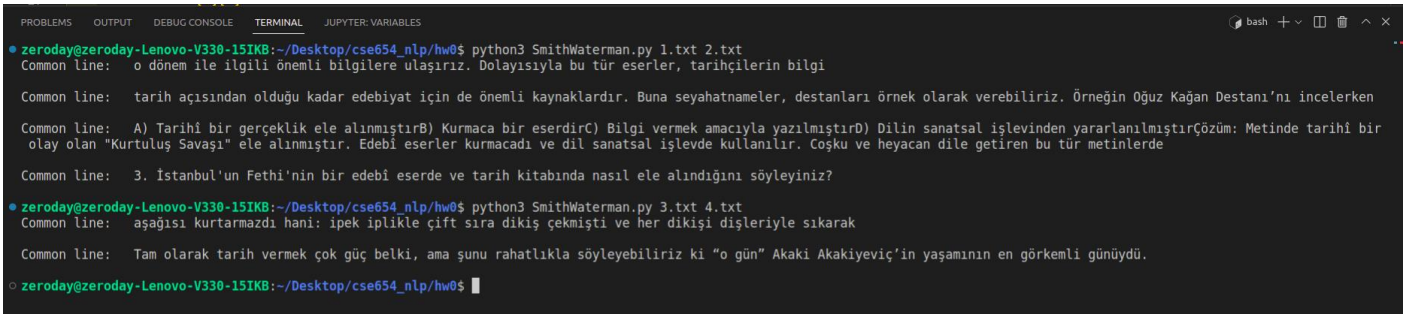
To solve this problem, I first started by filling the matrix array with scores. at this stage:

- Insertion -3
- Deletion -3
- Mismatch -3
- Match 3

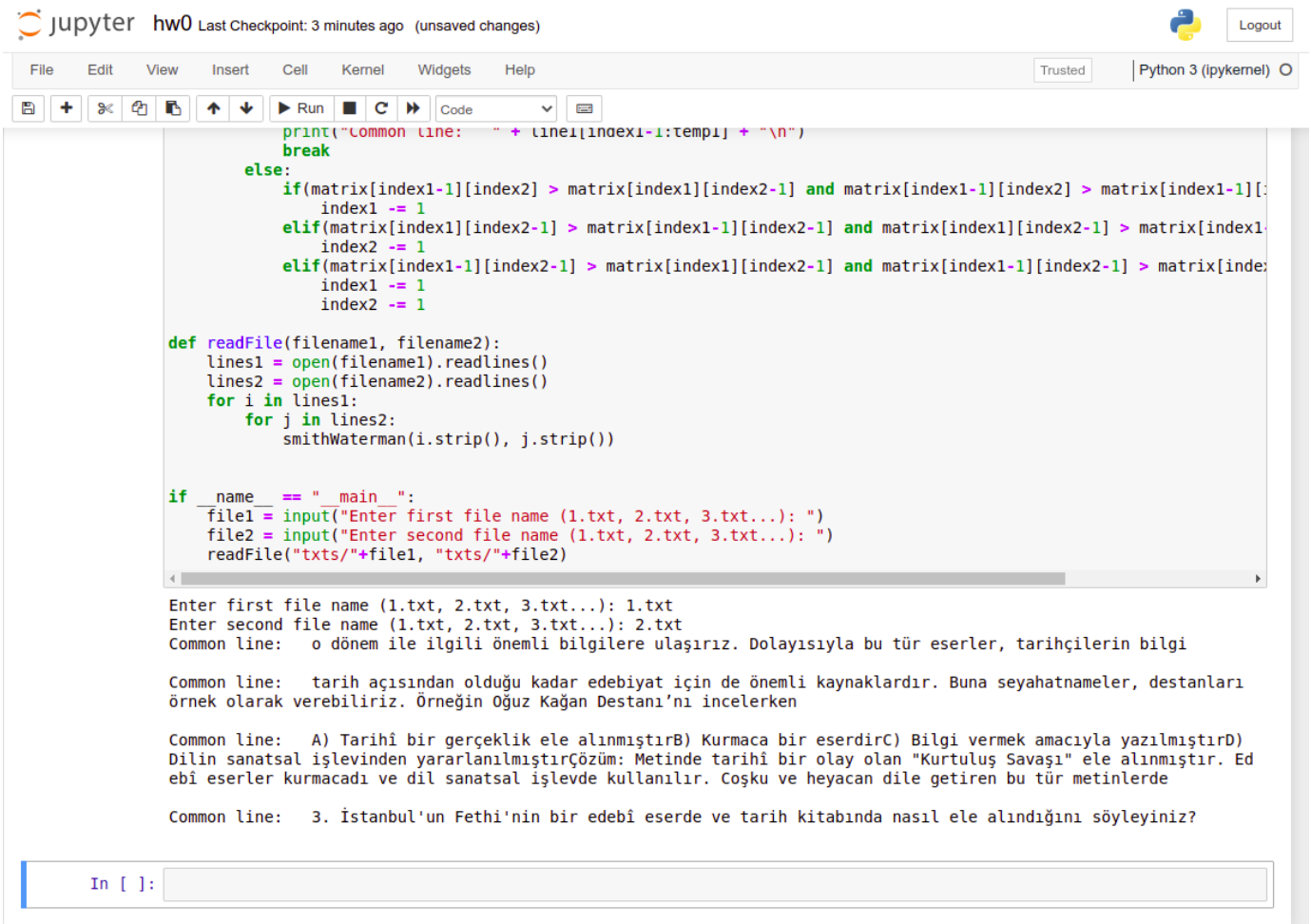
It starts upper left position of matrix. If it continues to the right or left of the matrix, the insertion cost is considered, if it continues up or down, the deletion cost is considered, if it goes diagonally, the match or mismatch cost is considered, and the maximum of these 4 integers is the score of the cell, and in this way, all the cells of the matrix are filled and go to the end.

Then, traceback algorithm is ran.
The largest element in the matrix is found. Starting from the indexes of the location of the maximum element, traceback is started. If we call the location of the maximum element the x and y coordinates, the locations we will look for traceback are [x-1][y], [x][y-1] and [x-1][y-1]. It goes to the largest of these three coordinates and iterates this loop until the three coordinates are zero. The coordinates of where it stops are where the common text ends. By the way, the starting coordinates are also kept in spare variables, so that common text is found according to the indexes of the all text.

For example, common lines between the given two texts are found in terminal:



In jupyter notebook:



In this matrix 36 is the maximum element. So, traceback algorithm finds the maximum values in order: 36, 33, 30, 27, 24, 21, 18, 15, 12, 9, 6 and 3. Common text is “sağlamıştır.” .

```
0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
0.0     3.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
0.0     0.0     6.0     3.0     0.0     3.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
0.0     0.0     3.0     9.0     6.0     3.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
0.0     0.0     0.0     6.0     12.0    9.0     6.0     3.0     0.0     0.0     0.0     0.0     0.0
0.0     0.0     3.0     3.0     9.0     15.0    12.0    9.0     6.0     3.0     0.0     0.0     0.0
0.0     0.0     0.0     0.0     6.0     12.0    18.0    15.0    12.0    9.0     6.0     3.0     0.0
0.0     0.0     0.0     0.0     3.0     9.0     15.0    21.0    18.0    15.0    12.0    9.0     6.0
0.0     0.0     0.0     0.0     0.0     6.0     12.0    18.0    24.0    21.0    18.0    15.0    12.0
0.0     0.0     0.0     0.0     0.0     3.0     9.0     15.0    21.0    27.0    24.0    21.0    18.0
0.0     0.0     0.0     0.0     0.0     0.0     6.0     12.0    18.0    24.0    30.0    27.0    24.0
0.0     0.0     0.0     0.0     0.0     0.0     3.0     9.0     15.0    21.0    27.0    33.0    30.0
0.0     0.0     0.0     0.0     0.0     0.0     0.0     6.0     12.0    18.0    24.0    30.0    36.0
Common line:    sağlamıştır.
```

In this matrix 33 is the maximum element. So, traceback algorithm finds the maximum values in order: 33, 30, 27, 24, 21, 18, 15, 12, 9, 6 and 3. Common text is “sen gittin. ” .

```
0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
0.0     3.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
0.0     0.0     6.0     3.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
0.0     0.0     3.0     9.0     6.0     3.0     0.0     0.0     0.0     0.0     3.0     0.0
0.0     0.0     0.0     6.0     12.0    9.0     6.0     3.0     0.0     0.0     0.0     0.0
0.0     0.0     0.0     3.0     9.0     15.0    12.0    9.0     6.0     3.0     0.0     0.0
0.0     0.0     0.0     0.0     6.0     12.0    18.0    15.0    12.0    9.0     6.0     3.0
0.0     0.0     0.0     0.0     3.0     9.0     15.0    21.0    18.0    15.0    12.0    9.0
0.0     0.0     0.0     0.0     0.0     6.0     12.0    18.0    24.0    21.0    18.0    15.0
0.0     0.0     0.0     0.0     0.0     3.0     9.0     15.0    21.0    27.0    24.0    21.0
0.0     0.0     0.0     0.0     0.0     0.0     6.0     12.0    18.0    24.0    30.0    27.0
0.0     0.0     0.0     3.0     0.0     0.0     6.0     12.0    18.0    24.0    30.0    27.0
0.0     0.0     0.0     0.0     0.0     0.0     3.0     9.0     15.0    21.0    27.0    33.0
Common line:    sen gittin.
```

In this matrix 33 is the maximum element. So, traceback algorithm finds the maximum values in order: 33, 30, 27, 24, 21, 18, 15, 12, 9, 6 and 3. Common text is “ben geldim.” .

```
0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
0.0     3.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
0.0     0.0     6.0     3.0     0.0     0.0     3.0     0.0     0.0     0.0     0.0     0.0
0.0     0.0     3.0     9.0     6.0     3.0     0.0     0.0     0.0     0.0     0.0     0.0
0.0     0.0     0.0     6.0     12.0    9.0     6.0     3.0     0.0     0.0     0.0     0.0
0.0     0.0     0.0     3.0     9.0     15.0    12.0    9.0     6.0     3.0     0.0     0.0
0.0     0.0     3.0     0.0     6.0     12.0    18.0    15.0    12.0    9.0     6.0     3.0
0.0     0.0     0.0     0.0     3.0     9.0     15.0    21.0    18.0    15.0    12.0    9.0
0.0     0.0     0.0     0.0     0.0     6.0     12.0    18.0    24.0    21.0    18.0    15.0
0.0     0.0     0.0     0.0     0.0     3.0     9.0     15.0    21.0    27.0    24.0    21.0
0.0     0.0     0.0     0.0     0.0     0.0     6.0     12.0    18.0    24.0    30.0    27.0
0.0     0.0     0.0     0.0     0.0     0.0     3.0     9.0     15.0    21.0    27.0    33.0
Common line:   ben geldim.
```