Iran University of Science and Technology, Computer Department

NLP Homework 2, Part 2

Vahid Kharazi

November 27, 2015

1 Local Alignment

According to wikipedia and course slides, we have to implement Smith-Waterman version of edit distance to solve local aligment problem.

$$H(i,0) = 0,\ 0 \le i \le m$$

$$H(0,j) = 0,\ 0 \le j \le n$$

$$H(i,j) = \max \begin{Bmatrix} 0 & \\ H(i-1,j-1) + s(a_i,b_j) & \text{Match/Mismatch} \\ \max_{k \ge 1}\{H(i-k,j) + W_k\} & \text{Deletion} \\ \max_{l \ge 1}\{H(i,j-l) + W_l\} & \text{Insertion} \end{Bmatrix},\ 1 \le i \le m, 1 \le j \le n$$

Where:

- a, b = Strings over the Alphabet Σ
- $m = \text{length}(a)$
- $n = \text{length}(b)$
- $s(a,b)$ is a similarity function on the alphabet
- $H(i,j)$ – is the maximum Similarity-Score between a suffix of a[1...i] and a suffix of b[1...j]
- W_i is the gap-scoring scheme

Test: There is an example in wikipedia to compute edit distance between ACACACTA and AGCACACA:

- $s(a, b) = +2$ if $a = b$ (match), -1 if $a \neq b$ (mismatch)
- $W_i = -1$

$$H = \begin{pmatrix} & - & A & C & A & C & A & C & T & A \\ - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A & 0 & 2 & 1 & 2 & 1 & 2 & 1 & 1 & 2 \\ G & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ C & 0 & 1 & 3 & 2 & 3 & 2 & 3 & 2 & 2 \\ A & 0 & 2 & 2 & 5 & 4 & 5 & 4 & 4 & 4 \\ C & 0 & 1 & 4 & 4 & 7 & 6 & 7 & 6 & 6 \\ A & 0 & 2 & 3 & 6 & 6 & 9 & 8 & 8 & 8 \\ C & 0 & 1 & 4 & 5 & 8 & 8 & 11 & 10 & 10 \\ A & 0 & 2 & 3 & 6 & 7 & 10 & 10 & 10 & 12 \end{pmatrix} \quad T = \begin{pmatrix} & - & A & C & A & C & A & C & T & A \\ - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A & 0 & \nwarrow & \leftarrow & \nwarrow & \leftarrow & \nwarrow & \leftarrow & \leftarrow & \nwarrow \\ G & 0 & \uparrow & \nwarrow & \uparrow & \nwarrow & \uparrow & \nwarrow & \nwarrow & \uparrow \\ C & 0 & \uparrow & \nwarrow & \leftarrow & \nwarrow & \leftarrow & \nwarrow & \leftarrow & \leftarrow \\ A & 0 & \nwarrow & \uparrow & \nwarrow & \leftarrow & \nwarrow & \leftarrow & \leftarrow & \nwarrow \\ C & 0 & \uparrow & \nwarrow & \uparrow & \nwarrow & \leftarrow & \nwarrow & \leftarrow & \leftarrow \\ A & 0 & \nwarrow & \uparrow & \nwarrow & \uparrow & \nwarrow & \leftarrow & \leftarrow & \nwarrow \\ C & 0 & \uparrow & \nwarrow & \uparrow & \nwarrow & \uparrow & \nwarrow & \leftarrow & \leftarrow \\ A & 0 & \nwarrow & \uparrow & \nwarrow & \uparrow & \nwarrow & \uparrow & \nwarrow & \nwarrow \end{pmatrix}$$

Similar to wikipedia example, this is my result for same:

```
(venv)vahid@kharazi:~/dev/nlp/two$ python two.py
   0    0    0    0    0    0    0    0    0
   0    2    1    2    1    2    1    1    2
   0    1    1    1    1    1    1    0    1
   0    1    3    2    3    2    3    2    2
   0    2    2    5    4    5    4    4    4
   0    1    4    4    7    6    7    6    6
   0    2    3    6    6    9    8    8    8
   0    1    4    5    8    8   11   10   10
   0    2    3    6    7   10   10   10   12
o    o    o    o    o    o    o    o    o
o    D    L    D    L    D    L    L    D
o    T    D    T    D    T    D   DTL   T
o    T    D    L    D    L    D    L    L
o    D    T    D    L    D    L    L   DL
o    T    D    T    D    L    D    L    L
o    D    T    D    T    D    L    L   DL
o    T    D    T    D    T    D    L    L
o    D    T    D    T    D    T    D    D
break
o    o    o    o    o    o    o    o    o
o    ,    L    D    L    D    L    L    D
o    ,    D    T    D    T    D   DTL   T
o    T    ,    L    D    L    D    L    L
o    D    T    ,    L    D    L    L   DL
o    T    D    T    ,    L    D    L    L
o    D    T    D    T    ,    L    L   DL
o    T    D    T    D    T    ,    ,    L
o    D    T    D    T    D    T    D    ,
['A', 'C', 'A', 'C', 'A', 'C', 'T', 'A']
['-', 'i', '-', '-', '-', '-', '-', 'd', '-']
['A', 'G', 'C', 'A', 'C', 'A', 'C', 'A']
```

The first matrix is H, Second matrix is T(T = Top, L = Left, D = Diag). Finally you can find the path of minimum edit distance:

```
['A', 'C', 'A', 'C', 'A', 'C', 'T', 'A']
['-', 'i', '-', '-', '-', '-', '-', 'd', '-']
['A', 'G', 'C', 'A', 'C', 'A', 'C', 'A']
```