

Methods in Computational Linguistics 2  
Spring 2015

Homework #4 - Dynamic Programming  
Due at 11:59pm on April 29

Use the minimum edit distance code posted on the course webpage as a starting point for this work.

1. Analyze the MTrace table to reconstruct the operations that are necessary to convert one string to the other. In the example we did in class, converting LEAD to LAST, this sequence would be:

```
Equal, L, L  
Delete, E  
Equal, A, A  
Substitute, D, S  
Insert, T
```

Note that this sequence is constructed in reverse order by starting in the lower right corner of the matrix  $M[k][l]$ , and moving back up to the top left corner  $M[0][0]$ . While the sequence is constructed in reverse order, your modified function should return the function in order.

2. In the implementation we used in class, insertions, deletions and substitutions all had the same cost, 1. Modify the implementation such that these three costs can be changed by changing the values of three variables: `INS_COST`, `DEL_COST`, and `SUB_COST`.
3. In the implementation we used in class, equality resulted in a zero cost “substitution”, while all other substitutions had the same cost. Modify the implementation such that all vowel-to-vowel (not ‘y’) substitutions have zero cost, all consonant-to-consonant substitutions have a cost of 1, and all vowel-to-consonant substitutions have cost of 1.5. This kind of operation is used to align unequal strings to each other. For example, phone sequences representing two different pronunciations of the same word, alternate orthographic representations of foreign names, and detecting spelling mistakes.