LING 227: Homework 2

Paul Fletcher-Hill

February 15, 2015

Part 1

Following are the results of my min_edit program for the sample pairs:

Minimum edit distance from drive to brief: 4
Minimum edit distance from animal to mammal: 4
Minimum edit distance from elementary to education: 15
Minimum edit distance from python to perl: 8

Part 2

The output of running score.py on my spelling_correction.py output is below:

Accuracy is 0.66742081448

It is worth noting that I defaulted to the last word with the minimum edit distance.

Part 3

I would expect the accuracy to increase, because it is more likely the correct word (or form of the word) is included in the lexicon. To test this hypothesis, I created a new dictionary that included every other word in the original dictionary. Therefore, the new dictionary had 1570 words. I then ran my spelling_correction.py program using the smaller dictionary. Sure enough, the accuracy of the output decreased significantly from the 3139 word dictionary:

Accuracy is 0.346153846154

This result would suggest that the size of the dictionary and the accuracy of the program are correlated, which is consistent with my hypothesis.

Part 4

My first two edits have to do with the substitution cost. I hypothesized that having the same number of letters but a mistyped letter is more common. Therefore, the cost associated with substituting two letters should be less than inserting and deleting. I initially simply set the default substitution cost to 1.5 rather

than 2.

But I could do better with the substitution cost, tying the cost more closely to the specific letters being substituted. I added a dictionary to my program that included all of the QWERTY keyboard keys as keys and arrays of adjacent keys as values. I then adjusted the substitution cost method to cost less if the substitution involved adjacent letters than if it did not.

The next edit I made used the same QWERTY dictionary but affected the deletion cost. Generally, the cost to delete a letter was constant at 1. Because typists might hit multiple keys at once, I checked whether the letter to be deleted and the last letter in the source were adjacent. If they were, the cost of deleting was less than if they were not.

Unfortunately, the two adjustments having to do with the QWERTY keyboard actually detracted from the accuracy of my spelling_correction program.

I then augmented the Levenshtein distance by decreasing the "length" between strings that had flipped letters—letters that were all the same just incorrectly ordered. This method is called the DamerauLevenshtein distance, and it improved my program's accuracy.

I also looked at the types of characters being substituted. I hypothesized that vowels being exchanged for vowels should be considered less of a change than a regular substitution. So upon each change, I checked if the character type was a vowel and adjusted the cost accordingly.

My final accuracy was 0.785067873303.