Russell Ronalds Comp. Ling.

2/20/19 HW 2

Questions, Part 2

1. *s* is a constant that helps place more similarity emphasis on less costly min-edit neighbors. The value that works best for *s* is .6: it essentially makes a cost-of-1 neighbor worth ~5 cost-of-2 neighbors, which are worth ~5 cost-of-3 neighbors. Which, of course, means that it takes more than 625 cost-of-4 neighbors to be worth 1 cost-of-1 neighbor.
2. The best correlation I can get with the default get\_neighbors N value left at -1: .667
3. Before finding a better fit for *s*, two particularly divergent examples are ‘drit’ and ‘trisk’. Once a good *s* is found, the amplitude of the divergence seems to decrease, but two examples that have rather low well-formedness ratings while having middling similarity calculations are ‘chool/Jul’ (4th worst WF rating) and ‘drice/dr2s’ (tied for 6th worst). Presumably these two wugs receive okay similarity calculations because they have some very uncostly edit neighbors. ‘chool’ has 1 cost-of-1, 7 cost-of-2 and 18 cost-of-3 neighbors; ‘drice’ has 2 cost-of-1, 6 cost-of-2, and 23 cost-of-3 neighbors. Other wugs that have similar WF ratings (the 5th and other 6th worst) have only 1 cost-of-1 and 6 cost-of-2 neighbors *between the both of them*. (As a comparison, ‘fro/fr5’ is both the best rated and has the highest similarity, and it has 3 cost-of-1, 18 cost-of-2, and 44 cost-of-3 neighbors.) So, then we have to ask why these verbs were rated so poorly. For ‘chool’, I would guess that verbs starting with the ‘J’ sound are rare—it seems there are only 45 present in the training list, so maybe that is relatively awkward for speakers to make new verbs with it, compared to the 203 that start with ‘fro’s’ first phoneme ‘f’.

Questions, Part 3

1. The accuracy of the model is ~71%.
2. The words that the model gets incorrect are:

Wug: Preferred Past: Model Prediction:

* 1. drice: NULL->t NULL->d
  2. rife: NULL->t NULL->d
  3. blafe: NULL->t NULL->d
  4. tesh: NULL->t NULL->d
  5. drit: NULL NULL->Id
  6. glit: NULL NULL->Id
  7. nold: NULL NULL->Id
  8. gude: NULL NULL->d
  9. fleep: iX->EXt NULL->t
  10. gleed: i->E NULL->Id
  11. queed: i->E NULL->Id
  12. skride: 2->5 NULL->d

For the first category of mistake, a prediction of NULL->d while a preference for NULL->t, seems to be based mostly on predicting a similarity to other verbs that have low min-edit costs, but that have final sounds that are voiced, while the four on the above list end in voiceless phonemes, which should take a ‘t’ for past tense. (I can’t even really say the model-predicted ‘tESd’) Close neighbors that are leading the model of NULL->d astray seem to be:

drice: (I presume the model’s mistake comes from a cocktail of irregulars, ‘d’, and ‘t’ endings: cost-of-1 (dried, diced), cost-of-2 (dressed, died, dyed, priced, drove, iced);

rife: rose, dried, fried, righted, pried, riled;

blafe: blared, blamed, blazed, belayed;

tesh: cost-of-2 (told, meshed [the irregular here helps invalidate the power of meshed]), cost-of-3 (towed, ate, tested, tied, treaded)

For the next category, predictions of NULL->Id with a preference for NULL, phonetically the predictions are grammatical, but for the first two they are probably affected by some cost-of-1 neighbors they have, but who don’t end up having the same ending as they do. (Actually, after doing part 4, I realized it’s because of the dearth of examples in the training data for irregular NULL changes. ‘drit’ doesn’t have a single example within 3 neighbors of a NULL change) :

drit: drifted

glit: glitter, glinted

‘nold’ is a bit of a mystery to me, as there is no verb on the training list that ends with ‘5ld’ and is in the NULL past tense category (They are all either 5ld->5ldId or 5ld->Eld). You would think ‘mold/mould’ or ‘fold’ would lead a listener to ‘nolded’. (Means that I don’t have a good idea of how to get the accuracy to 100%, certainly.)

For the last five verbs that the model got wrong, they are complicated mostly by too many close min-edit neighbors that actually have different features that are not taken into account, for example:

Fleep/’flip’ should turned into ‘flEpt’, but there is a cost-of-1 neighbor ‘flee’ that turns to ‘flEd’ and 3 cost-of-3s flip/flap/flop that overshadow the 4th one: sleep.

Questions, Part 4

First change: If a target word and the source word are doing a 0-cost substitution, and that substitution is happening on the last letter of the word, I am going to heavily favor this: the cost is now -2, which increases the accuracy to 83%. Just changing it to -1 took 4 incorrect predictions off the list, but left blafe, so I increased the favor. I was expecting this to help with glit and drit, but it didn’t! (But, after looking at them more closely, I think it is because they are looking to be NULL irregulars, and there are not that many of those to build a model off of. I got glit to be predicted, but I don’t see a good way to do it for drit.)

I also changed the substitutions to favor vowel to vowel shifts, but I actually saw no resulting change whatsoever…

Okay, so I was able to get my predictions up to 90%, but my correlation fell to .51! My main changes to min\_edit\_part4.py are in the screenshot below:

