Developing A Real-Word Spelling Corrector
Based on Dickinson, Brew, & Meurers (2013)
Spring 2017

Warning: this activity may cause you to think numerically.

1. We've talked about n-grams for language processing techniques, and I now want you to think about how you would use trigrams in order to develop a real-word spelling corrector. You're in charge here: it's up to you to figure out what to tell some programmers what to program. How are you going to build a system?

   Some issues to think about include:

   (a) **Data:** Where will you get your trigram probabilities? That is, what type of data are you interested in correcting and thus what type of data do you need to collect? How much do you need?
   - Also: how will you handle issues of *data sparsity*, cases where a word or a sequence of words has never been seen before?

   (b) **Candidate generation:** How will candidate correction *sentences* be generated?
   - Thinking about individual word candidate generation, how many possible candidates do you expect for any individual word?
   - Based on your previous answer: will you allow every word to be changed? How many changes per sentence will you allow? (Think about efficiency.)
   - Do you want to use pre-defined *confusion sets*, sets of commonly confused words (e.g., {*their*, *there*, *they're*})? How many sets would you need? What are some other examples?

   (c) **Candidate ranking:** What will the probability model look like? That is, which probabilities will you compare in order to compare *sentences*?
   - Which trigrams will you use to calculate your probabilities? i.e., which words will you use in your trigrams? For example, if you're trying to correct the word *crowd* to *crows* in *those nasty crowd in Bloomington*, do you look at $p(crows|those, nasty, \_)$, $p(crows|nasty, \_, in)$, $p(crows|\_, in, Bloomington)$? All of these? Some other probability? (Be sure to consider other examples, too, e.g., *that algorithm peaked my interest* needing to be corrected to *that algorithm piqued my interest*.)

   Sketch out a design in very broad terms.[1]

2. Now, let's not use trigrams, but instead base our system on these confusion sets. What other kinds of information would help us disambiguate such **content-based** confusions sets like {*weather*, *whether*}; {*principal*, *principle*}; etc.?[2]
   - What types of spelling errors will this catch that the other methods won't, do you think?

References

Golding, Andrew R. and Dan Roth (1999). A Winnow-Based Approach to Context-Sensitive Spelling Correction. *Machine Learning* 34(1-3), 107–130.

Hirst, Graeme and Alexander Budanitsky (2005). Correcting real-word spelling errors by restoring lexical cohesion. *Natural Language Engineering* 11(1), 87–111.

Mays, Eric, Fred J. Damerau and Robert L. Mercer (1991). Context based spelling correction. *Information Processing and Management* 23(5), 517–522.

Wilcox-O'Hearn, L. Amber, Graeme Hirst and Alexander Budanitsky (2006). Real-word spelling correction with trigrams: A reconsideration of th e Mays, Damerau, and Mercer model. `http://www.cs.toronto.edu/compling/Publications/Abstracts/Papers/WilcoxOHearn-etal-2006-abs.html`.

[1] For more on a trigram model, see: Mays et al. (1991); Wilcox-O'Hearn et al. (2006)
[2] See, e.g., Golding and Roth (1999); Hirst and Budanitsky (2005)