

Spelling Corrector

Eric Normand

Schedule

- Quick introduction
- Implementation in groups
- Presentation of results

Norvig Spelling Corrector

- Implemented on your handout
- Uses Bayes' Theorem
- Development + evaluation sets

Assignment

- Implement a function correct that returns the most likely spelling for a given word
- Details:
 - Should be fast (as you type)
 - Should be accurate

Plan

- Break into groups of 3-4
- Choose a language
- Get hacking
- Evaluate the speed and accuracy
- Present your work (last hour)

Presentation

- Implementation
- Speed & accuracy
- Advantages & disadvantages
 - type safety
 - parallelism
 - etc

A little bit more about the presentation. There will be a lot of questions here. Up to now should take no more than 10 minutes to present. I will go around and help groups form. They should get to work.

Multiple corrections

- **Modify correct to return a list of corrections, most likely first**

Add to dictionary

- Provide an interface for adding a new word to the training set.

This permutation is more about the real-world aspect of mutability. It is easy in functional programming languages to forget that interactions with the user develop a context--which may require mutable state. This is also an interesting challenge to the original probabilistic model of spelling correctness.

Mistyped space

- Solve the “mistyped space” problem.

In the original problem statement, space characters are considered absolutely correct. But this is often not the case, especially when typing on a touchscreen keyboard. It is common to accidentally hit space when trying to touch one of the letters on the bottom row. This will be difficult for some.

Touchscreen correction

- Touchscreens imperfectly detect which letter you intended to type. Modify the spelling corrector to account for this problem.