NormalizedTypoDifference

Algorithm

The algorithm returns:

- an exception if either argument is null (Score catches and handles this)
- 0.0 if both strings are of length 0
- a normalized float (between 0.0 and 1.0) be doing the following:
    1. determine the max length of the two strings
    2. evaluate the two strings using the Damerau-Levenshtein Edit Distance algorithm
    3. augment the distance by the result of getAugment()
    4. normalize the distance by evaluating ((maxLength - damlevDist)/maxLength)

Special Hook Methods

The Damerau-Levenshtein implementation has an additional hook method called distAugment() that takes information about what type of edit this is and the two characters.

The Typo Difference algorithm evaluates the two characters for proximity on a qwerty keyboard in the cases of a character swap (replace one character with a different one).

- If the keys for the corresponding characters are adjacent and would be typed with the same state of the shift key (ex: a shifted key swapped with an adjacent shifted key) the edit cost is reduced by .5 (normally a cost of 1 per edit, but that character swap is now only a cost of .5 due to the key proximity)
- If the keys for the corresponding characters are adjacent but would be typed with the different state of the shift key (ex: a shifted key swapped with an adjacent unshifted key) the edit cost is reduced by .3 (normally a cost of 1 per edit, but that character swap is now only a cost of .3 due to the key proximity)