HOMEWORK 3

(Participate in Kaggle competition AND upload your jupyter notebook or python script in NYU Classes)

Kaggle Competition Link: https://www.kaggle.com/t/224c05a5c9924eaaaa5c826e76e65c44

Goal :

Create a spell checker: correct spelling of a corrupted word

Steps:

1. Train a language model on training data.
2. Use your model to predict the outputs (correct spelling) for the test data (corrupted words).
3. Upload your prediction as a file to the in class Kaggle competition for evaluation and ranking (similar to homework 2).

Sample Model:

We see an observation x (a misspelled word) and our goal is to find the word w which was corrupted to generate the misspelled word, out of all possible words in a vocabulary V.

Model 1:

Generate all the words from misspelled word x with an edit distance of 0, 1 & 2 including insertions, deletions, and substitutions. Also, use transpositions for generating the words (called as Damerau-Levenshtein edit distance). After generating all the candidate words, select the word w which has the maximum number of occurrences (probability) in your original dataset, as the correct word.

Model 2:

Find w which maximizes p(w|x). (Find the maximum probability such that w is the corrected word given the misspelled word x). Using Bayes rule, p(a | b) = p(b | a) p(a) / p(b), compute the posterior p(w|x) = p(x|w) p(w). Compute p(w) from the training data. Compute p(w) using the data in file count_1w.txt. Compute p(x|w) using the data file spell-error.txt as follows:

Generate the following matrices:
An edi_distance_matrix lists the number of times one thing was confused with another. For example, a substitution matrix is a square matrix of size 26 X 26 (or more generally |A| X |A| , for an alphabet A) that represents the number of times one letter was incorrectly used instead of another.
del[x;y] : count(xy typed as x)
ins[x;y] : count(x typed as xy)
sub[x;y] : count(x typed as y)
trans[x;y] : count(xy typed as yx)

Once the edit_distance_matrices are4 computed, estimate P(x|w) as follows (where w_i is the ith character of the correct word w) and x_i is the ith character of the typo x :

$$P(x|w) = \begin{cases} \frac{\text{del}[x_{i-1}, w_i]}{\text{count}[x_{i-1} w_i]}, \text{ if deletion} \\ \frac{\text{ins}[x_{i-1}, w_i]}{\text{count}[w_{i-1}]}, \text{ if insertion} \\ \frac{\text{sub}[x_i, w_i]}{\text{count}[w_i]}, \text{ if substitution} \\ \frac{\text{trans}[w_i, w_{i+1}]}{\text{count}[w_i w_{i+1}]}, \text{ if transposition} \end{cases}$$

Then use p(w|x) = p(x|w) p(w) to find the word w such which maximizes p(w|x)
YOU CAN USE ANY MODELS OF YOUR INTEREST FOR THE KAGGLE COMPETITION OR USE THE ABOVE MODELS.

Data Description:

Training Datasets:

- ➔ big.txt: This file consists of lot of text. You can use this file to collect information about words. Number of occurrences of word.
- ➔ spell-errors.txt: The format of every line in this text is the correct word is specified followed by the misspelled words. For example, like "raining: rainning, raning". Form the 4 edit_distance_matrices from this data set.
- ➔ count_1w.txt & count_2w.txt: count_1w.txt has text in the format: <word1> <no. of occurrences> and count_2w.txt has text in the format: <word1> <word2> <no.of occurrences>. You can make use of these datasets to create much better models.

Test Datasets:

- ➔ test.csv consists of 504 incorrect/misspelled words.

Submission Format (in Kaggle competition):

For every student in the competition, submission files should contain two columns: 'ID & 'CORRECT'. ID will have values from 0 to 503 and 'CORRECT' column should have corrected words of 504 test samples of test.csv
Note: You can download 'test_submit.csv' to know how should your submission file be

Submission Format (in NYU Classes):

You must submit a jupyter notebook or a python script which has the model which was used to submit the file in Kaggle competition.
Team Size: At most 2 students in one team.
Complete Homework Grading Criteria:
1. Based on your rank on the leaderboard in Kaggle Competition
2. Your uploaded jupyter notebook file in NYU Classes (any one of the students in the team can submit the jupyter notebook in NYU Classes)