The Report of HW 1: Spelling Correction

Cheng Zhong 16307110259

Principle:

The processing of spelling correcting is actually the process of finding the words that are most likely to make the original sentence reasonable. And these probabilities can be obtained from the language model and the channel model. We assume that every word is related from its previous word, according to the Bayesian distribution, we can get the probably of the word in Bigram model.

In order to prevent the occurrence that we can't find the word in dictionary which will lead to the error of divided 0. We develop some smoothing methods for this case, for example, the add-k estimate、the Absolute Discounting Interpolation and Kneser-Ney smoothing. In this way, we can convert the probability (rationality) of the entire sentence into the joint probability distribution of each binary phrase, and finally select the most appropriate word by maximum likelihood estimation.

Data Processing:

Corpus: NLTK Reuters corpus *('Bigram.txt','Unigram.txt')*
Channel model: Peter Norvig 's list of counts of single-edit *('change.txt')*
Environment: Python 3.7， Windows 10
Modified "Salem,Ore" on line 540 as "Salem, Ore" (add a space)

Workflow:

As for spelling error, the processing flow of correcting is:

1) Traverse the entire dictionary to find all words which the edit distance is less than 3.
2) Calculate the Unigram probability for each word
3) Replace the original words with alternative words, and calculate the probability of whole sentence by Bigram model
4) Sum the probability above by weight, and choose the most likely one to substitute the error word.

Through the process above, we can quickly find the correct word.

Validations: (More details can be found in 'Vaildation.xlsx')

Interpolation:

First, I try to solve the problem by adjusting the weight of each language model.

$$\mathrm{P(w)} = \lambda_1 \mathrm{P(channel)} + \lambda_2 \mathrm{P(Unigram)} + \lambda_3 P(Bigram), \quad \sum_{i=1}^{3} \lambda_i = 1$$

From the result, we can get a conclusion that channel model and bigram are more effective in detecting misspellings. By adjusting the weight, we can increase the accuracy to 93.9%.

Smoothing:

When testing the effect of the k parameter in add-k smoothing (the formula is as follows), we found that modifying K has little effect on the result, but when we use the Absolute Discounting Interpolation (Kneser-Ney smoothing in Unigram case), The accuracy will be reduced to 83.6%

$$P_{Add-k}(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i) + k}{c(w_{i-1}) + kV} \quad P_{KN}(w_i \mid w_{i-1}) = \frac{\max(c(w_{i-1}, w_i) - d, 0)}{c(w_{i-1})} + \lambda(w_{i-1}) P_{CONTINUATION}(w_i)$$

Ameliorated orientation:

By observing errors that have not been corrected correctly, we find that it is mainly some misuse of the typos and fixed collocations of the context, as well as some grammatical structural errors. So, if we want to improve the accuracy, we should also consider the grammar and fixed collocation knowledge.