Spelling Correction Homework

Yanjian Zhang 16300200020

目录

1 Final result

My model reach 92% accuracy in the eval test for add-k method, and 93% accuracy for the kneser_key method, the result is concluded in result.txt

add-k	kneser_key
92%	93%

2 Environment

Python 3.6.3
numpy 1.13.1

3 Language model

3.1 ADD-k model

I use add-k thought in the first language model

$$P(w_i|w_{i-1}) = \frac{count(w_i|w_{i-1}) + \lambda}{count(w_{i-1}) + \lambda * count_{vocab}}$$

I also use the word after the wrong word to implement bigram . I calculate the result of the following and multiply them together

$$P(w_i|w_{i+1}) = \frac{count(w_i|w_{i+1}) + \lambda}{count(w_{i+1}) + \lambda * count_{vocab}}$$

$$P(w_i|w_{i-1}, w_{i+1}) = P(w_i|w_{i-1}) * P(w_i|w_{i+1})$$

3.2 Kneser-Ney model

I implement it in bigram way, it's formula as follow

$$p_{KN}\left(w_i|w_{i-1}\right) = \frac{\max\left(c\left(w_{i-1}, w_i\right) - \delta, 0\right)}{\sum_{w'} c\left(w_{i-1}, w'\right)} + \lambda_{w_{i-1}} p_{KN}\left(w_i\right)$$

$$p_{KN}\left(w_i\right) = \frac{\left|\{w' : 0 < c\left(w', w_i\right)\}\right|}{\left|\{(w', w'') : 0 < c\left(w', w''\right)\}\right|}$$

$$\lambda_{w_{i-1}} = \frac{\delta}{\sum_{w'} c\left(w_{i-1}, w'\right)} \left|\{w' : 0 < c\left(w_{i-1}, w'\right)\}\right|$$

I also use the word after the wrong word to implement bigram . I calculate the result of the following and multiply them together

$$p_{KN}\left(w_i|w_{i+1}\right) = \frac{\max\left(c\left(w_i, w_{i+1}\right) - \delta, 0\right)}{\sum_{w'} c\left(w', w_{i-1}\right)} + \lambda_{w_{i+1}} p_{KN}\left(w_i\right)$$

$$p_{KN} = p_{KN}\left(w_i|w_{i-1}\right) * p_{KN}\left(w_i|w_{i+1}\right)$$

4 Channel model

4.1 Confusion Matrix

4.1.1 detail

1. I used the count_1edit dataset provided by Peter Norvig to compute my confusion matrix [http://norvig.com/ngrams/count_1edit.txt]

2. I use trie to find the word within edit_distance, it turn out work efficiency

4.2 Channel model

4.2.1 channel prob

$$P_{del}(X|w) = \frac{del(w_{i-1}, w_i)}{count(w_{i-1}, w_i)}$$

$$P_{ins}(X|w) = \frac{ins(w_{i-1}, x_i)}{count(w_{i-1})}$$

$$P_{sub}(X|w) = \frac{sub(w_{i-1}, x_i)}{count(w_{i-1})}$$

$$P_{trans}(X|w) = \frac{trans(w_{i-1}, w_i)}{count(w_{i-1}, w_i)}$$

5 Final Model

5.1 detail

1. The combination of the language model and the channel model can be concluded into the following formulas:

$$P = P_{language}(x|w_{i-1}, w_{i+1}) * P_{channel}(x|w)$$

2. I treat the edit-distance of transpose as 1 instead of 2, which is defined in Damerau-Levenshtein distance instead of Levenshtein distance. Making transpose 1 edit-distance broaden the choice of 1 edit-distance, which is reasonalbe for transposing quite frequently appearing.

3. Further thought for the implementation: my result of kneser_key method is quite close to the add-k method，which mean I maybe didn't make the best use of its advantage.It may get better result in more gram stage.

6 code explanation

6.1 kgram.py

initCount() is implemented at first to tokenize the reutersCorpus.txt for counting the number of grams.

initCount2() is implemented when I use the reuters in nltk, it turn out to be a better result.

Count.pk is the result delivering file from kgram to spellCorrction.py, cutting the time for future initial

6.2 confusionMatrix.py

initMatrix() is implemented at first to make the matrix of spell-error for channel model, but since I use nearly all of the letter and signal, the matrix is really sparse.

initMatrix() is implemented when I use the count_1edit.txt to make the matrix of channel. The result turn out better.

Matrix.pk is the result delivering file from confusionMatrix to spellCorrection.py, cutting the time for future initial

6.3 spellCorrection.py

def Pxw(): it is used for calculating the channel result

def kneser_key(): it is used for calculating the language model result in kneser_key way

def add_k(): it is used for calculating the the language model result in add_k way

def P(): it is used for calculating the unigram frequecy of word

def get_candidate(): it is used for searching the result within the edit_distance