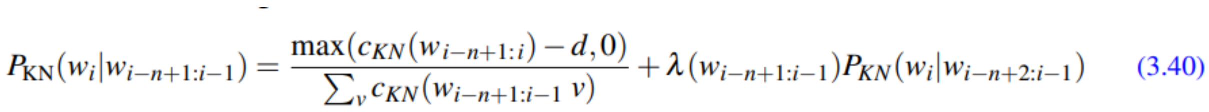
Report

2020101074

Naimeesh Narayan Tiwari

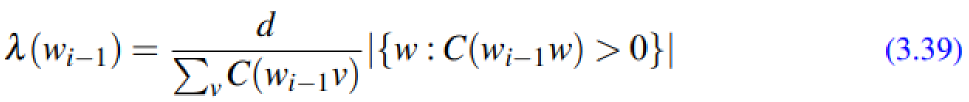
Implementation of Smoothing

1. Kneser Ney (Interpolated)



$$P_{\text{KN}}(w_i|w_{i-n+1:i-1}) = \frac{\max(c_{KN}(w_{i-n+1:i}) - d, 0)}{\sum_v c_{KN}(w_{i-n+1:i-1}\ v)} + \lambda(w_{i-n+1:i-1})P_{KN}(w_i|w_{i-n+2:i-1}) \qquad (3.40)$$

used this formulation where,



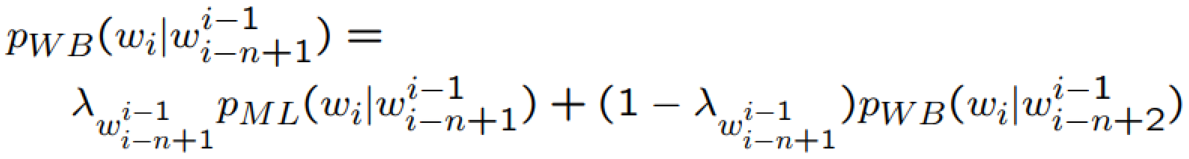
$$\lambda(w_{i-1}) = \frac{d}{\sum_v C(w_{i-1}v)}|\{w : C(w_{i-1}w) > 0\}| \qquad (3.39)$$

and d = discounting factor

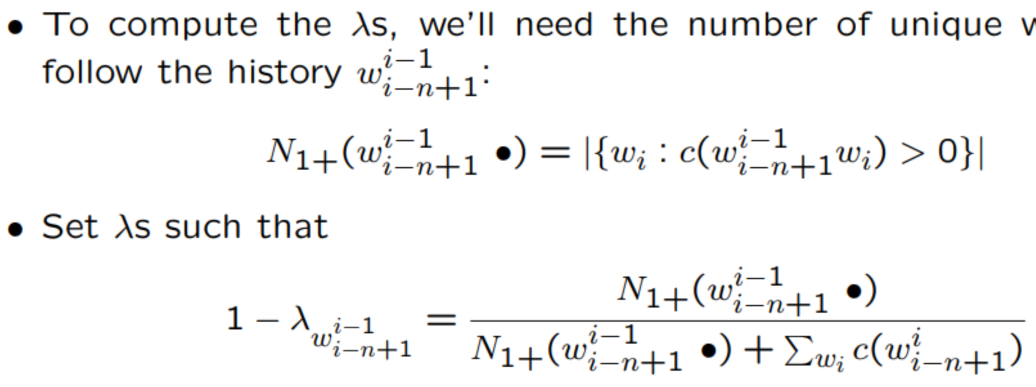
- using analysis, for n=1, d=0.5 and for n>1, d = 0.75 is taken

[screenshots from Jurafsky and Martin, 3rd Ed]

2. Witten Bell



$$p_{WB}(w_i|w_{i-n+1}^{i-1}) = \lambda_{w_{i-n+1}^{i-1}} p_{ML}(w_i|w_{i-n+1}^{i-1}) + (1 - \lambda_{w_{i-n+1}^{i-1}}) p_{WB}(w_i|w_{i-n+2}^{i-1})$$



- To compute the λs, we'll need the number of unique words that follow the history w_{i-n+1}^{i-1}:

$$N_{1+}(w_{i-n+1}^{i-1}\ \bullet) = |\{w_i : c(w_{i-n+1}^{i-1} w_i) > 0\}|$$

- Set λs such that

$$1 - \lambda_{w_{i-n+1}^{i-1}} = \frac{N_{1+}(w_{i-n+1}^{i-1}\ \bullet)}{N_{1+}(w_{i-n+1}^{i-1}\ \bullet) + \sum_{w_i} c(w_{i-n+1}^{i})}$$

[screenshot from Bill MacCartney , NLP Lunch Tutorial]

Comparisons

1. Witten bell runs faster than kneser ney mainly due to P_continuation calculation in kneser ney which makes it slower.
2. storing values helps in optimising the implementation
3. Here avg perplexities reported in training set are nearly the same for both witten bell and kneser ney
4. However kneser ney reports greater perplexity than witten bell for test set
5. This might be due to following reasons:
    a. **Tokenization issues** related to the corpus - cleaning more suitably might help in more accurate and comparable results since If the text is not properly tokenized and contains too many or too few tokens, it can lead to incorrect estimates of N-gram probabilities, which can negatively impact performance. In general, the Witten-Bell method is considered to be more robust to bad tokenization than the Kneser-Ney method. The reason for this is that the Witten-Bell method estimates the probability of unseen n-grams using a more complex discounting method that takes into account the frequency of the n-gram and the frequency of the context in which it appears. This allows the Witten-Bell method

to better handle unseen n-grams and OOV words that may arise from bad tokenization. In contrast, the Kneser-Ney method relies on a simpler discounting approach that estimates the probability of an unseen n-gram based on the frequency of its prefix in the training data. This method can be less effective at handling bad tokenization because it does not take into account the frequency of the context in which the n-gram appears.

b. **Handling Unknowns:** When we increase the threshold for converting the rare tokens into <UNK> tokens, we see a drop in perplexities. The Kneser-Ney smoothing method may benefit more from this approach than the Witten-Bell method. So keeping the threshold impractically lower might hurt Kneser Ney more than Witten-Bell.

The following link contains files with score after increasing threshold to 10

https://drive.google.com/drive/folders/13NXf5Q7d9JKVkkG5mr8QYSsWQaNJvytK?usp=sharing