1 Advanced Language Modelling

Exercise 1.1: Kneser-Ney Language Models

a) We know that,

$$\sum_{v \epsilon V} P_{KN}(w3 = v \mid w1w2) = 1 \text{ and}$$

$$\sum_{v \epsilon V} P_{KN}(w3 = v \mid w2) = 1, \text{ where } N(w1w2) > 0$$

$$P_{KN}\ (w3 \mid w1w2) = \frac{\max\{N(w1w2w3) - d, 0\}}{N(w1w2)} + \lambda(\mathrm{w1w2})\ P_{KN}\ (w3 \mid w2) \text{ [From eqn. (2)]}$$

Plugging the value of $\sum_{v \epsilon V} P_{KN}(w3 = v \mid w1w2)\ in\ eqn.(2)$:

$$1 = \frac{\max\{N(w1w2w3) - d, 0\}}{N(w1w2)} + \lambda(\mathrm{w1w2})\ P_{KN}\ (w3 \mid w2)$$

$$1 - \frac{\max\{N(w1w2w3) - d, 0\}}{N(w1w2)} = \lambda(\mathrm{w1w2})\ P_{KN}\ (w3 \mid w2)$$

So the back-off factor, $\lambda(\mathrm{w1w2}) = \frac{\left(1 - \frac{\max\{N(w1w2w3) - d, 0\}}{N(w1w2)}\right)}{P_{KN}\ (w3 \mid w2)}$

$$P_{KN}\ (w3 \mid w2) = \frac{\max\{N_{1+}(\bullet\ w2w3) - d, 0\}}{N_{1+}(\bullet\ w2\ \bullet)} + \lambda(\mathrm{w2})\ P_{KN}\ (w3) \text{ [From eqn. (4)]}$$

Plugging the value of $\sum_{v \epsilon V} P_{KN}(w3 = v \mid w2)$ in eqn. (4):

$$1 = \frac{\max\{N_{1+}(\bullet\ w2w3) - d, 0\}}{N_{1+}(\bullet\ w2\ \bullet)} + \lambda(\mathrm{w2})\ P_{KN}\ (w3)$$

$$1 - \frac{\max\{N_{1+}(\bullet\ w2w3) - d, 0\}}{N_{1+}(\bullet\ w2\ \bullet)} = \lambda(\mathrm{w2})\ P_{KN}\ (w3)$$

So the back-off factor, $\lambda(\mathrm{w2}) = \frac{\left(1 - \frac{\max\{N_{1+}(\bullet\ w2w3) - d, 0\}}{N_{1+}(\bullet\ w2\ \bullet)}\right)}{P_{KN}\ (w3)}$

b)

	w = ‘york’	w = ‘matter’
N(w)	2374	2367
$N_{1+}(\bullet\ w)$	9	270
$-\ log_2\ P_{ML}\ (w)$	12.258374283571046	12.262634512478337
$-\ log_2\ P_{Lids}\ (w)$	12.268770360977008	12.273028793453731
$-\ log_2\ P_{KN}\ (w)$	17.964084485236466	13.057193889627946

Table 1

From the table above we can see that the word ‘york’ and the word ‘matter’ almost have the same frequency. However, by observing their bigrams we can observe that ‘york’ comes after only 9 words whereas, ‘matter’ comes after 270 different words. So, in a unknown context the probability of the next word being ‘matter’ should be more than the ‘york’ which comes after some certain words. The maximum likelihood estimation and the lidstone smoothing technique give almost same probability which is not accurate in such kind of scenarios. However, Kneser-Ney smoothing can do the job way better by checking the history (bigram) and gives a better prediction of the next word in an unknown context. Since, ‘matter’ is preceded by more different words, it should have more probability than the word ‘york’ in an unknown context and Kneser-Ney smoothing is doing exactly the same by calculating ‘york’ as 17.964084485236466 and ‘matter’ 13.057193889627946. As it is expressed as – log 2 base the lower the value the better the probability of getting selected as a next word in an unknown context.

c) Among the various smoothing techniques Kneser-Ney smoothing is the most precise and gives better estimation. It is usually applied in interpolated form and is developed on the idea of absolute discounting interpolation that makes the use of both higher and lower order LM. The process is done by moving probability mass from higher to lower order language model.

The Kneser-Ney model uses technique of absolute discount smoothing interpolation as a backoff model. The Kneser-Ney design retains the first term of absolute discounting interpolation, but rewrites the second term. Whereas absolute discounting interpolation in a bigram model would simply default to a unigram model in the second term, Kneser-Ney depends upon the idea of a continuation probability associated with each unigram.

If we recall the maximum likelihood estimation of unigram language model we get the following:

$$P_{ML}\text{ (w) } = \frac{c(w)}{\sum_i c(w_i)}$$

However, from the equations of a) we can clearly see that Kneser-Ney smoothing replace raw counts with the count of histories. Instead of determining how likely a word w_i is to appear in unigram model, the Kneser-Ney smoothing evaluates how likely a word w_i is to appear in an unfamiliar bigram context.