Introduction to NLP (CS7.401)

Spring 2022, IIIT Hyderabad
21 Jan, Friday (Lecture 4)

Taught by Prof. Manish Shrivastava

Smoothing (contd.)

Interpolation and Backoff

Discounting algorithms help solve the problem of zero-frequency N-grams, but they do not use the knowledge we have of smaller N-grams. For example, if we are working with a zero-frequency 4-gram, we could consider the frequency of the 3-grams and 2-grams that make it up.

Interpolation mixes estimates from all smaller N-grams. For example, we have linear interpolation, in which we use the formula

$$\begin{aligned} P(w_n \mid w_{n-1}w_{n-2}) = {} & \lambda_1 P(w_n \mid w_{n-1}w_{n-2}) \\ & + \lambda_2 P(w_n \mid w_{n-1}) \\ & + \lambda_3 P(w_n), \end{aligned}$$

where

$$\sum_i \lambda_i = 1.$$

A slightly more sophisticated model would use context-dependent weights:

$$\begin{aligned} P(w_n \mid w_{n-1}w_{n-2}) = {} & \lambda_1(w_{n-2}^{n-1}) P(w_n \mid w_{n-1}w_{n-2}) \\ & + \lambda_2(w_{n-2}^{n-1}) P(w_n \mid w_{n-1}) \\ & + \lambda_3(w_{n-2}^{n-1}) P(w_n), \end{aligned}$$

where

$$\sum_i \lambda_i(w_{n-2}^{n-1}) = 1.$$

To compute the λ_i, we use the held-out corpus (an additional training corpus used to set parameters of the model like these).

In backoff, we check each smaller N-gram at a time: first we check the 3-gram, and if it has zero frequency, then we check the 2-gram, and then the 1-gram. It

can work better than interpolation, which is relatively simple.
One model is called Katz backoff, which calculates the probabilities as:

$$P_{\text{katz}}(w_n \mid w_{n-N+1}^{n-1}) = \begin{cases} P^*(w_n \mid w_{n-N+1}^{n-1}) & C(w_{n-N+1}^{n}) > 0 \\ \alpha(w_{n-N+1}^{n-1}) P_{\text{katz}}(w_n \mid w_{n-N+2}^{n-1}) & \text{otherwise.} \end{cases}$$

Here the discounted probability P^* is defined as

$$P^*(w_n \mid w_{n-N+1}^{n-1}) = \frac{c^*(w_{n-N+1}^{n})}{c(w_{n-N+1}^{n-1})}$$

and the function α as

$$\begin{aligned} \alpha(w_{n-N+1}^{n-1}) &= \frac{\beta(w_{n-N+1}^{n-1})}{\sum_{w_n : c(w_{n-N+2}^{n-1})>0} P_{\text{katz}}(w_n \mid w_{n-N+2}^{n-1})} \\ &= \frac{1 - \sum_{w_n : c(w_{n-N+2}^{n-1})>0} P^*(w_n \mid w_{n-N+1}^{n-1})}{1 - \sum_{w_n : c(w_{n-N+2}^{n-1})>0} P^*(w_n \mid w_{n-N+2}^{n-1})}. \end{aligned}$$

If $x(w_{n-N+1}^{n-1}) = 0$, then

$$\begin{aligned} P_{\text{katz}}(w_n \mid w_{n-N+1}^{n-1}) &= P_{\text{katz}}(w_n \mid w_{n-N+2}^{n-1}) \\ P^*(w_n \mid w_{n-N+1}^{n-1}) &= 0 \\ \beta(w_n \mid w_{n-N+1}^{n-1}) &= 1. \end{aligned}$$

Discounting tells us how much probability mass to set aside from nonzero-frequency counts, and backoff allows us to distribute it in a more informed manner.

Kneser-Ney Smoothing

Kneser-Ney smoothing is an algorithm that counts the *number of histories* a word has occurred with, and uses it to estimate the probability of the current context. Kneser-Ney backoff is formalised as follows (assuming α is defined so as to make everything sum to 1):

$$P_{\text{continuation}}(w_i) = \frac{|\{w_{i-1} : c(w_{i-1}w_i) > 0\}|}{\sum_{w_i} |\{w_{i-1} : c(w_{i-1}w_i) > 0\}|}$$

$$P_{\text{KN}}(w_i \mid w_{i-1}) = \begin{cases} \frac{c(w_{i-1}w_i) - D}{c(w_{i-1})} & c(w_{i-1}w_i) > 0 \\ \alpha(w_i) P_{\text{continuation}}(w_i) & \text{otherwise.} \end{cases}$$

Kneser-Ney interpolation, however, has been shown to be superior to the backoff algorithm. It is calculated as

$$P_{\text{KN}}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}w_i) - D}{c(w_{i-1})} + \beta(w_i) P_{\text{continuation}}(w_i).$$