

Handling Unseen Events

Remember that there are several methods for handling unseen events in n -grams:

1. UNK mapping: map every type that occurs $< k$ times to the distinguished type “UNK” before training and testing. This is a simple approximation of a class-based model.
2. “backoff:” if not enough evidence for n -gram to opine, ask k -gram, for $k < n$. This is a simple approximation of an interpolated model.
3. Witten-Bell smoothing: based upon the probability of seeing an event for the first time.
4. Good-Turing smoothing: based upon frequencies of frequencies.

All of them are attempts at a principled justification for how much probability mass they steal from seen events to give to unseen events.

Class-based Models

Base the probability of an unseen event on the class to which it belongs, e.g., part-of-speech:

$$P(w_n | w_1 \dots w_{n-1}) \equiv \sum_{c_1 \dots c_n} (P(c_n | c_1 \dots c_{n-1}) \cdot P(w_n | c_n) \cdot \prod_{i=1}^{n-1} P(c_i | w_i))$$

Often this is approximated as:

$$P(w_n | w_1 \dots w_{n-1}) \approx \max_{c_n} P(c_n | c_1 \dots c_{n-1}) P(w_n | c_n)$$

where the class labels, c_1, \dots, c_{n-1} are assumed to have been assigned with certainty.

Deleted Interpolation

Let:

$N_r \equiv$ the number of n -grams with frequency r in training data

$T_r \equiv$ the number of tokens in held-out data of n -grams appearing r times in training data, i.e.:

$$T_r = \sum_{C_1(w_1 \dots w_n)=r} C_2(w_1 \dots w_n)$$

Deleted interpolation uses training data to gather frequency counts, but held-out data to smoothe the probability estimates.

The average frequency (in held-out data) of n -grams occurring r times (in training data) is: $\frac{T_r}{N_r}$.

So we let $P(w_1 \dots w_n) = \frac{T_r}{N_r} \cdot \frac{1}{T}$,

where $r = C_1(w_1 \dots w_n)$ and T is the total number of tokens of held-out data (*held-out estimator*).

Deleted Interpolation

But since we're using held-out data, we can also cross-validate: use both training and held-out data

both ways: $\frac{T_{2,r}}{N_{1,r}} \cdot \frac{1}{T_2}$, $\frac{T_{1,r}}{N_{2,r}} \cdot \frac{1}{T_1}$

and then take a weighted average:

$$P(w_1 \dots w_n) = \frac{T_{1,r} + T_{2,r}}{(T_1 + T_2)(N_{1,r} + N_{2,r})}$$

(Generalized) Linear Interpolation

$$P(w_n | w_1 \dots w_{n-1}) = \sum_{i=1}^n \lambda_i(w_{n+1-i} \dots w_{n-1}) P_i(w_n | w_{n+1-i} \dots w_{n-1})$$

where for every $w_{n+1-i} \dots w_{n-1}$:

$$\sum_{i=1}^n \lambda_i(w_{n+1-i} \dots w_{n-1}) = 1.$$

Jelinek-Mercer smoothing:

$$\lambda_i(w_{n+1-i} \dots w_{n-1}) \equiv \lambda_i.$$

Katz “back-off” smoothing: every λ_i is 0 except the one corresponding to the model used by backoff (which is 1). For some threshold, k :

$$\lambda_i(w_{n+1-i} \dots w_{n-1}) \equiv \begin{cases} \alpha(w_{n+1-i} \dots w_{n-1}) & \text{if for all } i < j \leq n, \\ & C(w_{n+1-j} \dots w_{n-1}) \leq k, \\ & \text{but } C(w_{n+1-i} \dots w_{n-1}) > k \\ 0 & \text{o.w.} \end{cases}$$

Back-off Smoothing

The α parameters normalize such that only the remaining probability mass is distributed among n -grams below the threshold k :

$$\begin{aligned}
 & \alpha(w_{n+1-i} \dots w_{n-1}) \\
 &= \frac{1 - \sum_{w_n: C(w_{n+1-i} \dots w_n) > k} P(w_n | w_{n+1-i} \dots w_{n-1})}{1 - \sum_{w_n: C(w_{n+1-i} \dots w_n) > k} P(w_n | w_{n+2-i} \dots w_{n-1})} \\
 &= \frac{1 - \sum_{w_n: C(w_{n+1-i} \dots w_n) > k} P(w_n | w_{n+1-i} \dots w_{n-1})}{\sum_{w_n: C(w_{n+1-i} \dots w_n) \leq k} P(w_n | w_{n+2-i} \dots w_{n-1})}
 \end{aligned}$$

Parameter Tying

In practice, we never use a different

$\lambda_i(w_{n+1-i} \dots w_{n-1})$ for every history — instead we group them together by *tying* all the

$\lambda_i(w_{n+1-i} \dots w_{n-1})$ together for the histories that have the same average number of non-zero counts:

let

$$f(w_{n+1-i} \dots w_{n-1}) = \frac{C(w_{n+1-i} \dots w_{n-1})}{|w_n : C(w_{n+1-i} \dots w_n) > 0|}$$

For any pair of histories, $w_{n+1-i} \dots w_{n-1}$ and

$w'_{n+1-i} \dots w'_{n-1}$, we take $\lambda(w_{n+1-i} \dots w_{n-1}) \equiv \lambda(w'_{n+1-i} \dots w'_{n-1})$ precisely when

$$f(w_{n+1-i} \dots w_{n-1}) = f(w'_{n+1-i} \dots w'_{n-1}).$$