Logistics

• Short lecture: ~20-25 min

• Breakout rooms (random): ~15-20 min

• Gather back and discuss main points: ~10 min

• Each breakout room will designate a person who can relay the group’s thoughts
Smoothing

- Handle sparsity by making sure all probabilities are non-zero in our model
  - **Additive**: Add a small amount to all probabilities
  - **Discounting**: Redistribute probability mass from observed n-grams to unobserved ones
  - **Back-off**: Use lower order n-grams if higher ones are too sparse
  - **Interpolation**: Use a combination of different granularities of n-grams
Discounting

- Determine some “mass” to remove from probability estimates
- Redistribute mass among unseen n-grams
- Just choose an absolute value to discount (usually <1)

\[
P_{abs\_discount}(w_i | w_{i-1}) = \frac{c(w_{i-1}, w_i) - d}{c(w_{i-1})} \quad \text{if } c(w_{i-1}, w_i) > 0
\]

for all \( w' \) s.t. \( c(w_{i-1}, w') = 0 \) if \( c(w_{i-1}, w_i) = 0 \)

\[
\lambda(w_{i-1}) \frac{P(w_i)}{\sum_{w'} P(w')} \quad \text{Unigram probabilities}
\]
Interpolated Discounting

- Determine some “mass” to remove from probability estimates
- Redistribute mass among unseen n-grams
- Just choose an absolute value to discount (usually <1)

\[
P_{\text{abs\_discount}}(w_i | w_{i-1}) = \frac{\max(0, c(w_{i-1}, w_i) - d)}{c(w_{i-1})} + \lambda(w_{i-1})P(w_i)
\]
Issues with Discounting

\[ P_{\text{abs\_discount}}(w_i | w_{i-1}) = \frac{\max(0, c(w_{i-1}, w_i) - d)}{c(w_{i-1})} + \lambda(w_{i-1})P(w_i) \]

- I can’t read without my reading ____________

- “glasses” more likely filler than “Kong”....

- … but P(Kong) > P(glasses)!
  (maybe since Hong Kong appears a lot in the text)

- Simple unigram probability may not suffice!
A possible solution

• Instead of unigram probability, let us weight words by how many unique bigrams they complete

• i.e. $P_{\text{cont}}(w_i) \propto |\{v : C(vw_i) > 0\}|$

$$\implies P_{\text{cont}}(w_i) = \frac{|\{v : C(vw_i) > 0\}|}{\sum_w |\{v : C(vw) > 0\}|}$$

• With this, words appearing in only a few possible contexts (e.g. Kong) get downweighted
Kneser-Ney smoothing (interpolated)

- $P_{KN}(w_i \mid w_{i-1}) = \frac{\text{max}(0, c(w_{i-1}, w_i) - d)}{c(w_{i-1})} + \lambda(w_{i-1})P_{\text{cont}}(w_i)$

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

- $\lambda(w_{i-1})$ is the mass obtained by discounting, $P_{\text{cont}}(w_i)$ is the relative weight/share of each word within that $\lambda$

- Why interpolated? Because we add back part of the mass also to seen n-grams
Kneser-Ney smoothing (interpolated)

- In general, one can perform this discounting recursively for higher-order n-grams.

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

- where \( c_{KN}(\cdot) = \begin{cases} \text{count}(\cdot) & \text{for the highest order} \\ \text{continuation count}(\cdot) & \text{for lower orders} \end{cases} \)

- and the final term \( P_{KN}(w) = \frac{\max(c_{KN}(w) - d, 0)}{\sum_{w'} c_{KN}(w')} + \lambda(\epsilon) \frac{1}{V} \)

- Here \( \epsilon \) is empty string since there is no context for unigram.

- Final term helps handle unseen unigrams (or words).
Stupid backoff

\[ S(w_i|w_{i-k+1}) = \begin{cases} \frac{\text{count}(w_{i-k+1})}{\text{count}(w_{i-k+1})} & \text{if } \text{count}(w_{i-k+1}) > 0 \\ \lambda S(w_i|w_{i-k+2}) & \text{otherwise} \end{cases} \]

- Back-off from higher to lower order n-grams without any discounting
- Not a valid probability distribution…
- … but works well in practice!

Figure 5: BLEU scores for varying amounts of data using Kneser-Ney (KN) and Stupid Backoff (SB).

(Brants et al., 2007)