Natural Language Processing
Guidelines for Assignment 2

Note: do not include $<$ s $>$ or $<$ /s $>$ in the vocabulary (V) or when counting tokens (N).

Laplace Unigram

$$P(<\text{s}> \text{ I love Soho } <\text{/s}>) = P(\text{I}) * P(\text{love}) * P(\text{Soho})$$

$$P(\text{I}) = \frac{count(\text{I}) + 1}{N + V}$$

Laplace Bigram

$$P(<\text{s}> \text{ I love Soho } <\text{/s}>) = P(\text{I}/<\text{s}>) * P(\text{love}/\text{I}) * P(\text{Soho}/\text{love}) * P(<\text{/s}>/\text{Soho})$$

$$P(\text{I}/<\text{s}>) = \frac{count(<\text{s}>, \text{I}) + 1}{count(<\text{s}>) + V}$$

Stupid Backoff

$$P(<\text{s}> \text{ I love Soho } <\text{/s}>) = P(\text{I}/<\text{s}>) * P(\text{love}/\text{I}) * P(\text{Soho}/\text{love}) * P(<\text{/s}>/\text{Soho})$$

$$P(\text{I}/<\text{s}>) = \begin{cases} \frac{count(<\text{s}>,\text{I})}{count(<\text{s}>)} & \text{if } count(<\text{s}>, \text{I}) > 0 \\ 0.4 * \frac{count(\text{I})+1}{N+V} & \text{otherwise} \end{cases}$$

Kneser-Ney

$$P(<\text{s}> \text{ I love Soho } <\text{/s}>) = P(\text{I}/<\text{s}>) * P(\text{love}/\text{I}) * P(\text{Soho}/\text{love}) * P(<\text{/s}>/\text{Soho})$$

$$P(\text{I}/<\text{s}>) = \begin{cases} \frac{numerator}{count(<\text{s}>)} & \text{if } numerator > 0 \\ \frac{count(\text{I})+1}{N+V} & \text{otherwise} \end{cases}$$

$$numerator = \max\left(count(<\text{s}>, \text{I}) - d, 0\right) + d * nNext(<\text{s}>) * nPrev(\text{I})/nbigrams$$

where nbigrams is the total nummber of bigrams, nNext(w) is the total number of continuations for w and and nPrev(w) is the total number of word types seen to precede w.