# Language Models

- Assignment of probabilities to sequences of words
  - Can be used incrementally to predict the next word
- N-gram
  - Sequence of $n$ words (bigram, trigram, ...)
  - The size of the corpus constrains $n$
    - Can go high on web-scale data
    - In 2006, Google released $10^9$ (1, 2, 3, 4, 5)-grams occurring $\geq 40$ times in corpus of $10^{12}$ words ($1.3 \times 10^6$ unique)

# Predicting a Word

## Language models: bigram, trigram, n-gram

- Sequence of words: $w_1 \ldots w_n$
- $w_i^j$ means $w_i \ldots w_j$
- Chain rule: $P(w_1^n) = P(w_1)P(w_2|w_1) \ldots P(w_n|w_1^{n-1})$
- Not quite usable. Why?
  - Language use is creative
  - Huge amount of data needed to get enough coverage
- Bigram: Assume $P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-1})$
- Trigram: Look at two words in the past
- $n$-gram: Look at $n-1$ words in the immediate past

# Maximum Likelihood Estimation (MLE)

## Technique to estimate probabilities

- Symbols for sentence start <s> and end </s>
- Obtain a corpus
  - Calculate relative frequencies (bigram count ÷ unigram count)
- $P(w_n|w_{n-1}) = \frac{\text{count}(w_{n-1}w_n)}{\text{count}(w_{n-1})}$

Example:

| |
|---|
| <s> I am Sam </s> |
| <s> Sam I am </s> |
| <s> I do not like green eggs and ham </s> |

# Evaluation

- Extrinsic
  - Real-world usage
- Intrinsic
  - From the data itself—based on held out data seen only at the end
  - Split into training and test data
  - Safer to split into training, development (devset), and test data
  - n-fold testing

# Perplexity (Indicates Quality of a Model)

How much information a model needs to achieve accuracy: Lower information needed is better

- $N$th root of the inverse probability of the test set

$$
\begin{aligned}
PP(W) &= P(w_1 \ldots w_N)^{-1/N} \\
&= \sqrt[N]{\frac{1}{P(w_1 \ldots w_N)}} \\
&= \sqrt[N]{\prod_i^N \frac{1}{P(w_i | w_1 \ldots w_{i-1})}}
\end{aligned}
$$

- Weighted average branching factor of a language
  - Branching factor: the number of possible next words that can follow any word
  - Weighted by probability
- Calculate for the *Sam I Am* stanza

# Sparsity

- Rare n-grams may not appear in the corpus
- Zero count $\Rightarrow$ Estimated probability of zero
  - But human language is creative
  - So utterances with prior zero count do occur

# Unknown (Out of Vocabulary) Words

- Closed vocabulary
  - Assume all unknown words are the same `<UNK>`
- Open vocabulary
  - Treat all rare words as the same `<UNK>`
  - Treat the top $N$ most frequent words as words and replace the rest by `<UNK>`
- The number of unknown words can be over-estimated when a language has complex inflected forms
  - Stemming can reduce (apparent) unknowns but is a coarse approach
- Perplexity can be lowered by making the vocabulary smaller

# Smoothing

- Calculate for the *Sam I Am* stanza as a corpus
- Adjusted counts, $c^*$
- Discounting (i.e., reducing) of the nonzero counts
  - Frees up some probability mass to assign to the zero counts
- Laplace: add 1 to each count
  - Simple
  - Invented by Pierre-Simon Laplace in the early days of Bayesian reasoning
  - Since there so many zero count bigrams, Laplace takes away too much probability mass from the nonzero counts
- Add $k$ smoothing ($k < 1$)
  - Requires tuning, via devset

# Backoff

- Backoff: Reduce context when insufficient data
  - If not enough trigrams, use bigram (of last two)
  - If not enough bigrams, use unigram
- Interpolation: combine all n-gram estimators
  - Linear combination of probabilities estimated from unigram, bigram, trigram counts
- Use held-out corpus to estimate

# Kneser-Ney Smoothing

- Based on an empirical observation
  - Get counts of n-grams from one corpus
  - Get counts of the n-grams from a held-out corpus
  - The average counts in the second corpus are lower by about 0.75 (or 0.80) for bigrams
  - Bigrams of count zero are more popular in the second
  - Bigrams of count 1 average about 0.5
- Gale and Church: reduce by 0.75 for bigrams of counts of 3 or higher and place that probability mass on counts of bigrams 0 and 1
- Kneser-Ney
  - $P(\text{continuation}) \propto$ number of times a unigram has appeared in a distinct context—as second words of bigrams
  - Interpolate based on $P(\text{continuation})$