Foundations of Natural Language Processing
Semester 2, 2012-2013

Alex Lascarides
alex@inf.ed.ac.uk

School of informatics

Lecture 6 – Backoff
J&M 4.6–4.8
1st February 2013

Outline

Reminder: Good Turing Smoothing

- Push every probability total down to the count class below.
- Each *count* is reduced slightly (Zipf): we're discounting!

c	N_c	P_c	$P_c[total]$	$c*$	$P*_c$	$P*_c[total]$
0	N_0	0	0	$\frac{N_1}{N_0}$	$\frac{\frac{N_1}{N_0}}{N}$	$\frac{N_1}{N}$
1	N_1	$\frac{1}{N}$	$\frac{N_1}{N}$	$2\frac{N_2}{N_1}$	$\frac{2\frac{N_2}{N_1}}{N}$	$\frac{2N_2}{N}$
2	N_2	$\frac{2}{N}$	$\frac{2N_2}{N}$	$3\frac{N_3}{N_2}$	$\frac{3\frac{N_3}{N_2}}{N}$	$\frac{3N_3}{N}$

- c: count
- N_c: number of different items with count c
- P_c: MLE estimate of prob. of that item
- $P_c[total]$: total probability mass for *all* items with that count.
- $c*$: Good-Turing smoothed version of the count
- $P*_c$ and $P*_c[total]$: Good-Turing versions of P_c and $P_c[total]$

Problem

- 2 trigrams that are not seen in *Moby Dick* are
  - I spent three
    I spent pretty
- GT smoothing makes them equally likely.
- GT smoothing for these unseen trigrams doesn't exploit the relative likelihood of the bigrams spent pretty vs. spent three.
- Those bigrams might have different observed frequencies; we should exploit that!

Hybrid N-Gram Models: Backoff

- Backoff is an alternative to smoothing.
- If a particular trigram (e.g., *three years before*) has zero frequency, then
- you backoff to the bigram *years before*, and count that...
- and if that's zero, then you can backoff again to the unigram model (i.e., *before*).

Example: *three years before* and *Moby Dick*

- 96 occurrences of *years*.
- 33 types of bigram that start with *years*.
  - *years before* is the 5th most frequent among these, with 3 of them.

In General

1. Fill any gap in n-grams by looking 'back' to $n-1$-grams.
2. Or $n-2$ if $n-1$ count is 0 too.
3. If 0 count at unigram level, then smooth.

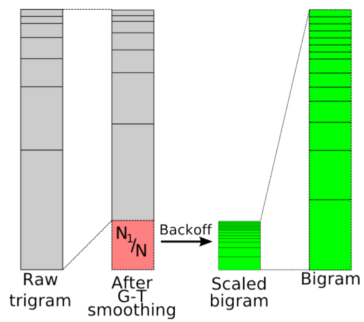
But Not So Fast!

We can't just add backed off probabilities to MLE probabilities!

- Only trigram beginning *I spent* is *I spent in*.
- So $P(in|I, spent) = 1$ (via MLE).
- Bigram *spent three* occurs, as 1 of 9 bigram tokens starting *spent*.
- So $P(three|spent) = \frac{1}{9} = 0.1111$.
- And $P(three|I, spent) = 0.1111$, by backing off to bigrams.
- Uh oh!
- Adding these makes probability mass for trigrams starting *I spent* > 1!!

Discounted Backoff: Make grey and green bits into a probability function!

- Solve the problem in a similar way to Good-Turing smoothing.
- Discount the trigram-based probability estimates.
- This leaves some probability mass to share among the estimates from the lower-order model(s).
- Katz backoff: Good-Turing discount the observed counts, but
- instead of using that saved mass for unseen items, use it for backoff estimates.



The Formulae for Katz Backoff (for trigram)

$$P_{katz}(w_3|w_1,w_2) = \begin{cases} P*(w_3|w_1,w_2) & \text{if } C(w_1,w_2,w_3) > 0 \\ \alpha(w_1,w_2)P_{katz}(w_3|w_2) & \text{otherwise} \end{cases}$$

- $P*$ the discounted probability (computed via Good-Turing)
- α is a normalising factor (details in J&M).
- Formula is recursive:
  you keep backing off until you hit a non-zero count!

Katz Threshold

- Recall how with Good-Turing smoothing, you could set a threshold above which you trusted the count completely and didn't discount it.
- You can do the same for the Katz backoff model.
- In other words, treat the MLE-probability estimate as reliable for high frequency items.
- So you set a frequency threshold, above which you don't discount
  - Replace $P*$ with P_{mle} in the Katz backoff formula.
- Usually, the threshold is around 5–7, making all discounting happen on items of frequency 1–4.

Data is Messy!

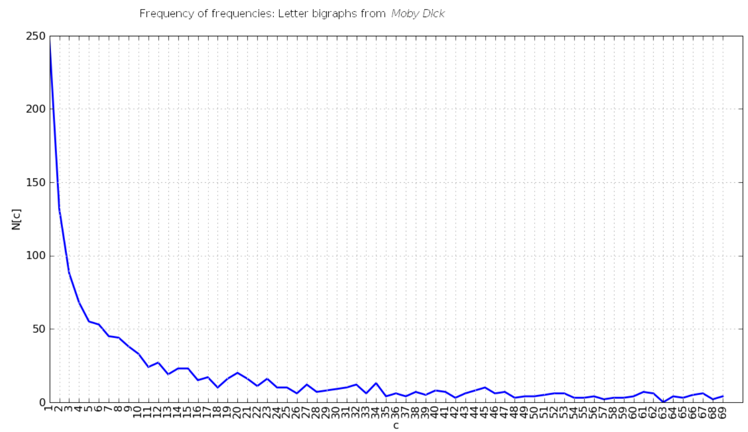
- All these approaches make some background assumptions.
- Real data isn't always as clean as those assumptions require.

Good-Turing Assumptions about the Data

1. Counts are dense:
   - That is, there is no c such that $c < c_{max}$ and $N_c = 0$.

   Zipf's law practically guarantees that there *will* be gaps among higher cs!
2. N_c decreases as c increases.
   Tend to get 'blips' with higher cs.

N_c and c for bigrams in *Moby Dick*



- We've already seen some workarounds.
  - Regression fitting for when $N_c = 0$ and $c < c_{max}$.

Summary

- Backoff estimation is an alternative to smoothing for dealing with sparse data.
- It is sensitive to predictions based on an observed context in a way that smoothing is not.
  - $P_{katz}(before|three, \ years) \neq P_{katz}(pretty|three, \ years)$ even though both are unseen.
- You re-use some of the maths concerning discounting from Good-Turing smoothing
  so as to ensure that your probability mass isn't > 1.
- But both backoff and smoothing methods make assumptions about the way the data behaves:
  - $N_c < N_{c'}$ iff $c_{max} \geq c > c'$
- While this is generally true for low c and c', it begins to be false for higher c and c'.
- But using linear regression can help to smooth out the blips in the curve.