CS 679: Natural Language Processing

Lectures #5: Language Model Smoothing: Discounting

Thanks to Dan Klein of UC Berkeley for many of the materials used in this lecture. Others by Eric Ringger.

Announcements

- Assignment #1
  - Due: Tuesday
- Reading Report #3
  - M&S 6.3-end (of ch. 6)

Objectives

- Discuss metrics for language models
- Discuss LM rescoring
- Get comfortable with the process of factoring and smoothing a joint model of a familiar object: text!
- Look closely at discounting techniques for smoothing language models.

Review: Language Models

- Is a Language Model (i.e., Markov Model) a joint or conditional distribution? Over what?
  \[ P(\text{sentence}) = P(w_1 \ldots w_n) \]
- Is a local model a joint or conditional distribution? Over what?
  \[ P(w_i|w_{i-1} \ldots w_{i-m}) \]
- How are the Markov model and the local model related?
  \[ P(w_1 \ldots w_n) = \prod_{i=1}^{n} P(w_i|w_{i-1} \ldots w_{i-m}) \]

Measuring Model Quality

- Word Error Rate (WER) = \[
\frac{\text{insertions} + \text{deletions} + \text{substitutions}}{\text{true sentence size}}
\]

Correct answer: Andy saw a part of the movie
Recognizer output: And he saw apart of a movie

WER: 5/7 = 71%

- Focused on task-level errors
- The “right” measure or speech recognition, OCR, handwriting reco.

Speech Recognition

- Recall from CS 401R
  - Or refer back to those lectures.
- The value of language models
Speech Reco. & “N-best Lists”

1. How to wreck a nice beach.
2. How to recognize beach.
3. How to recognize a beach.
4. How to recognize uh beach.
5. How to wreck a nice peach.
6. How to recognize speech.
7. How to wreck a nice speech.
8. ...

Language Model Rescoring

1. How to wreck a nice beach.
2. How to recognize speech.
3. How to recognize a beach.
4. How to recognize uh beach.
5. How to wreck a nice peach.
6. How to recognize speech.
7. How to recognize uh beach.
8. ...

New Improved Language Model

Measuring Model Quality

- The Shannon Game:
  - How well can we predict the next word?
    - When I order pizza, I wipe off the____
    - Many children are allergic to____
    - I saw a____
  - You're really good at this.
  - A unigram model is terrible at this! (Why?)

Count: the average number of guesses to fill each word slot.

- Cross Entropy
  - Of a text $S$ (containing $n$ tokens) according to some language model $M$
  - Under the assumption that language is ergodic*,

$$H_M(S) = H_{\text{cross-entrop}}(S, P_M) = -\frac{1}{n} \log_2 P_M(S) = -\frac{1}{n} \log \prod_{i=1}^{n} P_M(w_{i,j} | w_{i,j-4})$$

$$= -\frac{1}{n} \sum_{i=1}^{n} \log_2 P_M(w_{i,j} | w_{i,j-1})$$

* ergodic: relating to a process in which every sequence or sample of sufficient size is equally representative of the whole
Measuring Model Quality

- Cross-entropy:
  - Measured in bits

- Solution: Perplexity
  - Measured in terms of the number of possibilities
  \[ PP_m(S) = 2^{\frac{H_e(S)}{\ln 2}} = \sqrt[n]{\prod_{i=1}^{n} P_m(w_i | \Sigma_{i-1,i}^m)} \]
  - Note that even though our models require a stop (<s>) step, we don’t count it as a symbol when computing these averages.

Discounting

- Starting point: Maximum Likelihood Estimator (MLE)
- a.k.a. Empirical Estimator
  \[ P_{\text{MLE}}(w|h) = \frac{c(w,h)}{c(h)} \]
  - Key discounting problem:
    - What count \( c^*(w,h) \) should we use for an event that occurred \( c(w,h) \) times in \( N \) samples?
    - Let \( r^* = c^*(w,h) \) as a short-hand notation.
  - Corollary problem:
    - What probability \( p^* = \frac{c^*}{c} \) should we use for an event that occurred with probability \( p = \frac{c}{N} \)

Discounting: Add-One Estimation

- Idea: pretend we saw every n-gram once more than we actually did [Laplace]
  \[ \hat{P}(w|h) = \frac{c(w,h) + 1}{c(h) + V} \]
- And for unigrams:
  \[ \hat{P}(w) = \frac{c(w) + 1}{N + V} \]
  - Observed count \( r > 0 \) → re-estimated count \( 0 < r^* < r \)
  - \( V = N_1 + N_0 \)
  - Reserved \( \frac{1}{N_1 + N_0} \) for extra events
  - Observed \( \frac{1}{N_1} \) is distributed back to seen n-grams
  - "Unobserved" \( \frac{1}{N_0} \) is reserved for unseen words (nearly all of the vocab!)
  - Actually tells us not only how much to hold out, but where to put it (evenly).
  - Works astonishingly poorly in practice.

Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
<td>number of word tokens in training data</td>
</tr>
<tr>
<td>( c(w) )</td>
<td>frequency (count) of word tokens ( w ) in training data</td>
</tr>
<tr>
<td>( c(w,h) )</td>
<td>frequency (count) of word token ( w ) following history ( h ) in training data</td>
</tr>
<tr>
<td>( r^* )</td>
<td>estimated (reweighted) frequency of n-gram</td>
</tr>
<tr>
<td>( r^*_{\text{V}} )</td>
<td>estimated (reweighted) frequency of n-gram</td>
</tr>
<tr>
<td>( V )</td>
<td>total vocabulary size (number of word types in training data)</td>
</tr>
<tr>
<td>( N_0 )</td>
<td>number of word types with count ( r ) (wouldn't it be nicer as ( V_0 ))</td>
</tr>
<tr>
<td>( p^* )</td>
<td>( \frac{c^*}{c} ) probability of n-gram, on training data</td>
</tr>
<tr>
<td>( p^*_{\text{V}} )</td>
<td>( \frac{c^<em>}{c^</em> + V} ) estimated probability of n-gram</td>
</tr>
</tbody>
</table>
Discounting: Add-Epsilon

- Quick fix: add some small $\epsilon$ instead of 1
  [Lidstone, Jefferys]

$$P_{add-\epsilon}(w|h) = \frac{c(w,h) + \epsilon}{c(h) + \epsilon V}$$

- Slightly better, holds out less mass
- Still a bad idea

Discounting: Add-Epsilon

- Let $\epsilon = \delta \cdot \left( \frac{1}{V} \right)$

$$P_{add-\epsilon}(w|h) = \frac{c(w,h) + \epsilon}{c(h) + \epsilon V} = \frac{c(w,h) + \delta (1/V)}{c(h) + \delta}$$

- A uniform prior over vocabulary, from a Bayesian p.o.v.

Discounting: Unigram Prior

- Better to assume a unigram prior

$$P_{add-\text{unigram}}(w|h) = \frac{c(w,h) + \delta (1/V)}{c(h) + \delta}$$

$$P_{add-\text{unigram}}(w|h) = \frac{c(w,h) + \delta \tilde{P}(h)}{c(h) + \delta}$$

- Where the unigram is empirical or smoothed itself:

$$\tilde{P}(h) = \frac{c(h) + \delta / V}{N + \delta}$$

**CHECK:** $\sum_h \tilde{P}(h) = \delta$

How Much Mass to Withhold?

- Remember the key discounting problem:
- What count $r^*$ should we use for an event that occurred $r$ times in $N$ samples?
- For the rich, $r$ is too big
- For the poor, $r$ is too small (0)

$$r = \epsilon (\omega | h) \rightarrow r^* = \epsilon (\omega | h)$$

Idea: Use Held-out Data

Sample #1: N words (or n-grams)
- the
- interest
- rates
- Federal

Sample #2: N words (or n-grams)
- the
- interest
- rates
- Federal

$r = 2$

$r^* = $

[Jelinek and Mercer]
Discounting: Held-out Estimation

- **Summary:**
  - Get $N$ samples (sample #1)
  - Collect the set of types occurring $r$ times (in sample #1)
  - Get another $N$ samples (sample #2)
  - What is the average frequency of those types in sample #2?
    - e.g., in our example, doubletons occurred on average 1.5 times
  - Use that average as $r^*$
    - $r^* = 2 \rightarrow r^* = 1.5$
- In practice, much better than add-one, etc.
- Main issue: requires large (equal) amount of held-out data

What’s Next

- **Upcoming lectures:**
  - More sophisticated discounting strategies