Language Models

January 22, 2013
Still no MT??

• Today we will talk about models of $p(\text{sentence})$
• The rest of this semester will deal with $p(\text{translated sentence} \mid \text{input sentence})$
• Why do it this way?
  • Conditioning on more stuff makes modeling more complicated. That is: $p(\text{sentence})$ is easier than $p(\text{translated sentence} \mid \text{input sentence})$.
  • Language models are arguably the most important models in statistical MT
My legal name is Alexander Perchov.
My legal name is Alexander Perchov. But all of my many friends dub me Alex, because that is a more flaccid-to-utter version of my legal name.
My legal name is Alexander Perchov. But all of my many friends dub me Alex, because that is a more flaccid-to-utter version of my legal name. Mother dubs me Alexi-stop-spleening-me!, because I am always spleening her.
My legal name is Alexander Perchov. But all of my many friends dub me Alex, because that is a more flaccid-to-utter version of my legal name. Mother dubs me Alexi-stop-spleening-me!, because I am always spleening her. If you want to know why I am always spleening her, it is because I am always elsewhere with friends, and disseminating so much currency, and performing so many things that can spleen a mother.
My legal name is Alexander Perchov. But all of my many friends dub me Alex, because that is a more flaccid-to-utter version of my legal name. Mother dubs me Alexi-stop-spleening-me!, because I am always spleening her. If you want to know why I am always spleening her, it is because I am always elsewhere with friends, and disseminating so much currency, and performing so many things that can spleen a mother. Father used to dub me Shapka, for the fur hat I would don even in the summer month.
My legal name is Alexander Perchov. But all of my many friends dub me Alex, because that is a more flaccid-to-utter version of my legal name. Mother dubs me Alexi-stop-spleening-me!, because I am always spleening her. If you want to know why I am always spleening her, it is because I am always elsewhere with friends, and disseminating so much currency, and performing so many things that can spleen a mother. Father used to dub me Shapka, for the fur hat I would don even in the summer month. He ceased dubbing me that because I ordered him to cease dubbing me that.
My legal name is Alexander Perchov. But all of my many friends dub me Alex, because that is a more flaccid-to-utter version of my legal name. Mother dubs me Alexi-stop-spleening-me!, because I am always spleening her. If you want to know why I am always spleening her, it is because I am always elsewhere with friends, and disseminating so much currency, and performing so many things that can spleen a mother. Father used to dub me Shapka, for the fur hat I would don even in the summer month. He ceased dubbing me that because I ordered him to cease dubbing me that. It sounded boyish to me, and I have always thought of myself as very potent and generative.
Test data BLEU vs. LM training data size in million tokens.

- +0.66BP/x2
- target SB
Language Models Matter

- Language models play the role of ...
  - a judge of grammaticality
  - a judge of semantic plausibility
  - an enforcer of stylistic consistency
  - a repository of knowledge (?)
What is the probability of a sentence?

• Requirements

• Assign a probability to every sentence (i.e., string of words)
What is the probability of a sentence?

• Requirements
  • Assign a probability to every sentence (i.e., string of words)

• Questions
  • How many sentences are there in English?
  • Too many :)
What is the probability of a sentence?

- Requirements
  - Assign a probability to every sentence (i.e., string of words)

\[ \sum_{e \in \Sigma^*} p_{LM}(e) = 1 \]
\[ p_{LM}(e) \geq 0 \quad \forall e \in \Sigma^* \]
$n$-gram LMs

$p_{LM}(e)$
\[ p_{\text{LM}}(e) \]

Vector-valued random variable

\( n \)-gram LMs
\( n \)-gram LMs

\[ p_{\text{LM}}(e) \]
$n$-gram LMs

$$p_{LM}(e) = p(e_1, e_2, e_3, \ldots, e_{\ell})$$
$n$-gram LMs

\[ p_{LM}(e) = p(e_1, e_2, e_3, \ldots, e_\ell) \]
\[ = p(e_1) \times \]
\[ p_{LM}(e) = p(e_1, e_2, e_3, \ldots, e_\ell) \]
\[ = p(e_1) \times \]
\[ p(e_2 \mid e_1) \times \]
**n-gram LMs**

\[ p_{LM}(e) = p(e_1, e_2, e_3, \ldots, e_\ell) = p(e_1) \times p(e_2 \mid e_1) \times p(e_3 \mid e_1, e_2) \times p(e_4 \mid e_1, e_2, e_3) \times \cdots \times p(e_\ell \mid e_1, e_2, \ldots, e_{\ell-2}, e_{\ell-1}) \]
\[ n\text{-gram LMs} \]

\[ p_{LM}(e) = p(e_1, e_2, e_3, \ldots, e_\ell) \]

\[ \approx p(e_1) \times p(e_2 \mid e_1) \times p(e_3 \mid e_1, e_2) \times p(e_4 \mid e_1, e_2, e_3) \times \cdots \times p(e_\ell \mid e_1, e_2, \ldots, e_{\ell-2}, e_{\ell-1}) \]
n-gram LMs

\[ p_{\text{LM}}(e) = p(e_1, e_2, e_3, \ldots, e_\ell) \]

\[ \approx p(e_1) \times \]

\[ p(e_2 \mid e_1) \times \]

\[ p(e_3 \mid e_1, e_2) \times \]

\[ p(e_4 \mid e_1, e_2, e_3) \times \]

\[ \cdots \times \]

\[ p(e_\ell \mid e_1, e_2, \ldots, e_{\ell-2}, e_{\ell-1}) \]
**n-gram LMs**

\[
p_{LM}(e) = p(e_1, e_2, e_3, \ldots, e_\ell) \\
\approx p(e_1) \times \\
p(e_2 | e_1) \times \\
p(e_3 | e_1, e_2) \times \\
p(e_4 | e_1, e_2, e_3) \times \\
\cdots \times \\
p(e_\ell | e_1, e_2, \ldots, e_{\ell-2}, e_{\ell-1})
\]

**Which do you think is better? Why?**
\( n \)-gram LMs

\[
p_{\text{LM}}(e) = p(e_1, e_2, e_3, \ldots, e_\ell) \\
\approx p(e_1) \times \\
p(e_2 | e_1) \times \\
p(e_3 | e_1, e_2) \times \\
p(e_4 | e_1, e_2, e_3) \times \\
\cdots \times \\
p(e_\ell | e_1, e_2, \ldots, e_{\ell-2}, e_{\ell-1})
\]
**n-gram LMs**

\[ p_{LM}(e) = p(e_1, e_2, e_3, \ldots, e_\ell) \]

\[ \approx p(e_1) \times \]
\[ p(e_2 \mid e_1) \times \]
\[ p(e_3 \mid e_1, e_2) \times \]
\[ p(e_4 \mid e_1, e_2, e_3) \times \]
\[ \cdots \times \]
\[ p(e_\ell \mid e_1, e_2, \ldots, e_{\ell-2}, e_{\ell-1}) \]

\[ = p(e_1 \mid \text{START}) \times \prod_{i=2}^{\ell} p(e_i \mid e_{i-1}) \times p(\text{STOP} \mid e_\ell) \]
START
$p(\text{my} \mid \text{START})$
START   my   friends

\[ p(\text{my} \mid \text{START}) \times p(\text{friends} \mid \text{my}) \]
START my friends call

\[ p(\text{my} | \text{START}) \times p(\text{friends} | \text{my}) \times p(\text{call} | \text{friends}) \]
START     my     friends     call     me

\[ p(\text{my} | \text{START}) \times p(\text{friends} | \text{my}) \times p(\text{call} | \text{friends}) \times p(\text{me} | \text{call}) \]
START  my  friends  call  me  Alex

\( p(\text{my} \mid \text{START}) \times p(\text{friends} \mid \text{my}) \times p(\text{call} \mid \text{friends}) \times p(\text{me} \mid \text{call}) \times p(\text{Alex} \mid \text{me}) \)
my friends call me Alex

\[ p(\text{my} | \text{START}) \times p(\text{friends} | \text{my}) \times p(\text{call} | \text{friends}) \times p(\text{me} | \text{call}) \times p(\text{Alex} | \text{me}) \times p(\text{STOP} | \text{Alex}) \]
my friends call me Alex STOP

\[ p(\text{my} | \text{START}) \times p(\text{friends} | \text{my}) \times p(\text{call} | \text{friends}) \times p(\text{me} | \text{call}) \times p(\text{Alex} | \text{me}) \times p(\text{STOP} | \text{Alex}) \]

my friends dub me Alex STOP

\[ p(\text{my} | \text{START}) \times p(\text{friends} | \text{my}) \times p(\text{dub} | \text{friends}) \times p(\text{me} | \text{dub}) \times p(\text{Alex} | \text{me}) \times p(\text{STOP} | \text{Alex}) \]
$p(\text{my} | \text{START}) \times p(\text{friends} | \text{my}) \times p(\text{call} | \text{friends}) \times p(\text{me} | \text{call}) \times p(\text{Alex} | \text{me}) \times p(\text{STOP} | \text{Alex})$

$\leftarrow$

$\leftarrow$

$p(\text{my} | \text{START}) \times p(\text{friends} | \text{my}) \times p(\text{dub} | \text{friends}) \times p(\text{me} | \text{dub}) \times p(\text{Alex} | \text{me}) \times p(\text{STOP} | \text{Alex})$
These sentences have many terms in common.
Categorical Distributions

A categorical distribution characterizes a random event that can take on exactly one of $K$ possible outcomes.

*(nb. we often call these “multinomial distributions”)*

\[
p(x) = \begin{cases} 
p_1 & \text{if } x = 1 \\
p_2 & \text{if } x = 2 \\
\vdots \\
p_K & \text{if } x = K \\
0 & \text{otherwise}
\end{cases} \quad \sum_i p_i = 1 \\
p_i \geq 0 \quad \forall i
\]
Probability tables like this are the workhorses of language (and translation) modeling.
\[ p(\cdot \mid \text{some context}) \]

<table>
<thead>
<tr>
<th>Outcome</th>
<th>(p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>0.6</td>
</tr>
<tr>
<td>and</td>
<td>0.04</td>
</tr>
<tr>
<td>said</td>
<td>0.009</td>
</tr>
<tr>
<td>says</td>
<td>0.00001</td>
</tr>
<tr>
<td>of</td>
<td>0.1</td>
</tr>
<tr>
<td>why</td>
<td>0.1</td>
</tr>
<tr>
<td>Why</td>
<td>0.00008</td>
</tr>
<tr>
<td>restaurant</td>
<td>0.000008</td>
</tr>
<tr>
<td>destitute</td>
<td>0.0000064</td>
</tr>
</tbody>
</table>

\[ p(\cdot \mid \text{other context}) \]

<table>
<thead>
<tr>
<th>Outcome</th>
<th>(p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>0.01</td>
</tr>
<tr>
<td>and</td>
<td>0.01</td>
</tr>
<tr>
<td>said</td>
<td>0.003</td>
</tr>
<tr>
<td>says</td>
<td>0.009</td>
</tr>
<tr>
<td>of</td>
<td>0.002</td>
</tr>
<tr>
<td>why</td>
<td>0.003</td>
</tr>
<tr>
<td>Why</td>
<td>0.0006</td>
</tr>
<tr>
<td>restaurant</td>
<td>0.2</td>
</tr>
<tr>
<td>destitute</td>
<td>0.1</td>
</tr>
</tbody>
</table>
\[ p(\cdot | \text{some context}) \quad p(\cdot | \text{in}) \quad p(\cdot | \text{other context}) \quad p(\cdot | \text{the}) \]

<table>
<thead>
<tr>
<th>Outcome</th>
<th>( p )</th>
<th>Outcome</th>
<th>( p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>0.6</td>
<td>the</td>
<td>0.01</td>
</tr>
<tr>
<td>and</td>
<td>0.04</td>
<td>and</td>
<td>0.01</td>
</tr>
<tr>
<td>said</td>
<td>0.009</td>
<td>said</td>
<td>0.003</td>
</tr>
<tr>
<td>says</td>
<td>0.00001</td>
<td>says</td>
<td>0.009</td>
</tr>
<tr>
<td>of</td>
<td>0.1</td>
<td>of</td>
<td>0.002</td>
</tr>
<tr>
<td>why</td>
<td>0.1</td>
<td>why</td>
<td>0.003</td>
</tr>
<tr>
<td>Why</td>
<td>0.00008</td>
<td>Why</td>
<td>0.0006</td>
</tr>
<tr>
<td>restaurant</td>
<td>0.0000008</td>
<td>restaurant</td>
<td>0.2</td>
</tr>
<tr>
<td>destitute</td>
<td>0.00000064</td>
<td>destitute</td>
<td>0.1</td>
</tr>
</tbody>
</table>

\[ \begin{align*}
\mathbb{P}(X|\text{context}) & = p_X(X|\text{context})p(\text{context}) \\
\mathbb{P}(\text{context} | X) & = p_{X|\text{context}}(\text{context} | X)\mathbb{P}(X) \\
\mathbb{P}(X) & = \sum_{\text{context}} \mathbb{P}(X|\text{context})p(\text{context}) \\
\mathbb{P}(\text{context} | X) & = \frac{\mathbb{P}(X|\text{context})p(\text{context})}{\sum_{\text{context}} \mathbb{P}(X|\text{context})p(\text{context})} \\
\end{align*} \]
LM Evaluation

• Extrinsic evaluation: build a new language model, use it for some task (MT, ASR, etc.)

• Intrinsic: measure how good we are at modeling language

We will use **perplexity** to evaluate models

Given: \( w, p_{LM} \)

\[
PPL = 2^{\frac{1}{|w|} \log_2 p_{LM}(w)}
\]

\( 0 \leq PPL \leq \infty \)
Perplexity

- Generally fairly good correlations with BLEU for n-gram models

- Perplexity is a generalization of the notion of branching factor
  - How many choices do I have at each position?

- State-of-the-art English LMs have PPL of ~100 word choices per position

- A uniform LM has a perplexity of $|\Sigma|$

- Humans do much better

- ... and bad models can do even worse than uniform!
Whence parameters?
Whence parameters?

Estimation.
\[
p(x \mid y) = \frac{p(x, y)}{p(y)}
\]
\[
\hat{p}_{\text{MLE}}(x) = \frac{\text{count}(x)}{N}
\]
\[
\hat{p}_{\text{MLE}}(x, y) = \frac{\text{count}(x, y)}{N}
\]
\[
\hat{p}_{\text{MLE}}(x \mid y) = \frac{\text{count}(x, y)}{N} \times \frac{N}{\text{count}(y)}
\]
\[
= \frac{\text{count}(x, y)}{\text{count}(y)}
\]
\[ p(x \mid y) = \frac{p(x, y)}{p(y)} \]

\[ \hat{p}_{\text{MLE}}(x) = \frac{\text{count}(x)}{N} \]

\[ \hat{p}_{\text{MLE}}(x, y) = \frac{\text{count}(x, y)}{N} \]

\[ \hat{p}_{\text{MLE}}(x \mid y) = \frac{\text{count}(x, y)}{N} \times \frac{N}{\text{count}(y)} = \frac{\text{count}(x, y)}{\text{count}(y)} \]
\[
\hat{p}(x | y) = \frac{p(x, y)}{p(y)}
\]

\[
\hat{p}_{\text{MLE}}(x) = \frac{\text{count}(x)}{N}
\]

\[
\hat{p}_{\text{MLE}}(x, y) = \frac{\text{count}(x, y)}{N}
\]

\[
\hat{p}_{\text{MLE}}(x | y) = \frac{\text{count}(x, y)}{N} \times \frac{N}{\text{count}(y)} = \frac{\text{count}(x, y)}{\text{count}(y)}
\]

\[
\hat{p}_{\text{MLE}}(\text{call} | \text{friends}) = \frac{\text{count}(\text{friends call})}{\text{count}(\text{friends})}
\]
MLE & Perplexity

• What is the **lowest (best) perplexity possible** for your model class?

• Compute the MLE!

• Well, that’s easy...
START  my  friends  call  me  Alex  STOP

\[ p(\text{my} | \text{START}) \times p(\text{friends} | \text{my}) \times p(\text{call} | \text{friends}) \times p(\text{me} | \text{call}) \times p(\text{Alex} | \text{me}) \times p(\text{STOP} | \text{Alex}) \]

START  my  friends  dub  me  Alex  STOP

\[ p(\text{my} | \text{START}) \times p(\text{friends} | \text{my}) \times p(\text{dub} | \text{friends}) \times p(\text{me} | \text{dub}) \times p(\text{Alex} | \text{me}) \times p(\text{STOP} | \text{Alex}) \]
p(my | START) × p(friends | my) × p(call | friends) × p(me | call) × p(Alex | me) × p(STOP | Alex)

MLE

p(my | START) × p(friends | my) × p(dub | friends) × p(me | dub) × p(Alex | me) × p(STOP | Alex)

MLE
\[ p(\text{my} | \text{START}) \times p(\text{friends} | \text{my}) \times p(\text{call} | \text{friends}) \times p(\text{me} | \text{call}) \times p(\text{Alex} | \text{me}) \times p(\text{STOP} | \text{Alex}) \]

\textbf{MLE} \quad -3.65172

\[ p(\text{my} | \text{START}) \times p(\text{friends} | \text{my}) \times p(\text{dub} | \text{friends}) \times p(\text{me} | \text{dub}) \times p(\text{Alex} | \text{me}) \times p(\text{STOP} | \text{Alex}) \]

\textbf{MLE} \quad -3.65172
my friends call me Alex

-3.65172 -2.07101

my friends dub me Alex

-3.65172 -2.07101
<table>
<thead>
<tr>
<th>START</th>
<th>my</th>
<th>friends</th>
<th>call</th>
<th>me</th>
<th>Alex</th>
<th>STOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p(\text{my}</td>
<td>\text{START}) \times p(\text{friends}</td>
<td>\text{my}) \times p(\text{call}</td>
<td>\text{friends}) \times p(\text{me}</td>
<td>\text{call}) \times p(\text{Alex}</td>
<td>\text{me}) \times p(\text{STOP}</td>
<td>\text{Alex}) )</td>
</tr>
<tr>
<td>MLE</td>
<td>-3.65172</td>
<td>-2.07101</td>
<td>-3.32231</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>START</th>
<th>my</th>
<th>friends</th>
<th>dub</th>
<th>me</th>
<th>Alex</th>
<th>STOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p(\text{my}</td>
<td>\text{START}) \times p(\text{friends}</td>
<td>\text{my}) \times p(\text{dub}</td>
<td>\text{friends}) \times p(\text{me}</td>
<td>\text{dub}) \times p(\text{Alex}</td>
<td>\text{me}) \times p(\text{STOP}</td>
<td>\text{Alex}) )</td>
</tr>
<tr>
<td>MLE</td>
<td>-3.65172</td>
<td>-2.07101</td>
<td>-\infty</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>my</td>
<td>friends</td>
<td>call</td>
<td>me</td>
<td>Alex</td>
<td>STOP</td>
</tr>
<tr>
<td>-----</td>
<td>------</td>
<td>---------</td>
<td>-------</td>
<td>-------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>START</td>
<td>my</td>
<td>friends</td>
<td>call</td>
<td>me</td>
<td>Alex</td>
<td>STOP</td>
</tr>
<tr>
<td>p(my</td>
<td>START)×p(friends</td>
<td>my)×p(call</td>
<td>friends)×p(me</td>
<td>call)×p(Alex</td>
<td>me)×p(STOP</td>
<td>Alex)</td>
</tr>
<tr>
<td>MLE</td>
<td>-3.65172</td>
<td>-2.07101</td>
<td>-3.32231</td>
<td>-0.271271</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>my</th>
<th>friends</th>
<th>dub</th>
<th>me</th>
<th>Alex</th>
<th>STOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>my</td>
<td>friends</td>
<td>dub</td>
<td>me</td>
<td>Alex</td>
<td>STOP</td>
</tr>
<tr>
<td>p(my</td>
<td>START)×p(friends</td>
<td>my)×p(dub</td>
<td>friends)×p(me</td>
<td>dub)×p(Alex</td>
<td>me)×p(STOP</td>
<td>Alex)</td>
</tr>
<tr>
<td>MLE</td>
<td>-3.65172</td>
<td>-2.07101</td>
<td>-∞</td>
<td>-2.54562</td>
<td></td>
<td></td>
</tr>
<tr>
<td>START</td>
<td>my</td>
<td>friends</td>
<td>call</td>
<td>me</td>
<td>Alex</td>
<td>STOP</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------</td>
<td>---------</td>
<td>-------</td>
<td>----</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>( p(my \mid \text{START}) \times p(friends \mid my) \times p(call \mid friends) \times p(me \mid call) \times p(Alex \mid me) \times p(\text{STOP} \mid Alex) )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLE</td>
<td>-3.65172</td>
<td>-2.07101</td>
<td>-3.2231</td>
<td>-0.271271</td>
<td>-4.961</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>START</th>
<th>my</th>
<th>friends</th>
<th>dub</th>
<th>me</th>
<th>Alex</th>
<th>STOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p(my \mid \text{START}) \times p(friends \mid my) \times p(dub \mid friends) \times p(me \mid dub) \times p(Alex \mid me) \times p(\text{STOP} \mid Alex) )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLE</td>
<td>-3.65172</td>
<td>-2.07101</td>
<td>-\infty</td>
<td>-2.54562</td>
<td>-4.961</td>
<td></td>
</tr>
</tbody>
</table>
\[
p(my \mid \text{START}) \times p(\text{friends} \mid my) \times p(\text{call} \mid \text{friends}) \times p(\text{me} \mid \text{call}) \times p(\text{Alex} \mid \text{me}) \times p(\text{STOP} \mid \text{Alex})
\]

MLE

\[
\begin{array}{cccccc}
-3.65172 & -2.07101 & -3.32231 & -0.271271 & -4.961 & -1.96773
\end{array}
\]

START my friends call me Alex STOP

\[
p(my \mid \text{START}) \times p(\text{friends} \mid my) \times p(\text{dub} \mid \text{friends}) \times p(\text{me} \mid \text{dub}) \times p(\text{Alex} \mid \text{me}) \times p(\text{STOP} \mid \text{Alex})
\]

MLE

\[
\begin{array}{cccccc}
-3.65172 & -2.07101 & -\infty & -2.54562 & -4.961 & -1.96773
\end{array}
\]
MLE assigns probability zero to unseen events.
Zeros

- Two kinds of zero probs:
  - **Sampling zeros**: zeros in the MLE due to impoverished observations
  - **Structural zeros**: zeros that should be there.
    
    *Do these really exist?*

- Just because you haven’t seen something, doesn’t mean it doesn’t exist.

- In practice, we don’t like probability zero, even if there is an argument that it is a structural zero.
Zeros

• Two kinds of zero probs:
  
  • **Sampling zeros**: zeros in the MLE due to impoverished observations
  
  • **Structural zeros**: zeros that should be there.  
  \textit{Do these really exist?}

• Just because you haven’t seen something, doesn’t mean it doesn’t exist.

• In practice, we don’t like probability zero, even if there is an argument that it is a structural zero.

\textit{the a ’s are nearing the end of their lease in oakland}
Smoothing an refers to a family of estimation techniques that seek to model important general patterns in data while avoiding modeling noise or sampling artifacts. In particular, for language modeling, we seek

\[ p(e) > 0 \quad \forall e \in \Sigma^* \]

We will assume that \( \Sigma \) is known and finite.
Add-$\alpha$ Smoothing

\[ p \sim \text{Dirichlet}(\alpha) \]
\[ x_i \sim \text{Categorical}(p) \quad \forall 1 \leq i \leq |x| \]

Assuming this model, what is the most probable value of $p$, having observed training data $x$?

(bunch of calculus - read about it on Wikipedia)

\[ p^*_x = \frac{\text{count}(x) + \alpha_x - 1}{N + \sum_{x'} (\alpha_{x'} - 1)} \quad \forall \alpha_x > 1 \]
Add-$\alpha$ Smoothing

- Simplest possible smoother
- Surprisingly effective in many models
- Does not work well for language models
- There are procedures for dealing with $0 < \alpha < 1$
- When might these be useful?
Interpolation

- "Mixture of MLEs"

\[
\hat{p}(\text{dub} \mid \text{my friends}) = \lambda_3 \hat{p}_{\text{MLE}}(\text{dub} \mid \text{my friends}) + \lambda_2 \hat{p}_{\text{MLE}}(\text{dub} \mid \text{friends}) + \lambda_1 \hat{p}_{\text{MLE}}(\text{dub}) + \lambda_0 \frac{1}{|\Sigma|}
\]
Interpolation

• “Mixture of MLEs”

\[
p(\text{dub} | \text{my friends}) = \lambda_3 \hat{p}_{\text{MLE}}(\text{dub} | \text{my friends}) \\
+ \lambda_2 \hat{p}_{\text{MLE}}(\text{dub} | \text{friends}) \\
+ \lambda_1 \hat{p}_{\text{MLE}}(\text{dub}) \\
+ \lambda_0 \frac{1}{|\Sigma|}
\]

Where do the lambdas come from?
Discounting

Discounting adjusts the frequencies of observed events downward to reserve probability for the things that have not been observed.

Note \( f(w_3 \mid w_1, w_2) > 0 \) only when \( \text{count}(w_1, w_2, w_3) > 0 \)

We introduce a discounted frequency:

\[
0 \leq f^*(w_3 \mid w_1, w_2) \leq f(w_3 \mid w_1, w_2)
\]

The total discount is the zero-frequency probability:

\[
\lambda(w_1, w_2) = 1 - \sum_{w'} f^*(w' \mid w_1, w_2)
\]
Back-off

Recursive formulation of probability:

\[
\hat{p}_\text{BO}(w_3 \mid w_1, w_2) = \begin{cases} 
  \hat{p}_\text{BO}(w_3 \mid w_1, w_2) & \text{if } f^*(w_3 \mid w_1, w_2) > 0 \\
  \alpha_{w_1, w_2} \times \lambda(w_1, w_2) \times \hat{p}_\text{BO}(w_3 \mid \uparrow_1, w_2) & \text{otherwise}
\end{cases}
\]
Back-off

Recursive formulation of probability:

\[
\hat{p}_{\text{BO}}(w_3 \mid w_1, w_2) = \begin{cases} 
  f^*(w_3 \mid w_1, w_2) & \text{if } f^*(w_3 \mid w_1, w_2) > 0 \\
  \alpha_{w_1, w_2} \times \lambda(w_1, w_2) \times \hat{p}_{\text{BO}}(w_3 \mid w_1, w_2) & \text{otherwise}
\end{cases}
\]

“Back-off weight”
Back-off

Recursive formulation of probability:

\[
\hat{p}_{BO}(w_3 \mid w_1, w_2) = \begin{cases} 
    f^*(w_3 \mid w_1, w_2) & \text{if } f^*(w_3 \mid w_1, w_2) > 0 \\
    \alpha_{w_1,w_2} \times \lambda(w_1, w_2) \times \hat{p}_{BO}(w_3 \mid \text{BOS} \mid w_1, w_2) & \text{otherwise}
\end{cases}
\]

“Back-off weight”

Question: how do we discount?
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[ \lambda(a, b) \propto \]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[ a \]

\[ \lambda(a, b) \propto \]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[ \lambda(a, b) \propto \]

\[ a \ b \]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[ \lambda(a, b) \propto | \]

\[ a \ b \ c \]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[ \lambda(a, b) \propto a \]

\[ a \ b \ c \ a \]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[ \lambda(a, b) \propto |a - b| \]

\[ a \ b \ c \ a \ b \]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[ \lambda(a, b) \propto | \]

\[ a \ b \ c \ a \ b \ c \]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[ \ell(a, b) \propto | \begin{array}{cccc}
    a & b & c & a \\
    a & b & c & a
  \end{array} | \]
Let’s assume that the probability of a zero off can be estimated as follows:

\[ \lambda(a, b) \propto | \]

\[ a \ b \ c \ a \ b \ c \ a \ b \]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[ \lambda(a, b) \propto |+| \]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[ a \ b \ c \ a \ b \ c \ a \ b \times \ a \]

\[ \lambda(a, b) \propto |+| \]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[ \lambda(a, b) \propto |+| \]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

$$\lambda(a, b) \propto |a| + |b|$$
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[
\lambda(a, b) \propto |+|
\]
Let’s assume that the probability of a zero off can be estimated as follows:

\[
\lambda(a, b) \propto |a - b|
\]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[
\begin{array}{cccccccc}
  & a & b & c & a & b & c & a & b \\
  \times & a & b & c & c & a & b & a & b
\end{array}
\]

\[
\lambda(a, b) \propto | + | 
\]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[ \lambda(a, b) \propto |+|+| \]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[ \lambda(a, b) \propto \left| + \right| + \left| + \right| \]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[
\lambda(a, b) \propto | + | + |
\]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[ \lambda(a, b) \propto |+|+| \]

(a, b c a b c a b x a b c c a b a b x c)
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[ \lambda(a, b) \propto |+| + | = 3 \]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[
\begin{align*}
    &a \ b \ c \ a \ b \ c \ a \ b \ \times \ a \ b \ c \ c \ a \ b \ a \ b \ \times \ c \\
    \lambda(a, b) \propto |+|+| &= 3 \\
    t(a, b) = |\{x : \text{count}(a, b, x) > 0\}| 
\end{align*}
\]
Witten-Bell Discounting

Let’s assume that the probability of a zero off can be estimated as follows:

\[ \lambda(a, b) \propto |++|^3 \]

\[ t(a, b) = |\{x : \text{count}(a, b, x) > 0\}| \]

\[ \lambda(a, b) = \frac{t(a, b)}{\text{count}(a, b) + t(a, b)} \]
Witten-Bell Discounting

Let's assume that the probability of a zero off can be estimated as follows:

\[
\begin{align*}
\lambda(a, b) &\propto | + | + | = 3 \\
 t(a, b) &= |\{x : \text{count}(a, b, x) > 0\}| \\
 \lambda(a, b) &= \frac{t(a, b)}{\text{count}(a, b) + t(a, b)} \\
 f^*(c | a, b) &= \frac{\text{count}(a, b, c)}{\text{count}(a, b) + t(a, b)}
\end{align*}
\]
Kneser-Ney Discounting

• State-of-the-art in language modeling for 15 years
• Two major intuitions
  • Some contexts have lots of new words
  • Some words appear in lots of contexts
• Procedure
  • Only register a lower-order count the first time it is seen in a backoff context
• Example: bigram model
  • “San Francisco” is a common bigram
  • But, we only count the unigram “Francisco” the first time we see the bigram “San Francisco” - we change its unigram probability
Kneser-Ney II

\[ f^* (b \mid a) = \frac{\max \{ t(\cdot, a, b) - d, 0 \} }{t(\cdot, a, \cdot)} \]

\[ t(\cdot, a, b) = |\{ w : \text{count}(w, a, b) > 0 \}| \]

\[ t(\cdot, a, \cdot) = |\{(w, w') : \text{count}(w, a, w') > 0\}| \]
Kneser-Ney II

\[ f^*(b \mid a) = \frac{\max\{t(\cdot, a, b) - d, 0\}}{t(\cdot, a, \cdot)} \]

\[ t(\cdot, a, b) = |\{w : \text{count}(w, a, b) > 0\}| \]

\[ t(\cdot, a, \cdot) = |\{(w, w') : \text{count}(w, a, w') > 0\}| \]

Max-order n-grams estimated normally!
Other Formulations

- $N$-gram class-based language models

$$p(w) = \prod_{i=1}^{\ell} p(c_i | c_{i-n+1}, \ldots, c_{i-1}) \times p(w_i | c_i)$$
Other Formulations

- $N$-gram class-based language models

\[ p(w) = \prod_{i=1}^{\ell} p(c_i \mid c_{i-n+1}, \ldots, c_{i-1}) \times p(w_i \mid c_i) \]

- Syntactic language models

- Generative syntactic models induce distributions over strings

\[ p(w) = \sum_{\tau: \text{yield(}\tau\text{)=}w} p(\tau, w) \]
Pauls & Klein (2012)

\[ p(\tau, w) = p(\tau) \times p(w | \tau) \]
Pauls & Klein (2012)

\[ p(\tau, w) = p(\tau) \times p(w | \tau) \]
Pauls & Klein (2012)

\[ p(\tau, w) = p(\tau) \times p(w | \tau) \]
Feature-based Models

- Rosenfeld (1996)
  - “Maximum entropy” language models
  - Replace independent parameters with a multinomial logit distribution
  - Encode domain-specific knowledge
  - Expressive, but expensive
Less Stupid Multinomials
Less Stupid Multinomials

Features of $w$

- Ends in \textit{-ing}?
- Contains a digit?
- Found in Gigaword?
- Contains a capital letter?
Less Stupid Multinomials

Parameters → Features of $w$

- Ends in -ing?
- Contains a digit?
- Found in Gigaword?
- Contains a capital letter?
Less Stupid Multinomials
Less Stupid Multinomials
Less Stupid Multinomials
Less Stupid Multinomials
Less Stupid Multinomials

No analytic solution! :( 

Tuesday, January 22, 13
Announcements

• First language-in-10 start next week
  • Tuesday, Jan 29: David - Latin
  • Thursday, Jan 31: Weston - Mandarin
• HW 1 will be posted Thursday after class