Word Prediction

Suppose you read the following sequence of words:
- Sue swallowed the large green ...

What word do you expect to see next?
- pill? frog?
- tree? car? mountain?
- the? that?

We might want to use syntactic or semantic knowledge to answer this question ...
Word Prediction

But suppose we just use probabilities:

- $P(\text{pill} \mid \text{sue, swallowed, the, large, green})$
- $P(\text{tree} \mid \text{sue, swallowed, the, large, green})$
- $P(\text{that} \mid \text{sue, swallowed, the, large, green})$
- ...

Generally:

$$P(w_i \mid w_1, w_2, \ldots, w_{i-1}) = \frac{P(w_1, w_2, \ldots, w_{i-1}, w_i)}{P(w_1, w_2, \ldots, w_{i-1})}$$

Chain Rule

$$P(w_1, w_2, \ldots, w_n)$$

$$= P(w_1) P(w_2 \mid w_1) P(w_3 \mid w_1, w_2) \ldots P(w_n \mid w_1, \ldots, w_{n-1})$$

$$= \prod_{i=1}^{n} P(w_i \mid w_1, \ldots, w_{i-1})$$

Note: Also add markers \texttt{<s>} and \texttt{/s>} at the beginning and end of the sentence.
Markov Assumption

\[ P(w_i \mid w_1, \ldots, w_{i-1}) \]

unigram:
\[ = P(w_i) \]

bigram:
\[ = P(w_i \mid w_{i-1}) \]

trigram:
\[ = P(w_i \mid w_{i-2}, w_{i-1}) \]

Example

\[ P(\text{sue, swallowed, the, large, green, pill}) = \]
\[ P(\text{sue} \mid <s>) \] * 
\[ P(\text{swallowed} \mid \text{sue}) \] * 
\[ P(\text{the} \mid \text{swallowed}) \] * 
\[ P(\text{large} \mid \text{the}) \] * 
\[ P(\text{green} \mid \text{large}) \] * 
\[ P(\text{pill} \mid \text{green}) \] * 
\[ P(</s> \mid \text{pill}) \]

using a bigram language model.
Estimating Probabilities

The Maximum Likelihood Estimate (MLE) is:

\[ P(w_j \mid w_{i-1}) = \frac{\text{count}(w_{i-1}w_j)}{\text{count}(w_{i-1})} \]

using a bigram language model.

\[ P(w_j \mid w_{i-2}, w_{i-1}) = \frac{\text{count}(w_{i-2}w_{i-1}w_j)}{\text{count}(w_{i-2}w_{i-1})} \]

\[ P(\text{pill} \mid \text{large}, \text{green}) = \frac{\text{count}(\text{large green pill})}{\text{count}(\text{large green})} \]

using a trigram language model.

Example

Suppose the text is:

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

\[
\begin{align*}
P(\text{I} \mid \text{a}) &= \frac{4}{8} = 0.5 \\
P(\text{Sam} \mid \text{a}) &= \frac{1}{5} = 0.2 \\
P(\text{am} \mid \text{I}) &= \frac{2}{3} = 0.67 \\
P(\text{am}) &= 0.5 \\
P(\text{do} \mid \text{I}) &= \frac{1}{3} = 0.33
\end{align*}
\]
What Should We Count?

From the Brown corpus:

- He stepped out into the hall, was delighted to encounter a water brother.

This sentence has 15 words counting “,” and “.” as words, 13 otherwise.

From the Switchboard corpus:

- I do uh main- mainly business data processing

Should we count the filler “uh”?

Should we count the fragment “main-”?

What Should We Count?

Tokens and Types:

- They picnicked by the pool, then lay back on the grass and looked at the stars.

This sentence has 16 tokens (not including punctuation), and 14 types.

Common notation for a corpus:

- $V =$ the size of the vocabulary, or the number of types.
- $N =$ the total number of running words, or the number of tokens.
What Should We Count?

Should we distinguish:
- “cats” and “cat”?
- “geese” and “goose”?

Terminology:
- **Lemma**: a set of lexical forms having the same stem, major part of speech, and rough word sense.
- **Wordform**: a fully inflected surface form.

For N-grams, we usually use wordforms.

Large Corpora

Brown et al. (1992):
- 583,000,000 wordform tokens.
- 293,181 wordform types.

- 1,024,908,267,229 tokens.
- 13,588,391 wordform types.

Why so many types?
- misspellings, numbers, names, acronyms, ...
Training and Testing

• Training Set (~80%):
  ◦ often includes separate “held out” data to estimate parameters of the model.

• Test Set, often split into:
  ◦ “development set” (~10%), to be used in the iterative process of developing an algorithm.
  ◦ “final test set” (~10%).

• Cardinal Rule:
  ◦ Don't train your model on the test set!

Chomsky's Argument

*Syntactic Structures* (1957):

Second, the notion “grammatical” cannot be identified with “meaningful” or “significant” in any semantic sense. Sentences (1) and (2) are equally nonsensical, but any speaker of English will recognize that only the former is grammatical.

(1) Colorless green ideas sleep furiously.
(2) Furiously sleep ideas green colorless.

...
Chomsky's Argument

*Syntactic Structures* (1957):

... Third, the notion “grammatical in English” cannot be identified in any way with the notion “high order of statistical approximation to English”. It is fair to assume that neither sentence (1) nor (2) (nor indeed any part of these sentences) has ever occurred in an English discourse. Hence, in any statistical model for grammaticalness, these sentences will be ruled out on identical grounds as equally ‘remote’ from English. Yet (1), though nonsensical, is grammatical, while (2) is not. ...

Problem: Sparse Data

Bigram counts for the Berkeley Restaurant Project corpus of 9332 sentences (J&M website):

<table>
<thead>
<tr>
<th></th>
<th>i</th>
<th>want</th>
<th>to</th>
<th>eat</th>
<th>chinese</th>
<th>food</th>
<th>lunch</th>
<th>spend</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>5</td>
<td>827</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>want</td>
<td>2</td>
<td>0</td>
<td>608</td>
<td>1</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>to</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>686</td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>211</td>
</tr>
<tr>
<td>eat</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>16</td>
<td>2</td>
<td>42</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>chinese</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>82</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>food</td>
<td>15</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>lunch</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>spend</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

What to do about the zero counts?
**Problem: Sparse Data**

Divide bigram counts by unigram counts:

<table>
<thead>
<tr>
<th></th>
<th>want</th>
<th>to</th>
<th>eat</th>
<th>chinese</th>
<th>food</th>
<th>lunch</th>
<th>spend</th>
</tr>
</thead>
<tbody>
<tr>
<td>2533</td>
<td>927</td>
<td>2417</td>
<td>746</td>
<td>158</td>
<td>1093</td>
<td>341</td>
<td>278</td>
</tr>
</tbody>
</table>

to compute bigram probabilities:

<table>
<thead>
<tr>
<th></th>
<th>i</th>
<th>want</th>
<th>to</th>
<th>eat</th>
<th>chinese</th>
<th>food</th>
<th>lunch</th>
<th>spend</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>0.002</td>
<td>0.33</td>
<td>0</td>
<td>0.0036</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00079</td>
</tr>
<tr>
<td>want</td>
<td>0.0022</td>
<td>0</td>
<td>0.66</td>
<td>0.0011</td>
<td>0.0065</td>
<td>0.0065</td>
<td>0.0054</td>
<td>0.0111</td>
</tr>
<tr>
<td>to</td>
<td>0.00083</td>
<td>0</td>
<td>0.0017</td>
<td>0.28</td>
<td>0.00083</td>
<td>0</td>
<td>0.0025</td>
<td>0.087</td>
</tr>
<tr>
<td>eat</td>
<td>0</td>
<td>0</td>
<td>0.0027</td>
<td>0</td>
<td>0.021</td>
<td>0.0027</td>
<td>0.056</td>
<td>0</td>
</tr>
<tr>
<td>chinese</td>
<td>0.0063</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.52</td>
<td>0.0063</td>
<td>0</td>
</tr>
<tr>
<td>food</td>
<td>0.014</td>
<td>0</td>
<td>0.014</td>
<td>0</td>
<td>0.00092</td>
<td>0.0037</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>lunch</td>
<td>0.0059</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0029</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>spend</td>
<td>0.0036</td>
<td>0</td>
<td>0</td>
<td>0.0036</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Zipf's Law**

<table>
<thead>
<tr>
<th>Word</th>
<th>Freq.</th>
<th>Rank</th>
<th>$f \cdot r$</th>
<th>Word</th>
<th>Freq.</th>
<th>Rank</th>
<th>$f \cdot r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>3332</td>
<td>1</td>
<td>3332</td>
<td>turned</td>
<td>51</td>
<td>200</td>
<td>10200</td>
</tr>
<tr>
<td>and</td>
<td>2972</td>
<td>2</td>
<td>5944</td>
<td>you'll</td>
<td>30</td>
<td>300</td>
<td>9000</td>
</tr>
<tr>
<td>a</td>
<td>1775</td>
<td>3</td>
<td>5235</td>
<td>name</td>
<td>21</td>
<td>400</td>
<td>8400</td>
</tr>
<tr>
<td>he</td>
<td>877</td>
<td>10</td>
<td>8770</td>
<td>comes</td>
<td>16</td>
<td>500</td>
<td>8000</td>
</tr>
<tr>
<td>but</td>
<td>410</td>
<td>20</td>
<td>8400</td>
<td>group</td>
<td>13</td>
<td>600</td>
<td>7800</td>
</tr>
<tr>
<td>be</td>
<td>294</td>
<td>30</td>
<td>8820</td>
<td>lead</td>
<td>11</td>
<td>700</td>
<td>7700</td>
</tr>
<tr>
<td>there</td>
<td>222</td>
<td>40</td>
<td>8880</td>
<td>friends</td>
<td>10</td>
<td>800</td>
<td>8000</td>
</tr>
<tr>
<td>one</td>
<td>172</td>
<td>50</td>
<td>8600</td>
<td>begin</td>
<td>9</td>
<td>900</td>
<td>8100</td>
</tr>
<tr>
<td>about</td>
<td>158</td>
<td>60</td>
<td>9480</td>
<td>family</td>
<td>8</td>
<td>1000</td>
<td>8000</td>
</tr>
<tr>
<td>more</td>
<td>138</td>
<td>70</td>
<td>9660</td>
<td>brushed</td>
<td>4</td>
<td>2000</td>
<td>8000</td>
</tr>
<tr>
<td>never</td>
<td>124</td>
<td>80</td>
<td>9920</td>
<td>sins</td>
<td>2</td>
<td>3000</td>
<td>6000</td>
</tr>
<tr>
<td>Oh</td>
<td>116</td>
<td>90</td>
<td>10440</td>
<td>Could</td>
<td>2</td>
<td>4000</td>
<td>8000</td>
</tr>
<tr>
<td>two</td>
<td>104</td>
<td>100</td>
<td>10400</td>
<td>Applausive</td>
<td>1</td>
<td>8000</td>
<td>8000</td>
</tr>
</tbody>
</table>

*Table 1.3* Empirical evaluation of Zipf's law on *Tom Sawyer.*
Zipf's Law

Brown Corpus: \( f \times r = k \)

Mandelbrot's Formula

Brown Corpus: \( f = P(r + \rho)^{-B} \)
Smoothing

Basic Idea:
- Decrease the probability of the previously seen events, so that there is some probability left over for the unseen events.

Several Methods:
- Laplace (or “add one”) smoothing.
- Good-Turing discounting.
- Interpolation.
- Katz backoff.

Laplace Smoothing

Laplace's Law (1814):
- Add “one” to all the counts!
- Thus, for bigrams:
  \[ P_{\text{Lap}}(w_i \mid w_{i-1}) = \frac{\text{count}(w_{i-1}, w_i) + 1}{\text{count}(w_{i-1}) + V} \]
- Alternatively, compute a “reconstructed count”:
  \[ e^*(w_{i-1}, w_i) = \frac{[\text{count}(w_{i-1}, w_i) + 1] \times \text{count}(w_{i-1})}{\text{count}(w_{i-1}) + V} \]
Laplace Smoothing

Smoothed bigram counts:

<table>
<thead>
<tr>
<th></th>
<th>i</th>
<th>want</th>
<th>to</th>
<th>eat</th>
<th>chinese</th>
<th>food</th>
<th>lunch</th>
<th>spend</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>6</td>
<td>828</td>
<td>1</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>want</td>
<td>3</td>
<td>1</td>
<td>609</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>to</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>687</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>212</td>
</tr>
<tr>
<td>eat</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>17</td>
<td>3</td>
<td>43</td>
<td>1</td>
</tr>
<tr>
<td>chinese</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>83</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>food</td>
<td>16</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>lunch</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>spend</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Laplace Smoothing

Smoothed bigram probabilities:

\[
P_{\text{Lap}}^*(w_i | w_{i-1}) = \frac{\text{count}(w_{i-1}w_i) + 1}{\text{count}(w_{i-1}) + V}
\]

<table>
<thead>
<tr>
<th></th>
<th>i</th>
<th>want</th>
<th>to</th>
<th>eat</th>
<th>chinese</th>
<th>food</th>
<th>lunch</th>
<th>spend</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>0.0015</td>
<td>0.21</td>
<td>0.00025</td>
<td>0.0025</td>
<td>0.00025</td>
<td>0.00025</td>
<td>0.00025</td>
<td>0.00075</td>
</tr>
<tr>
<td>want</td>
<td>0.0013</td>
<td>0.00042</td>
<td>0.0013</td>
<td>0.0018</td>
<td>0.00078</td>
<td>0.00026</td>
<td>0.0018</td>
<td>0.055</td>
</tr>
<tr>
<td>to</td>
<td>0.00078</td>
<td>0.00046</td>
<td>0.0014</td>
<td>0.00046</td>
<td>0.0078</td>
<td>0.0014</td>
<td>0.02</td>
<td>0.00046</td>
</tr>
<tr>
<td>eat</td>
<td>0.0012</td>
<td>0.00062</td>
<td>0.00062</td>
<td>0.00062</td>
<td>0.00062</td>
<td>0.052</td>
<td>0.0012</td>
<td>0.00062</td>
</tr>
<tr>
<td>chinese</td>
<td>0.0063</td>
<td>0.00039</td>
<td>0.0063</td>
<td>0.00039</td>
<td>0.0079</td>
<td>0.0002</td>
<td>0.00039</td>
<td>0.00039</td>
</tr>
<tr>
<td>food</td>
<td>0.0017</td>
<td>0.00056</td>
<td>0.00056</td>
<td>0.00056</td>
<td>0.00056</td>
<td>0.0011</td>
<td>0.00056</td>
<td>0.00056</td>
</tr>
<tr>
<td>lunch</td>
<td>0.0012</td>
<td>0.00058</td>
<td>0.0012</td>
<td>0.00058</td>
<td>0.00058</td>
<td>0.00058</td>
<td>0.00058</td>
<td>0.00058</td>
</tr>
</tbody>
</table>
Laplace Smoothing

Reconstructed counts:

\[ c^*(w_{i-1}w_i) = \frac{\text{count}(w_{i-1}w_i) + 1}{\text{count}(w_{i-1}) + V} \times \text{count}(w_{i-1}) \]

<table>
<thead>
<tr>
<th></th>
<th>i</th>
<th>want</th>
<th>to</th>
<th>eat</th>
<th>chinese</th>
<th>food</th>
<th>lunch</th>
<th>spend</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.8</td>
<td>527</td>
<td>0.64</td>
<td>6.4</td>
<td>0.64</td>
<td>0.64</td>
<td>1.9</td>
<td></td>
</tr>
<tr>
<td>want</td>
<td>1.2</td>
<td>0.39</td>
<td>238</td>
<td>0.78</td>
<td>2.7</td>
<td>2.7</td>
<td>2.3</td>
<td>0.78</td>
</tr>
<tr>
<td>to</td>
<td>1.9</td>
<td>0.63</td>
<td>3.1</td>
<td>430</td>
<td>1.9</td>
<td>0.63</td>
<td>4.4</td>
<td>133</td>
</tr>
<tr>
<td>eat</td>
<td>0.34</td>
<td>0.34</td>
<td>1</td>
<td>0.34</td>
<td>5.8</td>
<td>1</td>
<td>15</td>
<td>0.34</td>
</tr>
<tr>
<td>chinese</td>
<td>0.2</td>
<td>0.098</td>
<td>0.098</td>
<td>0.098</td>
<td>8.2</td>
<td>0.2</td>
<td>0.098</td>
<td></td>
</tr>
<tr>
<td>food</td>
<td>6.9</td>
<td>0.43</td>
<td>6.9</td>
<td>0.43</td>
<td>0.86</td>
<td>2.2</td>
<td>0.43</td>
<td>0.43</td>
</tr>
<tr>
<td>lunch</td>
<td>0.57</td>
<td>0.19</td>
<td>0.19</td>
<td>0.19</td>
<td>0.19</td>
<td>0.38</td>
<td>0.19</td>
<td>0.19</td>
</tr>
<tr>
<td>spend</td>
<td>0.32</td>
<td>0.16</td>
<td>0.32</td>
<td>0.16</td>
<td>0.16</td>
<td>0.16</td>
<td>0.16</td>
<td>0.16</td>
</tr>
</tbody>
</table>

For N-gram models, this technique transfers too much probability mass to unseen events:

- \( \text{count}(\text{want to}) \) is reduced from 608 to 238.
- \( P(\text{to} | \text{want}) \) is reduced from 0.66 to 0.26.

We can measure this reduction by the discount:

\[ d_c = \frac{c^*}{c} \]

- the discount for "want to" is 0.39.
- the discount for "Chinese food" is 0.10.

Result: Laplace smoothing does not work very well in empirical tests (Church and Gale, 1991a).
Good-Turing Discounting

Good (1953) attributes the idea to Alan Turing:

- Let $N_c$ be the number of N-grams that have occurred $c$ times (the “frequency of frequency” $c$).
- Re-estimate the smoothed counts as follows:
  \[ c^* = (c + 1) \frac{N_{c+1}}{N_c} \]
- Based on Good-Turing Theorem, assuming binomial distributions.
- We want to know the probability attributed to the N-grams that have not previously been seen, which we can think of as the “missing mass”.

Thus we can estimate the count of unseen N-grams using the count of N-grams we have seen once.
Fishing Example

Assume there are 8 species of fish in a lake, and we have caught 6 species with the following counts:

<table>
<thead>
<tr>
<th></th>
<th>carp</th>
<th>perch</th>
<th>whitefish</th>
<th>trout</th>
<th>salmon</th>
<th>eel</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>10</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The other two species (catfish and bass) have not yet been seen.

What is the probability that the next fish we catch will be one of the previously unseen species?

\[
P_{\text{GT}}(\text{catfish}) = \frac{N - n}{N} = \frac{8}{18} = .67
\]

\[
P_{\text{GT}}(\text{trout}) = \frac{n - 1}{N} = \frac{2}{17} = .12
\]

The probability that the next fish is a catfish is:

\[
P_{\text{GT}}(\text{catfish}) = .085
\]

Note: The count for trout was reduced from 1 to:

\[
c^*(\text{trout}) = .67
\]
Good-Turing Discounting

Bigram “frequencies of frequencies” and Good-Turing re-estimates for a 22 million word AP Newswire corpus (Church and Gale, 1991) and for the 9332 sentences in the Berkeley Restaurant corpus:

<table>
<thead>
<tr>
<th></th>
<th>AP Newswire</th>
<th>Berkeley Restaurant</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N_c$</td>
<td>$c^+$ (GT)</td>
</tr>
<tr>
<td>c (MLE)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>74,671,100,000</td>
<td>0.0000270</td>
</tr>
<tr>
<td>1</td>
<td>2,018,046</td>
<td>0.446</td>
</tr>
<tr>
<td>2</td>
<td>449,721</td>
<td>1.26</td>
</tr>
<tr>
<td>3</td>
<td>188,933</td>
<td>2.24</td>
</tr>
<tr>
<td>4</td>
<td>105,668</td>
<td>3.24</td>
</tr>
<tr>
<td>5</td>
<td>68,379</td>
<td>4.22</td>
</tr>
<tr>
<td>6</td>
<td>48,190</td>
<td>5.19</td>
</tr>
</tbody>
</table>

In practice:

- Since some values of $N_c$ could be zero, smooth these values before discounting, e.g., use a log-linear regression:
  \[
  \log(N_c) = a + b \log(c)
  \]

- Since large counts are reliable, don’t apply discounting when $c > k$ for some threshold $k$, e.g., when $c > 5$.
- Treat counts of 1 as if they were 0.
- Combine discounting with interpolation and backoff.
Interpolation

Combine trigrams, bigrams and unigrams:

\[
\hat{P}(w_i \mid w_{i-2}, w_{i-1}) = \lambda_1 P(w_i) + \lambda_2 P(w_i \mid w_{i-1}) + \lambda_3 P(w_i \mid w_{i-2}, w_{i-1})
\]

\(\lambda_i > 0, \sum_i \lambda_i = 1\)

Verify: \(\sum_w \hat{P}(w \mid w_{i-2}, w_{i-1}) = 1\)

Set \(\lambda\)s using “held out” corpus.

Katz Backoff

For trigrams:

\[
P_{KB}(w_i \mid w_{i-2}, w_{i-1}) = \begin{cases} 
  P^*(w_i \mid w_{i-2}, w_{i-1}), & \text{if} \ count(w_{i-2}w_{i-1}w_i) > 0 \\
  \alpha(w_{i-2}, w_{i-1}) P_{KB}(w_i \mid w_{i-1}), & \text{if} \ count(w_{i-2}w_{i-1}) > 0 \\
  P^*(w_i), & \text{otherwise}
\end{cases}
\]

For bigrams:

\[
P_{KB}(w_i \mid w_{i-1}) = \begin{cases} 
  P^*(w_i \mid w_{i-1}), & \text{if count}(w_{i-1}w_i) > 0 \\
  \alpha(w_{i-1}) P^*(w_i), & \text{otherwise}
\end{cases}
\]
Katz Backoff with Smoothing

Katz backoff bigram probabilities for eight words (out of V = 1446) using Good-Turing discounting with k = 5 and counts of 1 replaced with 0:

<table>
<thead>
<tr>
<th></th>
<th>i</th>
<th>want</th>
<th>to</th>
<th>eat</th>
<th>chinese</th>
<th>food</th>
<th>lunch</th>
<th>spend</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>0.0014</td>
<td>0.326</td>
<td>0.00248</td>
<td>0.00355</td>
<td>0.006205</td>
<td>0.0017</td>
<td>0.00073</td>
<td>0.000489</td>
</tr>
<tr>
<td>want</td>
<td>0.00134</td>
<td>0.00152</td>
<td>0.056</td>
<td>0.000485</td>
<td>0.00455</td>
<td>0.00455</td>
<td>0.00384</td>
<td>0.000485</td>
</tr>
<tr>
<td>to</td>
<td>0.000112</td>
<td>0.00152</td>
<td>0.00165</td>
<td>0.244</td>
<td>0.00812</td>
<td>0.0017</td>
<td>0.00175</td>
<td>0.0073</td>
</tr>
<tr>
<td>eat</td>
<td>0.00101</td>
<td>0.00152</td>
<td>0.00106</td>
<td>0.00189</td>
<td>0.0214</td>
<td>0.00160</td>
<td>0.0563</td>
<td>0.00085</td>
</tr>
<tr>
<td>chinese</td>
<td>0.000813</td>
<td>0.00152</td>
<td>0.00248</td>
<td>0.00189</td>
<td>0.006205</td>
<td>0.519</td>
<td>0.00283</td>
<td>0.00085</td>
</tr>
<tr>
<td>food</td>
<td>0.0137</td>
<td>0.00152</td>
<td>0.0137</td>
<td>0.00189</td>
<td>0.006409</td>
<td>0.00306</td>
<td>0.00073</td>
<td>0.00085</td>
</tr>
<tr>
<td>lunch</td>
<td>0.00363</td>
<td>0.00152</td>
<td>0.00248</td>
<td>0.00189</td>
<td>0.006205</td>
<td>0.00131</td>
<td>0.00073</td>
<td>0.00085</td>
</tr>
<tr>
<td>spend</td>
<td>0.00161</td>
<td>0.00152</td>
<td>0.00161</td>
<td>0.00189</td>
<td>0.006205</td>
<td>0.0017</td>
<td>0.00073</td>
<td>0.00085</td>
</tr>
</tbody>
</table>

Evaluating Language Models

Subjective Evaluation

- Use a technique attributed to Claude Shannon:

  - Generate texts randomly by sampling from the N-gram models, e.g., for Shakespeare and for the Wall Street Journal.
  - Do they look the same?
Shakespeare

To him swallowed confess hear both. Which. Of save on trail for are ny device and more life have
Every enter now severally so, let
Hill he late speaks, of! a more to leg less first you enter
Are where extent and sighs have rise excellency took of. Sleep knife we, near; vile like

Bigram
What means, sir. I confess she? then all sorts, he is trim, captain.
Why dost stand forth thy casque, forsooth; he is this palpable hit the King Henry. Live king. Follow.
What we, hath got so she that I rest and sent to scold and nature bankrupt, nor the first gentleman?

Trigram
Sweat prince, Falstaff shall die. Harry of Mornmouth’s grave.
This shall forbid it should be branded, if renown made it empty.
Indeed the duke; and had a very good friend.
Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, ‘tis done.

Quadrigram
King Henry: What! I will go seek the traitor Gloucester. Exempt some of the watch. A great banquet serv’d up; Will you not tell me who I am?
It cannot be but so.
Indeed the short and the long. Marry, ‘tis a noble Lepidas.

The quadrigram really looks like Shakespeare. Why?

Wall Street Journal

Months the any and issue of year foreign new exchange’s septemner were recession ex-
change new endorsed a acquire to six executives

Bigram
Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor
would seem to complete the major central planners one point five percent of U. S. E. has
already old M. X. corporation of living on information such as more frequently fishing to
keep her

Trigram
They also point to ninety nine point six billion dollars from two hundred four oh six three
percent of the rates of interest stores as Mexico and Brazil on market conditions

The Wall Street Journal is not Shakespeare.
Can we measure these differences objectively?
Perplexity

Given a test set \( W = w_1 w_2 \ldots w_N \), the perplexity is the probability of the test set, normalized by \( N \):

\[
PP(W) = P(w_1, w_2, \ldots, w_N)^{-1/N}
\]

For a bigram model, the perplexity is:

\[
PP(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}}
\]

Perplexity

Models: Unigram, Bigram, Trigram.
Training Set: 38 million word Wall Street Journal corpus with \( V = 19,979 \).
Method: Katz backoff with Good-Turing discounting.
Test Set: 1.5 million words.
Results:

<table>
<thead>
<tr>
<th></th>
<th>Unigram</th>
<th>Bigram</th>
<th>Trigram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perplexity</td>
<td>962</td>
<td>170</td>
<td>109</td>
</tr>
</tbody>
</table>