Lecture 5: Evaluating language models
How do we know whether one language model is better than another?

There are two ways to evaluate models:
- **intrinsic evaluation** captures how well the model captures what it is supposed to capture
- **extrinsic evaluation** captures how useful the model is in a particular task.

Both cases require an **evaluation metric** that allows us to measure and compare the performance of different models.
Intrinsic Evaluation: Perplexity
Perplexity

Perplexity is the inverse of the probability of the test set (as assigned by the language model), normalized by the number of word tokens in the test set.

Minimizing perplexity = maximizing probability!

Language model $\text{LM}_1$ is better than $\text{LM}_2$ if $\text{LM}_1$ assigns lower perplexity (= higher probability) to the test corpus $w_1 \ldots w_N$

NB: the perplexity of $\text{LM}_1$ and $\text{LM}_2$ can only be directly compared if both models use the same vocabulary.
Perplexity $PP(w_1 \ldots w_n)$

Given a test corpus with $N$ tokens, $w_1 \ldots w_N$, and an $n$-gram model $P(w_i \mid w_{i-1}, \ldots, w_{i-n+1})$ the perplexity $PP(w_1 \ldots w_N)$ is defined as follows:

$$PP(w_1 \ldots w_N) = P(w_1 \ldots w_N)^{-\frac{1}{N}}$$

$$= \sqrt[N]{\frac{1}{P(w_1 \ldots w_N)}}$$

$$= \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i \mid w_1 \ldots w_{i-1})}}$$

$$= \text{def} \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i \mid w_{i-1}, \ldots, w_{i-n+1})}}$$

(Chain rule)

(N-gram model)
Practical issues

Since language model probabilities are very small, multiplying them together often yields to underflow.

It is often better to use logarithms instead, so replace

$$PP(w_1 \ldots w_N) = \text{def} \sqrt[n]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1}, \ldots, w_{i-n+1})}}$$

with

$$PP(w_1 \ldots w_N) = \text{def} \exp \left( - \frac{1}{N} \sum_{i=1}^{N} \log P(w_i|w_{i-1}, \ldots, w_{i-n+1}) \right)$$
An experiment

Models:
   Unigram, Bigram, Trigram models
   (with Good-Turing smoothing)

Training data:
   38M words of WSJ text (Vocabulary: 20K types)

Test data:
   1.5M words of WSJ text

Results:

<table>
<thead>
<tr>
<th>Cloze Test</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>

Conclusion: The bigram is much better than the unigram, and the trigram is even better.
Extrinsic (task-based) Evaluation: Word Error Rate
Task-based evaluation

Perplexity doesn’t tell us how good the model is in an application (e.g. when we need to generate strings).

We sometimes need a task-based evaluation:

Train model A, plug it into your system.
Evaluate performance of system A on task.
Train model B, plug it in, evaluate system B on same task.
Compare scores of system A and system B
Language Models for speech recognition

Speech recognition:
Predict the most likely sequence of words $W = w_1...w_n$
from the acoustic signal $S$:
$$W^* = \arg\max_W P(W|S)$$

Word error rate (WER)
How much does the predicted sequence of words differ from
the actual sequence of words in the correct transcript of $S$?
$$\text{WER} = \frac{\text{Insertions} + \text{Deletions} + \text{Substitutions}}{\text{Actual words in transcript}}$$

Insertions: “eat lunch” $\rightarrow$ “eat a lunch”
Deletions: “see a movie” $\rightarrow$ “see movie”
Substitutions: “drink ice tea” $\rightarrow$ “drink nice tea”
To recap....
Today’s key concepts

Intrinsic evaluation of language models: Perplexity
Extrinsic (task-based) evaluation: word error rate

Today’s reading:
Jurafsky and Martin, Chapter 4: sec. 1-6, 9.1, 4.10