Language model unit:

- What a language model is
  - Analogy with human language models
  - Extrinsic evaluation
  - Intrinsic evaluation
- Statistics about words
  - *n*-grams
  - Counts, hapaxes, ranks
  - Zipf's law
- Basic language models
  - Training and testing
  - Maximum likelihood
  - Unigram, bigram, trigram
- Smoothing
  - What's wrong with maximum likelihood
  - Laplace smoothing
  - What's wrong with Laplace smoothing
- Good-Turing smoothing
  - Principle
  - Practice
- Linear interpolation
- Alternate approaches

Basic definitions and axioms of probability

- *Event* is a primitive term. Informally, events are the things whose probability we want to compute, estimate, or predict.
- The set of all events that we are modeling is the *event space*.
- For *language models*, the most common event is a word (type) occurring in a given context (being a token). Language models can also be used for predicting other linguistic events, like characters, word sequences, parts of speech, sentences, ...
- For event w, $P(w)$ is the probability of w. Moreover, $0 \leq P(w) \leq 1$
- If V is the event space, then $\sum_{w \in V} P(w) = 1$
- The *conditional probability* of event A in light of event B is

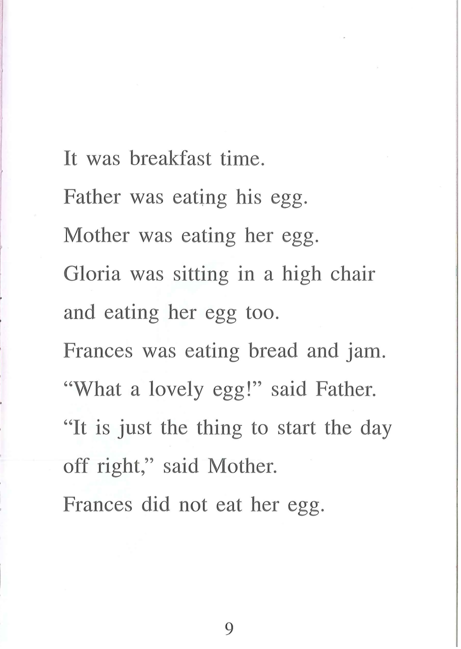
$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

- *Bayes's theorem* allows us to convert from one conditional probability to another

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Language technology	**Source**	**Channel**	**Observation**
Text decompression	Original text	Compressor	Compressed text
Sentiment analysis	Writer's sentiment	Writing process	Text whose sentiment is to be determined
Spelling correction	Correctly spelled word	Typing process	Possibly misspelled word
POS tagging	POS	Writing process	Word
Machine translation	Text in target language	Writing process	Text in original language

$$\arg\max_S P(S|O) = \arg\max_S \frac{P(O|S)P(S)}{P(O)} = \arg\max_S P(O|S)P(S)$$


It was breakfast time.
Father was eating his egg.
Mother was eating her egg.
Gloria was sitting in a high chair
and eating her egg too.
Frances was eating bread and jam.
“What a lovely egg!” said Father.
“It is just the thing to start the day
off right,” said Mother.
Frances did not eat her egg.

9

Russell Hoban and Lillian Hoban, *Bread and Jam for Frances*. I Can Read edition 2008; originally published 1964

$$
\begin{aligned}
&P(\texttt{Frances did not eat her egg})\\
&\quad= P(\texttt{egg}|\texttt{Frances did not eat her}) \cdot P(\texttt{Frances did not eat her})\\
&\quad= P(\texttt{egg}|\ldots) \cdot P(\texttt{her}|\texttt{Frances did not eat}) \cdot P(\texttt{Frances ...eat})
\end{aligned}
$$

$$P(w_{1:n}) = P(w_1)P(w_2|w_1)P(w_3|w_{1:2})\cdots P(w_n|w_{1:n-2})$$

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-1}) \text{ or } P(w_n|w_{n-2}w_{n-1})$$

Perplexity (and logs)

$$\begin{aligned}
P(w_0 \cdots w_{K-1})^{\frac{-1}{K}} &= \sqrt[K]{\frac{1}{P(w_0 \cdots w_{K-1})}} \\
= \left(\prod_{i=0}^{K-1} P(w_i \mid h)\right)^{\frac{-1}{K}} &= \sqrt[K]{\frac{1}{\prod_{i=0}^{K-1} P(w_i \mid h)}} \\
= 2^{\frac{-1}{K}\sum_{i=0}^{K-1} \lg P(w_i|h)}
\end{aligned}$$

$$\begin{aligned}
\lg xy &= \lg x + \lg y \\
\lg x^y &= y \ \lg x \\
2^{\lg x} &= x
\end{aligned}$$

Interpretation of perplexity:

> *We suspect that speech recognition people prefer to report on the larger non-logarithmic numbers given by perplexity mainly because it is much easier to impress funding bodies by saying that "we've managed to reduce perplexity from 950 to only 540" than by saying that "we've reduced cross entropy from 9.9 to 9.1 bits." However, perplexity does also have an intuitive reading: a perplexity of k means that you are as surprised on average as you would have been if you had had to guess between k equiprobable choices at each step.*
>
> *Manning and Schütze, Foundations of Statistical Natural Language Processing, pg 78.*

Summary so far (i.e., summary from last time)

- A language model is a probability function for linguistic events.
  Words: $P(w)$ Word sequences: $P(w_0 w_1)$, $P(W_{0:N})$ Words in context: $P(w|h)$
- Humans have a natural language model.
- How good is a language model? We can evaluate a language model
  - Extrinsically: Measure the performance of a tool that uses the language model (example—text decompressor).
  - Intrinsically: Compute the *perplexity* of a language model on a test text.
    - Perplexity is a measure of how "surprised" the language model is by the text text.
    - Perplexity is the inverse probability of the test text normalized (geometric mean) by the size of the text.

$$\sqrt[K]{\frac{1}{P(w_0 \cdots w_{K-1})}} = \left(\prod_{i=0}^{K-1} P(w_i \mid h)\right)^{\frac{-1}{K}} = 2^{\frac{-1}{K}\sum_{i=0}^{K-1} \lg P(w_i|h)}$$

- Language models are trained on textual data. By exploring textual data, we can observe
  - The most common words are function words.
  - The most common non-stopwords reveal subject matter and genre.
  - A type's frequency is proportional to the inverse of its rank (Zipf's law)

$$f \cdot r = C \qquad f \propto \frac{1}{r}$$

  - The frequency of various *n*-grams changes over time.

A **stationary** *process is one that does not change over time. This is clearly wrong for language: new expressions regularly enter the language while others die out. . . . Nevertheless, for a snapshot of text from a certain period, we can assume that the language is near enough to unchanging, and so this is an acceptable approximation to truth.*

Manning and Schütze, Foundations of Statistical Natural Language Processing, pg 76.

The assumption that the probability of a word depends only on the previous word is called a **Markov assumption***. Markov models are the class of probabilistic models that assume we can predict the probability of some future unit without looking too far into the past.*

Jurafsky and Martin, Speech and Language Processing 3e, §3.1

Let V be the vocabulary; we also use V to mean $|V|$ to reduce clutter.

Let N be the size of the training text and K be the size of the test text.

Let $C(w)$ be the number of tokens of type w in the training text; similarly define $C(w_1 w_2)$ etc.

Note that $\sum_{w \in V} C(w) = N$ and $\sum_{w in V} P(w) = 1$.

Maximum Likelihood Fallacy

"Which road leads to the Wicked Witch of the West?" asked Dorothy.

"There is no road," answered the Guardian of the Gates. "No one ever wishes to go that way."

"How, then, are we to find her?" inquired the girl.

"That will be easy," replied the man, "for when she knows you are in the country of the Winkies she will find you, and make you all her slaves."

"Perhaps not," said the Scarecrow, "for we mean to destroy her."

"Oh, that is different," said the Guardian of the Gates. "No one has ever destroyed her before, so I naturally thought she would make slaves of you, as she has of the rest. "

F Baum, The Wonderful Wizard of Oz

Perspectives on the maximum likelihood fallacy

- MLE is biased high for rare events and (infinitely) low for unseen events.
- Rare events are given too much probability mass, unseen events are given too little.
- Predicting the future is not the same thing as predicting a randomly chosen past event.
- If a word occurs once in training, what is most likely?
  - That word actually does occur about once every N tokens.
  - The word is actually more common than $\frac{1}{N}$, but got unlucky in training.
  - The word is actually less common than $\frac{1}{N}$, but was lucky to be in the training set at all.

Good-Turing smoothing

Let r range over frequencies; $r = C(w)$ for some $w \in V$.

Let n_r be the number of types that occur exactly r times in the training text. n_r is the *frequency of frequency* r.

$$n_r = |\{w \in V \mid C(w) = r\}|$$

Thus n_1 is the number of hapaxes, and n_0 is the number of unseen words.

$$\sum_{r=0}^{\infty} n_r = V$$

$$\sum_{r=0}^{\infty} (r \cdot n_r) = N$$

Axiom 1. Good-Turing smoothing is good.

Axiom 2. Frequency (as frequency rank) vs frequency of frequency follows Zipf's law.

Theorem. Laplace smoothing is bad. (Gale and Church, 1994)

Proof. *Suppose Laplace smoothing is good. Then, by Axiom 1, its formula would reduce to the formula for Good-Turing, that is,*

$$\begin{aligned}
\frac{(r+1)n_{r+1}}{Nn_r} &= \frac{r+1}{N+V} && \textit{by equating Lapace and GT probabilities} \\
& && \textit{for some type with count } r \\
\frac{n_{r+1}}{n_r} &= \frac{N}{N+V} \\
n_{r+1} &= \frac{N}{N+V} n_r \\
n_r &= \frac{N}{N+V} n_{r-1} && \textit{by change of variable} \\
n_r &= \left(\frac{N}{N+V}\right)^r n_0
\end{aligned}$$

By Axiom 2, Zipf's law predicts $n_r = \frac{c}{r}$ *for some* c*. Contradiction.* $\square$

Good-Turing with Katz's k cut off:

$$P_{GT-Katz}(w) = \begin{cases} \frac{n_1}{N n_0} & \text{if } C(w) = 0 & \text{(unseen words)} \\[2em] \dfrac{(r+1)\frac{n_{r+1}}{n_r} - r\frac{(k+1)\cdot n_{k+1}}{n_1}}{N\left(1 - \frac{(k+1)\cdot n_{k+1}}{n_1}\right)} & \text{if } 1 \leq r = C(w) \leq k & \text{(rare words)} \\[2em] \frac{C(w)}{N} & \text{otherwise} & \text{(common words)} \end{cases}$$

Theorem. If each constituent language model P_j is a proper language model and $\sum_{j=0}^{k-1} \lambda_j = 1$, then P_{LI} is a proper language model.

Proof. *Suppose all that. Then*

$$\begin{aligned}
\sum_{w \in V} P_{LI}(w) &= \sum_{w \in V} \sum_{j=0}^{k-1} \lambda_j P_j(w) \\
&= \sum_{j=0}^{k-1} \sum_{w \in V} \lambda_j P_j(w) \\
&= \sum_{j=0}^{k-1} \lambda_j \sum_{w \in V} P_j(w) \\
&= \sum_{j=0}^{k-1} \lambda_j = 1 \qquad \square
\end{aligned}$$

Coming up:

- Huffman encoding assignment (Wed, Sept 17)
- Reading from J&M, Sections 3.(0-8) (Mon, Sept 15) At least 3.(0& 1) by duedate; you may spread out the rest
- Language model quiz (Tues, Sept 23) (not posted yet)
- Language model programming assignment (Fri, Sept 26)
- Reading from Larson, *The Myth of Artificial Intelligence* (Fri, Sept 19)
- Reading from J&M, Sections 8.(0–4) (Fri, Sept 22)