Language models

- **Why?** – to estimate prior probability of occurence of a word
- use redundancy of language
- **How to do that?**
  - use grammar rules – not very practical
  - stochastical ansatz – build LM *empirically* from training corpus used here only

example:

Spoken	No LM	Trigram LM
we	we	we
expect	expects	expect
to	to	to
make	make	make
investments	investments	investments
of	abel	of
a		
larger	larger	larger
nature	nature	nature
in	in	in
the		the
industrial	industrial	industrial
area	area	area

1. m-gram language models

- w_n is the n-th word in a sequence $w_1 w_2 \dots w_n$
- that means $w_1 \dots w_{n-1}$ is (*history* or *context*) of the word w_n
- conditional probability $P(w_n|w_1 \dots w_{n-1})$

for m-gram language model:

$$P(w_n|w_1 \dots w_{n-1}) \approx P(w_n|w_{n-m+1} \dots w_{n-1})$$

examples:

- $m = 1$ – unigram-model

$$P(w_n|w_1 \dots w_{n-1}) \approx P(w_n)$$

- $m = 2$ – bigram-model

$$P(w_n|w_1 \dots w_{n-1}) \approx P(w_n|w_{n-1})$$

- $m = 3$ – trigram-modell

$$P(w_n|w_1 \dots w_{n-1}) \approx P(w_n|w_{n-2} w_{n-1}).$$

probability P of word sequence $w_1 \dots w_N$:

$$P(w_1 \dots w_N) = P(w_1) P(w_2|w_1) \cdots P(w_N|w_1 \dots w_{N-1}).$$

approximation for $P(w_1 \dots w_N)$:

$$\begin{aligned}
P(w_1 \dots w_N) \approx P_{1G}(w_1 \dots w_N) &:= \prod_{n=1}^{N} P(w_n), \\
P(w_1 \dots w_N) \approx P_{2G}(w_1 \dots w_N) &:= P(w_1) \prod_{n=2}^{N} P(w_n|w_{n-1}), \\
P(w_1 \dots w_N) \approx P_{2G}(w_1 \dots w_N) &:= P(w_1) P(w_2|w_1) \prod_{n=3}^{N} P(w_n|w_{n-2} w_{n-1})
\end{aligned} \quad .$$

- W number of *different* words in the LM
- $\Rightarrow$ number of m tupel for m-gram-LM: W^m

2. perplexity – a measure for quality of the LM

perplexity PP of a language model for a certain test corpus:

$$PP := [P(w_1 \dots w_N)]^{-1/N}$$

probability of the sequence $w_1 \dots w_N$ (test corpus) given a (m-gram-) LM
$\Rightarrow$ with history h_n of w_n:

$$PP = \left[\prod_{n=1}^{N} P(w_n|h_n)\right]^{-1/N}$$

perplexity = average number of possible words to choose between per word position

building a LM – minimizing of the perplexity

- LM is built *empircally* from a training corpus $w_1 \dots w_N$
- search for all different sequences with m words (history h and word w)
- $N(h, w)$ number of occurrences of the string "hw"
- number of occurrences of history h:

$$N(h) = \sum_w N(h, w)$$

conditional probability $P(w|h)$; equal to relative frequencies:

$$P(w|h) = \frac{N(h, w)}{N(h)}$$

this is equivalent to *minimizing the perplexity on training corpus*
i.e., (maximum likelihood estimation) maximize the log likelihood function:

$$\frac{1}{N}\sum_{i=1}^{N} \log P(w_i|h_i) = \sum_{h,w} \frac{N(h, w)}{N} \log P(w|h) = -\log PP$$

with $\sum_w P(w|h) = 1$ for all histories h

3. Smoothing

Problem: big training corpus, e.g., $N = 10^7$ words and $W = 10^4$ different words
bigram model: $W^2 = 10^8$ possible pairs
trigram model $W^3 = 10^{12}$ possible tripel
$\Rightarrow$ estimation of probabilities with relative frequences can result in zero;
$\Rightarrow PP \to \infty$
$\Rightarrow$ probabilities have to be *smoothed*

- N_k – frequencies of events (occurence of a sequence "hw")
- K – number of *possible* events (eg., W^3 for trigram)
- T – number of actual *measured* events in training corpus
- η_r – iteriert frequencies = number of events k with the same frequnecies $N_k = r$

$$\sum_{r=0}^{\infty} \eta_r r = T$$

p_k – probability for event k:

$$p_k = \frac{N_k^*}{\sum_{k=1}^{K} N_k^*}$$

with N_k^* – different function (so that $p_k = 0$ inpossible)
simple example: Jeffrey smootihing $N_k^* = N_k + 1$

Good-Turing smoothing:

$$p_k = \frac{r^*}{T} \quad \text{mit } r^* = (r+1)\frac{\eta_{r+1}}{\eta_r} \qquad \forall k: \; N_k = r$$

it follows:

$$\sum_{j|N_j=r} p_j = \eta_r \frac{r^*}{T} = \eta_{r+1} \frac{r+1}{T}$$

(and with ML estimation we would get $\eta_r r/T$)

$$\sum_{r=0}^{\infty} \eta_{r+1}(r+1)/T = \sum_{s=1}^{\infty} \eta_s s/T = T/T = 1$$

4. Other methods

use probabilities of LMs with lower order

$$\tilde{P}(w|h) = \begin{cases} P(w|h) & \text{wenn } N(h,w) > 0 \\ \beta(h)\tilde{P}(w|h') & \text{wenn } N(h,w) = 0 \end{cases}$$

with shorted history h' and smoothed probability $P(w|h)$

$$\beta(h) = \frac{\sum_w \{P(w|h)|N(h,w) = 0\}}{\sum_w \{\tilde{P}(w|h')|N(h,w) = 0\}}$$

which fulfills $\sum_w \tilde{P}(w|h) = 1$.

Interpolation

$N(h,w) > 0$ but small $\Rightarrow$ relative frequency is poor approximation for probability
lineare interpolation:

$$\tilde{P}(w|h) = \lambda_2 P(w|h) + \lambda_1 P(w) + \lambda_0 1/W$$

maximum entropy LMs

only $P(w)$ is know for a training corpus
want to know $P(w_1 w_2)$
no unique solution exists
$\Rightarrow$ What is the best guess for $P(w_1 w_2)$?
we know

$$\sum_{w_1} P(w_1 w_2) = P(w_2), \quad \sum_{w_2} P(w_1 w_2) = P(w_1)$$

$\Rightarrow 2W$ equation for W^2 values of $P(w_1 w_2)$
need more equations
$\Rightarrow$ maximum entropy method
maximize the entropy

$$H = -\sum_{w_1 w_2} P(w_1 w_2) \log P(w_1 w_2)$$

results in most probable $P(w_1 w_2)$ without biases