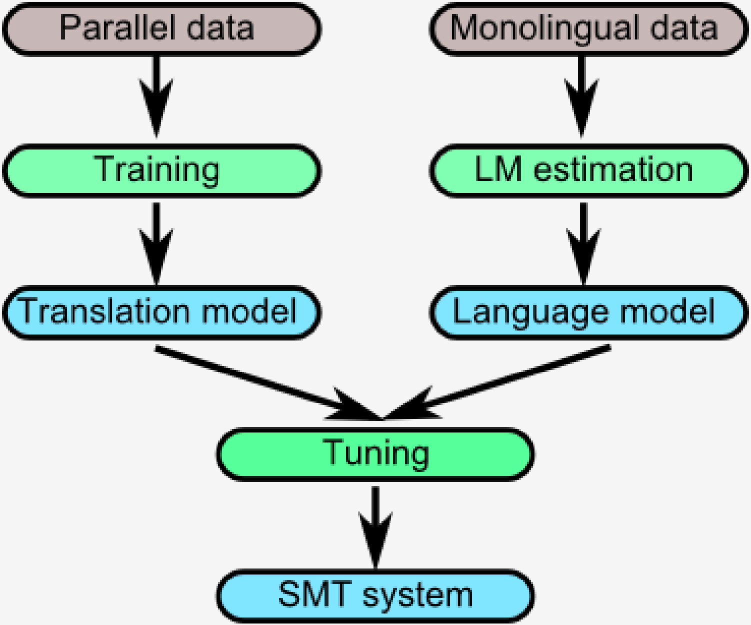
Statistical Machine Translation

-language models (cont.)-

Artem Sokolov
Computerlinguistik
Universität Heidelberg
Sommersemester 2015

material from P. Koehn, R. Shapire, D. McAllester, Chen & Goodman


Parallel data
Monolingual data
Training
LM estimation
Translation model
Language model
Tuning
SMT system

Why smmothing?

- the simplest way to estimate LM – MLE
- is problematic on sparse data:
  - problem with zero counts
  - $\Rightarrow$ zero probabilities for unseen n-grams
  - high perplexity on test
  - $\Rightarrow$low performance of the model as a whole

Laplace Smoothing: Add α

$$p = \frac{c + \alpha}{n + \alpha v}, \qquad \alpha < 1, \alpha \text{ optimized on held-out set}$$

$$r^* = \frac{T_r^1 + T_r^2}{N_r^1 + N_r^2}$$

N_r – # of n-grams that occur r times

$$r^* = (r+1)\frac{N_{r+1}}{N_r}$$

N_r – # of n-grams that occur r times

Jelinek-Mercer smoothing: Recursive Interpolation

$$p_n^I(w_i|w_{i-n+1},...,w_{i-1}) = \lambda_{w_{i-n+1},...,w_{i-1}} \; p_n(w_i|w_{i-n+1},...,w_{i-1}) \\ + (1 - \lambda_{w_{i-n+1},...,w_{i-1}}) \; p_{n-1}^I(w_i|w_{i-n+2},...,w_{i-1})$$

$$p_n^{BO}(w_i|w_{i-n+1},...,w_{i-1}) = \begin{cases} d_n(w_{i-n+1},...,w_{i-1})\; p_n(w_i|w_{i-n+1},...,w_{i-1}) \\ \qquad\qquad \text{if count}_n(w_{i-n+1},...,w_i) > k \\ \alpha_n(w_i|w_{i-n+1},...,w_{i-1})\; p_{n-1}^{BO}(w_i|w_{i-n+2},...,w_{i-1}) \\ \qquad\qquad \text{otherwise} \end{cases}$$

What is the probability of observing w "in the wild", if we saw it with some frequency in the corpus?

- $p(w)$
- sample S of size $|S|$ from $p(w)$
- $c(w)$ = the number of times w occurs in S
- $S_r = \{w \; : \; c(w) = r\}$
- $M_r = \sum_{w \in S_r} p(w)$

M_r is a useful quantity:

- If we know it, $p(w)$ for $w \in S_r$ is $\frac{M_r}{|S_r|}$
  (total mass divided by total number of distinct elements)

Example: *To see the need for a smoothing, imagine we have sampled a large S where each n-gram occurs exactly once (quite unlikely event). The naive way of estimating M_1 would be*

$$\frac{\text{\# of times } w \text{ occurs in } S \;\times\; \text{\# of different words we are ok with}}{\text{total size of } S} = \frac{k \times |S_1|}{|S|} = \frac{1 \times |S|}{|S|} = 1.$$

However, for any reasonable distribution $p(w)$ the probability M_1, given such an unlikely sample S, should be close to 0 .

Find the expectation of M_r:

$$\mathbb{E}[M_r] = \sum_w p(w)P[w \in S_r] \quad = \frac{r+1}{|S|-r}\mathbb{E}[|S_{r+1}|] - \frac{r+1}{|S|-r}\mathbb{E}[M_{r+1}]$$

Almost unbiased estimate of M_r

$$\frac{r+1}{|S|-r}\mathbb{E}[|S_{r+1}|] \simeq \frac{r+1}{|S|}|S_{r+1}|.$$

Final formula

$$r^* = (r+1)\frac{|S_{r+1}|}{|S_r|}$$

"spite" , "constant"

both occur 993 times in the Europarl

- 9 words follow "spite"; almost always followed by "of" (979 times)
- 415 words follow "constant"; "and" (42), "concern" (27), "pressure" (26) and singletons (268)

Much more likely to see a new bigram that starts with "constant" than with "spite".

WB-smoothing is an instance of the recursive interpolation:

$$p_{WB}(w_i|w_{i-n+1}^{i-1}) = \lambda_{w_{i-n+1}^{i-1}} p_{ML}(w^i|w_{i-n+1}^{i-1}) + (1 - \lambda_{w_{i-n+1}^{i-1}}) p_{WB}(w_i|w_{i-n+2}^{i-1})$$

Intuition:

- in back-offs, we back-off to lower-order is higher-order is missing
- interpret $(1 - \lambda_{w_{i-n+1}^{i-1}})$ as the probability of recurring to the lower-order model
- use the number of unique words that follow the history to estimate this likeliness

- define the number of possible extensions of a history $w_1, ..., w_{n-1}$:

$$N_{1+}(w_1, ..., w_{n-1}, \bullet) = |\{w_n : c(w_1, ..., w_{n-1}, w_n) > 0\}|$$

- Define

$$1 - \lambda_{w_1,...,w_{n-1}} = \frac{N_{1+}(w_1, ..., w_{n-1}, \bullet)}{N_{1+}(w_1, ..., w_{n-1}, \bullet) + \sum_{w_n} c(w_1, ..., w_{n-1}, w_n)}$$

$$1 - \lambda_{\textit{spite}} = \frac{N_{1+}(\textsf{spite}, \bullet)}{N_{1+}(\textsf{spite}, \bullet) + \sum_{w_n} c(\textsf{spite}, w_n)}$$

$$= \frac{9}{9 + 993} = 0.00898$$

$$1 - \lambda_{\textit{constant}} = \frac{N_{1+}(\textsf{constant}, \bullet)}{N_{1+}(\textsf{constant}, \bullet) + \sum_{w_n} c(\textsf{constant}, w_n)}$$

$$= \frac{415}{415 + 993} = 0.29474$$

Observation:
Discount value $1 - d_r$ in the GT smoothing are often "almost constant" (for $r \gg 1$).

Idea:
Jelinek-Mercer interpolation with $\lambda_{w_{i-n+1}^{i-1}} p(w_i | w_{i-n+1}^{i-1})$ set to $\frac{\max\{c(w_{i-n+1}^{i}) - D, 0\}}{\sum_{w_i} c(w_{i-n+1}^{i})}$.

Final formula

$$\hat{p}(w_i | w_{i-n+1}^{i-1}) = \frac{\max(c(w_{i-n+1}^{i}) - D, 0)}{c(w_{i-n+1}^{i})} + \frac{D N_{1+}(w_{i-n+1}^{i-1} \bullet)}{\sum_{w_i} c(w_{i-n+1}^{i})} \hat{p}(w_i | w_{i-n+2}^{i-1})$$

Consider the word "York" (477 times). As frequent as the words "foods", "indicates" or "providers".

In a unigram LM, will have a respectable probability.

However, it almost always directly follows "New" (473 times).

Problem

- unigram model is used, if the bigram model is inconclusive.
- "York" is unlikely to be the second word in an unseen bigram
- therefore "York" **should have a low probability**.

Idea:
set the unigram probability to the number of different words that it follows instead of number of occurrences
Formalize:

$$\frac{c(w_i)}{\sum_{w_i} c(w_i)} = \sum_{w_{i-1}} \hat{p}(w_{i-1}w_i) = \sum_{w_{i-1}} \hat{p}(w_i|w_{i-1})p(w_{i-1})$$
$$= \sum_{w_{i-1}} \hat{p}(w_i|w_{i-1}) \frac{c(w_{i-1})}{\sum_{w_{i-1}} c(w_{i-1})}.$$

For absolute discounting we had:

$$\hat{p}(w_i|w_{i-n+1}^{i-1}) = \frac{\max(c(w_{i-n+1}^{i}) - D, 0)}{c(w_{i-n+1}^{i})} + \frac{DN_{1+}(w_{i-n+1}^{i-1}\bullet)}{\sum_{w_i} c(w_{i-n+1}^{i})}\hat{p}(w_i|w_{i-n+2}^{i-1})$$

Substitute into the constraint:

$$c(w_i) = c(w_i) - N_{1+}(\bullet w_i)D + D\hat{p}(w_i)N_{1+}(\bullet\bullet),$$

where

$$\begin{aligned} N_{1+}(\bullet w_i) &= |\{w_{i-1} : c(w_{i-1}w_i) > 0\}|, \\ N_{1+}(\bullet\bullet) &= |\{(w_{i-1}, w_i) : c(w_{i-1}w_i) > 0\}|. \end{aligned}$$

$$\hat{p}(w_i) = \frac{N_{1+}(\bullet w_i)}{N_{1+}(\bullet\bullet)}.$$

Idea

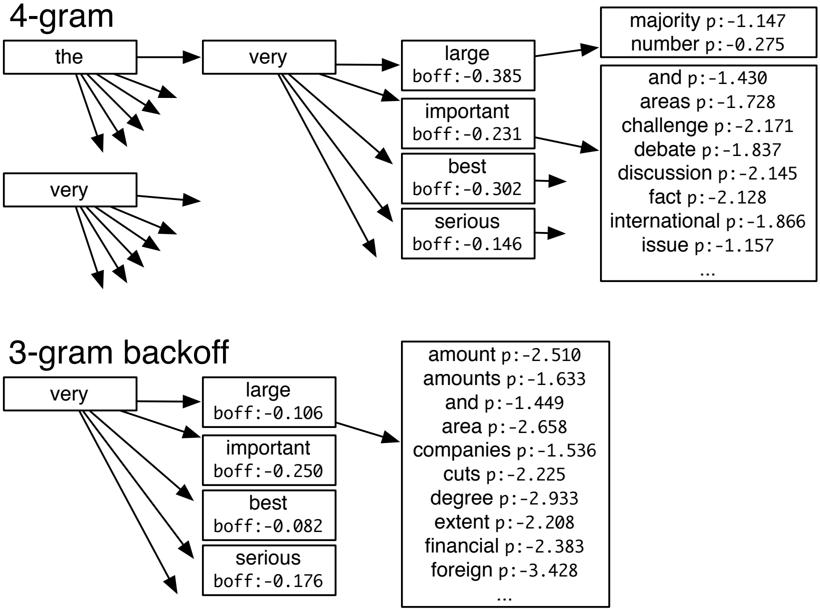
Use 3 discount factors D_1, D_2, D_{3+}

$$D = D(c) = D_1 \mathbf{1}[c = 1] + D_2 \mathbf{1}[c = 2] + D_{3+} \mathbf{1}[c > 2].$$

Perplexity for language models trained on the Europarl corpus:

Smoothing method	**bigram**	**trigram**	**4-gram**
Good-Turing	96.2	62.9	59.9
Witten-Bell	97.1	63.8	60.4
Modified Kneser-Ney	95.4	61.6	58.6
Interpolated Modified Kneser-Ney	94.5	59.3	54.0

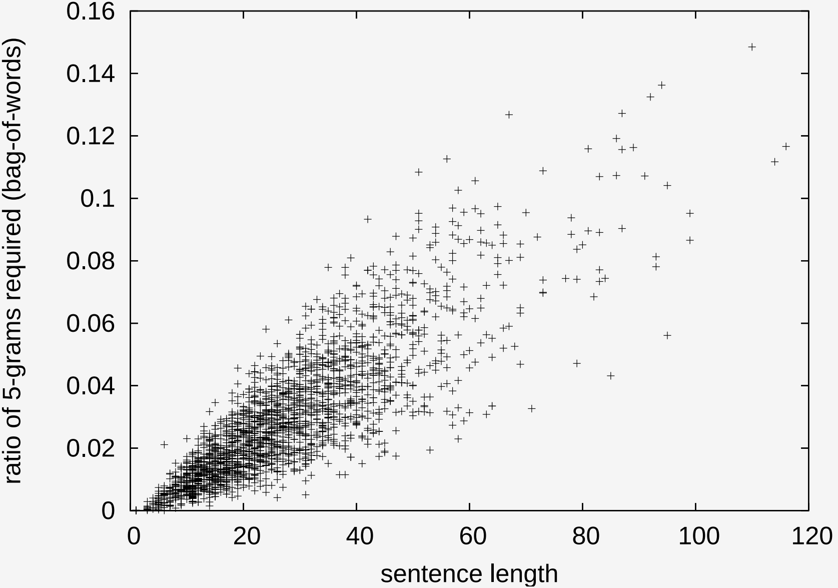
- estimation on disk
- effcient structures (trie)
  - ➡ ‘the very large majority’
  - ➡ ‘the very large number’
  - ➡ shared history


4-gram
the
very
very
large
boff:-0.385
important
boff:-0.231
best
boff:-0.302
serious
boff:-0.146
majority p:-1.147
number p:-0.275
and p:-1.430
areas p:-1.728
challenge p:-2.171
debate p:-1.837
discussion p:-2.145
fact p:-2.128
international p:-1.866
issue p:-1.157
...
3-gram backoff
very
large
boff:-0.106
important
boff:-0.250
best
boff:-0.082
serious
boff:-0.176
amount p:-2.510
amounts p:-1.633
and p:-1.449
area p:-2.658
companies p:-1.536
cuts p:-2.225
degree p:-2.933
extent p:-2.208
financial p:-2.383
foreign p:-3.428
...

Backoff from 4-gram to 3-gram:

$$\begin{aligned} p_{\text{LM}}(\text{amount}|\text{the very large}) =& \text{backoff}(\text{the very large}) \\ & \cdot p_3(\text{amount}|\text{very large}) \\ =& \exp(-0.385 + -2.510) \end{aligned}$$

- estimation on disk
- effcient structures (trie)
  - ‘the very large majority’
  - ‘the very large number’
  - shared history
- fewer bits to store numbers (num. indexes/huffman)
- bin probabilities
- reduce vocabulary (dates/numbers)
- filtering irrelevant n-grams


0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
20
40
60
80
100
120
ratio of 5-grams required (bag-of-words)
sentence length