Chapter 5:
Backing-Off Language Models

5.1. Unknown words and unseen unigrams

2

Goal of this section

- How to define a vocabulary for language tasks
- Notion of “out-of-vocabulary” words

3

Example „Crime and Punishment“

- Training corpus: 11 000 tokens
- Test corpus:
  - 3000 tokens
  - 108 of them not seen in training corpus

- Usually training corpus defines vocabulary
- Unseen words in test corpus are hence not part of the vocabulary

OOV-words: “out-of-vocabulary” words (tokens)

4

Example „Crime and Punishment“

- Training corpus: 11 000 tokens
- Test corpus:
  - 3000 tokens
  - 108 of them not seen in training corpus

Definition: $$\text{OOV-rate} = \frac{\#\,\text{unseen tokens in test corpus}}{\#\,\text{of tokens in test corpus}}$$

5

OOV-Rate on "Crime and Punishment"

- OOV-Rate = 108/3000 = 0.036 = 3.6%

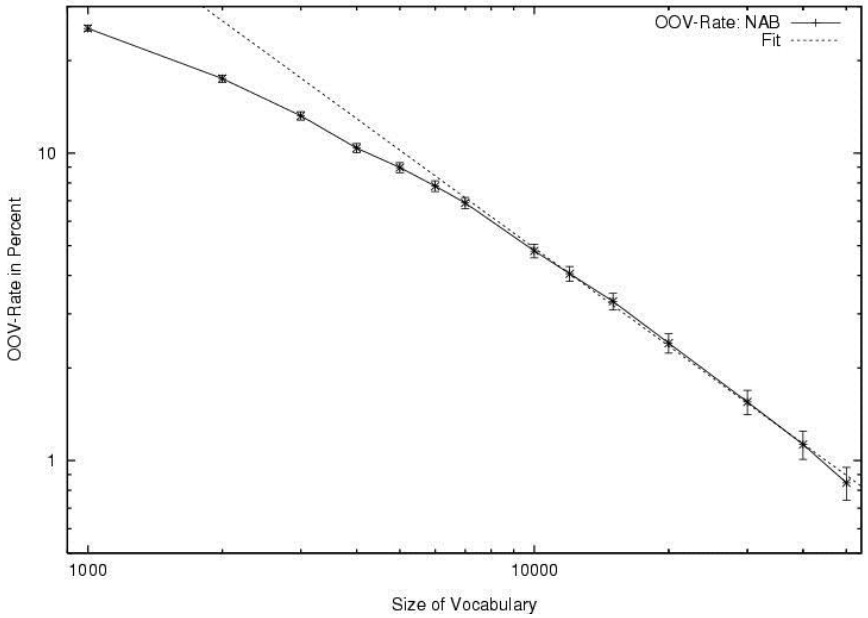
6

How to define a vocabulary

- Count the frequency of each word in a corpus
- Use the most frequent N words for the vocabulary

7

OOV-Rate vs. Vocabulary



- OOV rate depends on vocabulary size
- Morphologically rich languages have higher OOV rates for the same vocabulary size
- Specifically the relation is a power law

$$\text{OOV - Rate} \propto \frac{1}{(\text{Size of Vocabulary})^{\alpha}}$$

8

Corresponding Problem with sequences of words (M-Gram)

- Size of vocabulary: N
- Number of sequences of length M with N words:

$$N^M$$

- Numerical example: N=64000 and M=3
  - $N^M = 3\ 10^{14}$
  - No realistic possibility to observe all trigrams in a training corpus

$\mapsto$ need methods to account for unseen events

9

5.2 Maximum Likelihood Estimate of Probabilities

10

Goal of this section

- Simple way to determine a language model P(w|h)

11

From Perplexity to Likelihood

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

$$= \exp\left(-\sum_{w,h} f(w,h) \log\left(P(w \mid h)\right) \right)$$

$$-\log(PP) = \sum_{w,h} f(w,h) \log\left(P(w \mid h)\right) = \frac{1}{N} \sum_{w,h} N(w,h) \log\left(P(w \mid h)\right)$$

Define likelihood $F = \sum_{w,h} N(w,h) \log\left(P(w \mid h)\right)$

Minimize perplexity $\Leftrightarrow$ Maximize likelihood

12

Maximum-Likelihood Estimation of Probabilities

Maximize

$$F = \sum_{w,h} N(w,h) \log\left(P(w \mid h)\right)$$

Using the normalization constraint

$$\sum_{w} P(w \mid h) = 1 \quad \text{for all h}$$

Modified likelihood using Lagrange multiplier

$$F' = \sum_{w,h} N(w,h) \log\left(P(w \mid h)\right) + \sum_{h} \lambda(h) \left[1 - \sum_{w} P(w \mid h)\right]$$

13

Maximum-Likelihood Estimation of Probabilities (cont.)

$$F' = \sum_{w,h} N(w,h)\log\left(P(w|h)\right) + \sum_{h} \lambda(h)\left[1 - \sum_{w} P(w|h)\right]$$

$$\frac{\partial F'}{\partial P(w'|h')} = \frac{N(w',h')}{P(w'|h')} - \lambda(h') = 0 \qquad \Rightarrow P(w'|h') = \frac{N(w',h')}{\lambda(h')}$$

Constraint

$$1 = \sum_{w} P(w|h) = \sum_{w'} \frac{N(w',h')}{\lambda(h')} = \frac{N(h')}{\lambda(h')} \qquad \Rightarrow \lambda(h') = N(h')$$

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

Maximum likelihood estimate results in relative frequencies

Issues with Maximum Likelihood Estimate

- Unseen events

$\mapsto$ vanishing probability estimate

$\mapsto$ this sequence cannot be found in speech recognition

$\mapsto$ IR: this document does not cover all query terms, it cannot be relevant

Need other modeling scheme

15

5.3 Absolute discounting

(Kneser-Ney-Smoothing Part 1)

16

Goal of this section

- Discuss smoothing techniques that avoid vanishing probabilities

17

Backing-off language model

$$P(w \mid h) = \begin{cases} \dfrac{N(w,h)}{N(h)} + \\ 0 \end{cases}$$

für $N(w,h) > 0$

Backing-off weight $\alpha(h)$
Backing-off distribution $\beta(w|h)$ (normalized!)
Discounting parameter d

18

Backing-off language model (without animation)

$$P(w \mid h) = \begin{cases} \dfrac{N(w,h) - d}{N(h)} + \alpha(h)\ \beta(w \mid h) & \text{für } N(w,h) > 0 \\ \alpha(h)\ \beta(w \mid h) \end{cases}$$

Backing-off weight α(h)
Backing-off distribution β(w|h) (normalized!)
Discounting parameter d

19

Calculating the backing-off weight α(h)

Use normalization

$$1 = \sum_w P(w|h) = \sum_{w:N(w,h)>0} \frac{N(w,h)-d}{N(h)} + \sum_w \alpha(h)\ \beta(w|h)$$

$$= 1 - \frac{\sum_{w:N(w,h)>0} d}{N(h)} + \alpha(h)$$

Solve for α(h)

$$\alpha(h) = \frac{\sum_{w:N(w,h)>0} d}{N(h)} = \frac{d \sum_{w:N(w,h)>0} 1}{N(h)} = \frac{d\,R(h)}{N(h)} \qquad \text{with } R(h) = \sum_{w:N(w,h)>0} 1$$

20

Calculating the backing-off weight $\alpha(h)$

Backing-off weight

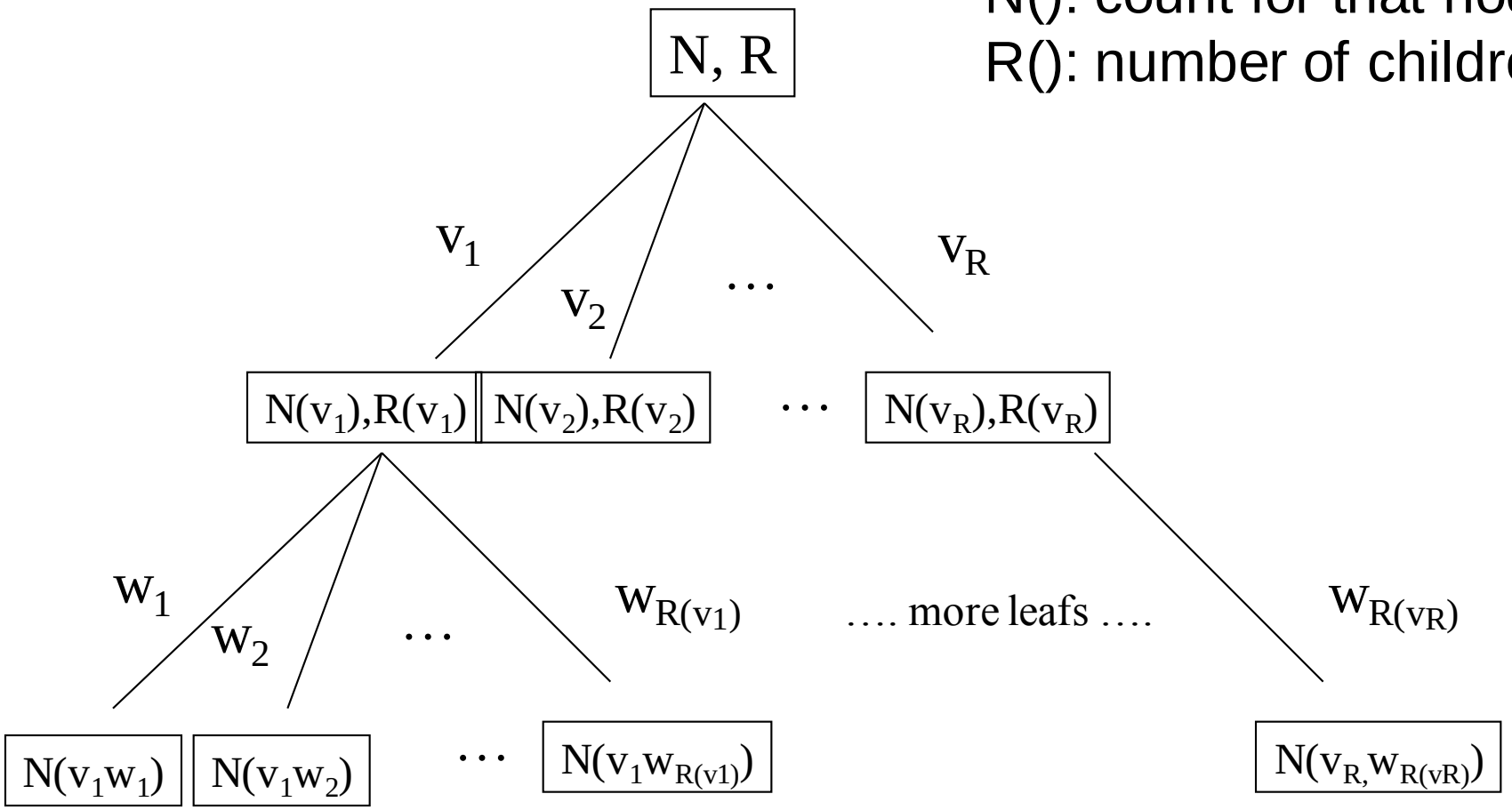
$$\alpha(\mathrm{h}) = \frac{d\,R(h)}{N(h)}$$

with $R(h) = \sum_{w:N(w,h)>0} 1$

21

Storing the counts (Example: Bigram v,w)

N(): count for that node
R(): number of children



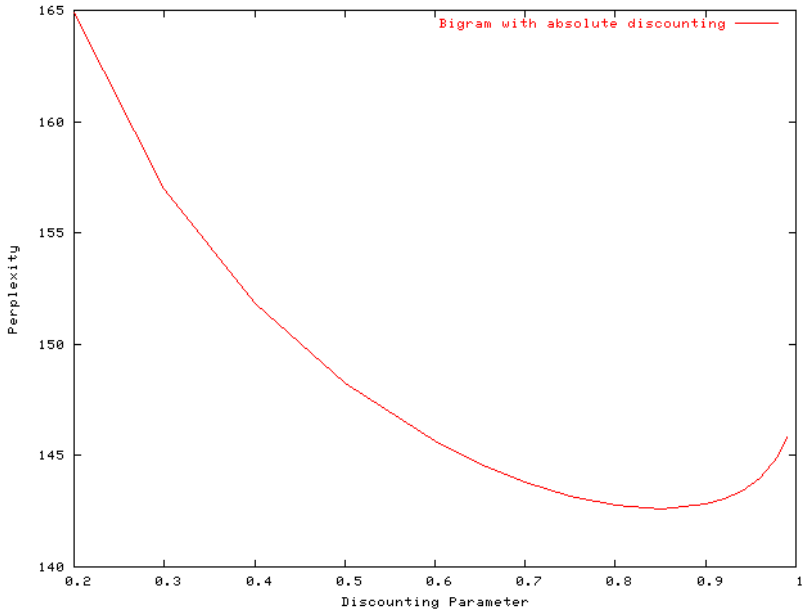
22

Examples

- Example of backing-off code
- Example of language model

23

Influence of Discounting Parameter



- Discounting parameter has a significant impact on perplexity
- Values of between 0.7 and 0.9 are typically good choices

24

5.4 Leaving-one-out estimate of the discounting parameters

25

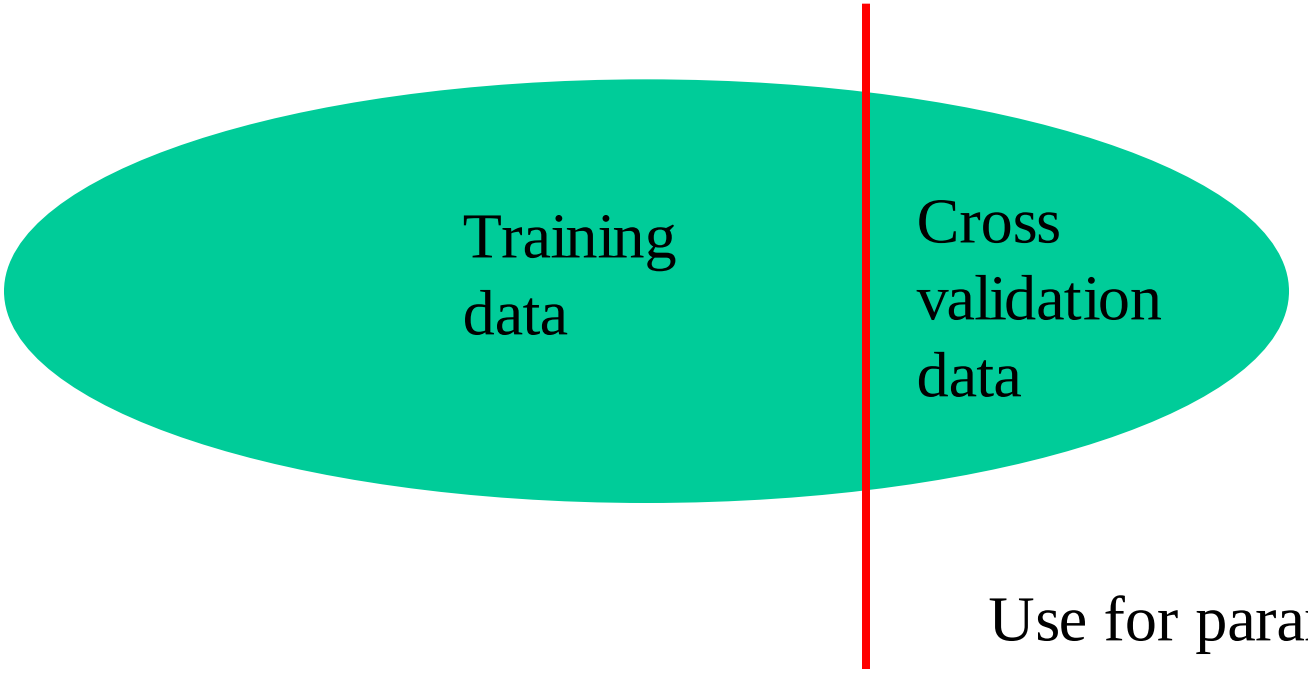
Goal

- How to get a good estimate for the discounting parameter without tuning it explicitly?

26

Closed form optimization solution for discounting parameter

- Idea of cross validation: split training data



27

Variants of cross validation

- Suppose you want to train a unigram language model:
  - What is the smallest possible cross validation corpus?
  - How do you avoid fluctuations?

28

Leaving-one-out

- Each unigram of the corpus can be the cross validation corpus
- Calculate average of all possible cross validation corpora

29

Change of counts for leaving one out

- $w_1\ w_2\ w_3\ w_4\ w_5\ \cancel{w_6}\ w_7\ w_8\ w_9\ \ldots\ w_N$

$$N(w_6) \quad \mapsto \quad N(w_6)\text{-}1$$

$$N \quad \mapsto \quad N\text{-}1$$

30

Leaving-one-out likelihood

$$F_{LOO} = \sum_{n=1}^{N} \log P_{LOO}(w_n) = \sum_{w \in W} N(w) \log P_{LOO}(w)$$

$$= \sum_{w \in W : N(w) > 1} N(w) \log \left[\frac{N(w) - 1 - d}{N - 1} + \frac{Rd}{N - 1} \beta \right]$$

$$+ \sum_{w \in W : N(w) = 1} \log \left[\frac{(R - 1)d}{N - 1} \beta \right]$$

Calculate $\frac{\partial F_{LOO}}{\partial d} = 0$

31

Maximizing Leaving-one-out Likelihood

Calculate $\frac{\partial F_{LOO}}{\partial d} = 0$

- Skip calculating the derivative
- Result

$$0 = \sum_{r=2}^{\infty} n_r \, r \frac{R\beta - 1}{r - 1 - d + R\beta d} + \frac{n_1}{d}$$

with

n_1 : number of words seen once (singeltons)

n_r : number of words that have a count of r (count-of-counts)

$\mapsto$ no closed form solution to this euqation

32

Iterative solution

Split off terms in n_1 and n_2

Solve for d and use as an iterative equation

$$d_{i+1} = \frac{n_1}{(1-R\beta)\left(n_1 + 2n_2 + \sum_{r=3}^{\infty} \frac{n_r r(1-d_i(1-R\beta))}{r-1-d_i(1-R\beta)}\right)}$$

33

Approximate solution

Just use first iteration and initialize with $d_0=0$

$$d_1 = \frac{n_1}{(1-R\beta)\left(n_1 + 2n_2 + \sum_{r=3}^{\infty} \frac{n_r r}{r-1}\right)}$$

34

Approximate solution

Usually $R\beta<1$

Ignore the sum (only a small change to discounting parameter)

$$d \approx \frac{n_1}{n_1 + 2n_2}$$

Works very well for all practical applications

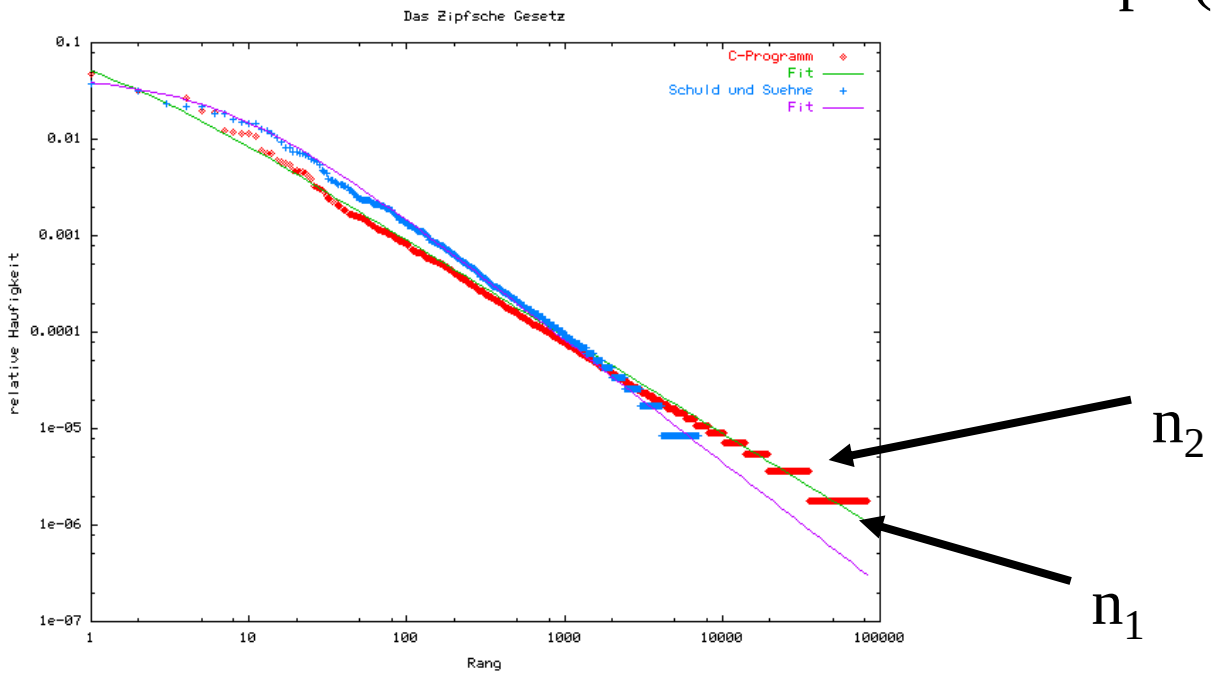
35

Demonstrate Approximate Solution

- -> Test formula using switchboard bigram

36

Remember Zipf´s Law



Zipf (simplified):

$$N(r) = \frac{N_0}{r^{\gamma}}$$

37

Estimate n_1 and n_2

- Length of the first step in the Zipf-Distribution

$$N(r) = \frac{N_0}{r^{\gamma}} \qquad \Rightarrow \qquad r = \left(\frac{N_0}{N}\right)^{\frac{1}{\gamma}}$$

$$\Rightarrow \qquad n_1 = \left(\frac{N_0}{1/2}\right)^{\frac{1}{\gamma}} - \left(\frac{N_0}{3/2}\right)^{\frac{1}{\gamma}}$$

$$\text{and} \qquad n_2 = \left(\frac{N_0}{3/2}\right)^{\frac{1}{\gamma}} - \left(\frac{N_0}{5/2}\right)^{\frac{1}{\gamma}}$$

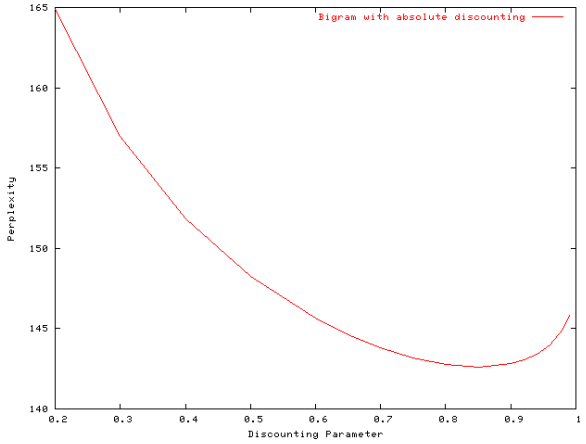
38

Estimate Discounting Parameter

$$d \approx \frac{n_1}{n_1 + 2n_2} = \frac{1}{1 + 2\frac{n_2}{n_1}}$$

$$= \frac{1}{1 + 2\frac{((2/3)^{1/\gamma} - (2/5)^{1/\gamma})}{(2^{1/\gamma} - (2/3)^{1/\gamma})}}$$

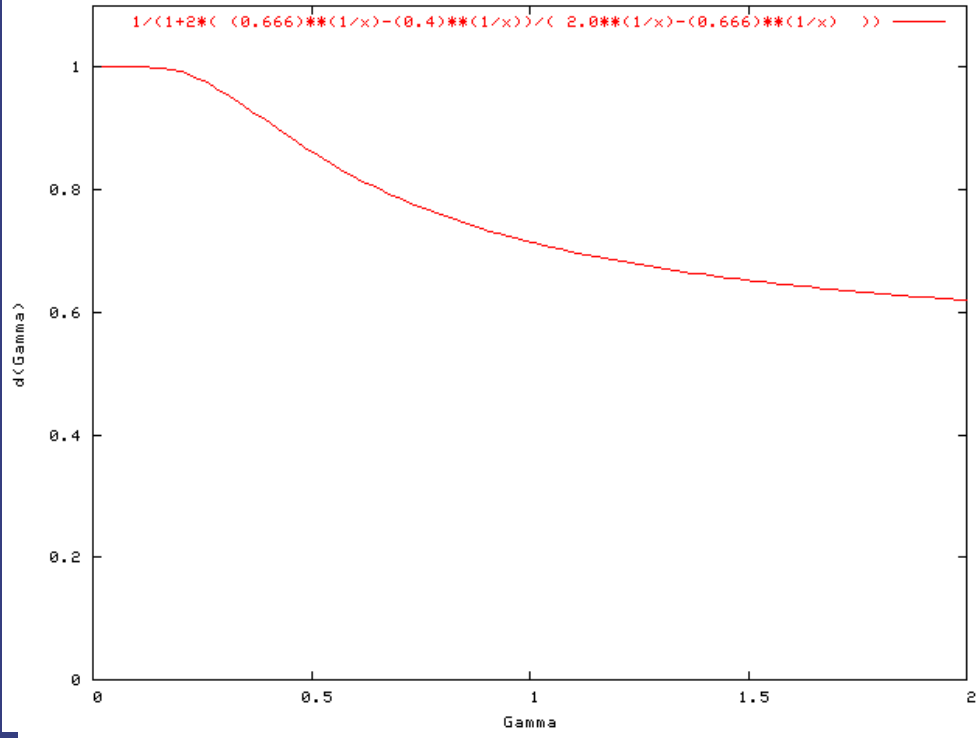
For $\gamma = 1$: $d \approx \frac{5}{7} \approx 0.71$



Estimate only gives degradation as compared to optimal numerical solution

39

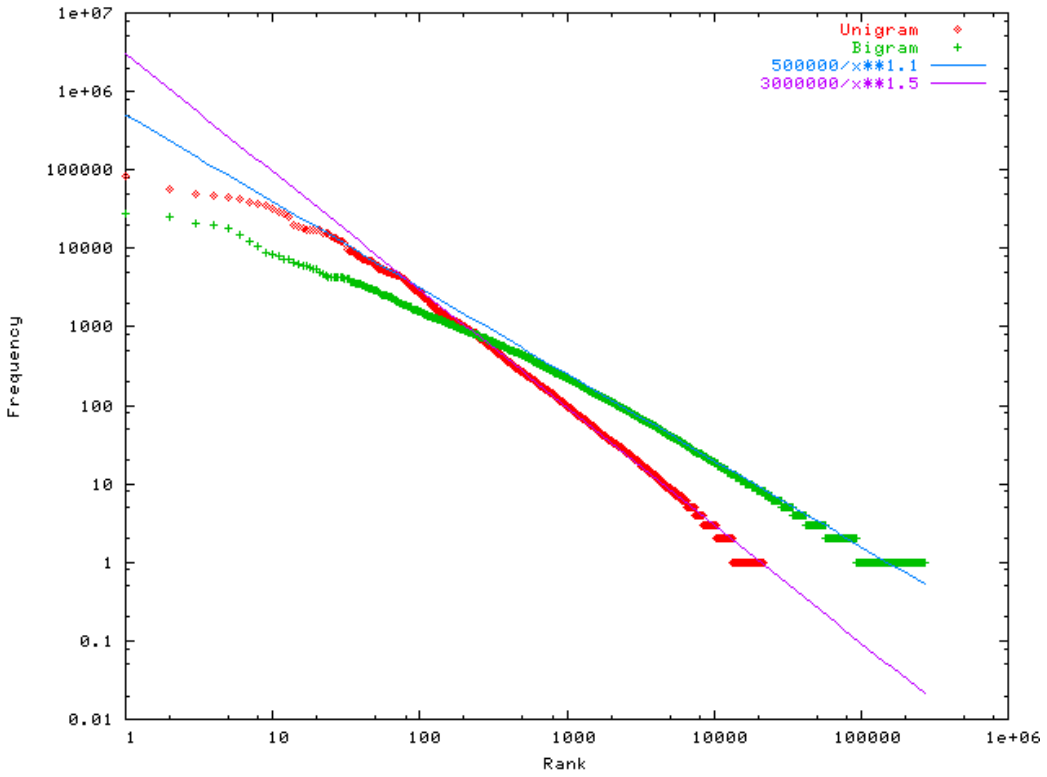
Slope in Zipf vs. Discounting



The larger the slope, the smaller the discounting parameter

40

Zipf for Unigram and Bigram



Longer range models have larger discounting parameters

41

5.5 Other smoothing methods

42

Goal

- Smoothing techniques that are simpler
- Smoothing techniques that are used for other reasons

43

Floor discounting ("add epsilon")

aka additive smoothing; Lidstone smoothing, Laplace Smoothing, …

Add a little bit to each count

$$P(w \mid h) = \frac{1}{Z}\left(N(w,h) + \varepsilon\right)$$

1/Z: Normalisation

$$1 = \sum_{w=1}^{V} P(w \mid h) = \frac{1}{Z}\left(N(h) + \varepsilon V\right) \quad \Rightarrow \quad Z = \left(N(h) + \varepsilon V\right)$$

$$P(w \mid h) = \frac{N(w,h) + \varepsilon}{N(h) + \varepsilon V}$$

44

Linear Discounting

$$P(w|h) = (1-\varepsilon)\frac{N(w,h)}{N(h)} + \varepsilon\beta(w|h)$$

ε: determines amount of smoothing

$\beta(w|h)$: "backing-off distribution"

Variant: $$P(w|h) = (1-\varepsilon(h))\frac{N(w,h)}{N(h)} + \varepsilon(h)\beta(w|h)$$

Popular value: $$\varepsilon(h) = \frac{n_1(h)}{N(h)} \quad \text{with } n_1(h) = \sum_{w:N(w,h)=1} 1$$

See example

45

Good-Turing Discounting

Adjusted counts:

$$N^*(w,h) = \begin{cases} [N(w,h)+1]\dfrac{n_{N(w,h)+1}}{n_{N(w,h)}} & \text{für } N(w,h) \le 5 \\ N(w,h) & \text{sonst} \end{cases}$$

Model:

$$P(w \mid h) = \frac{N^*(w,h)}{N(h)} + \lambda(h)\beta(w \mid h)$$

Backing-off weight from normalization: $\lambda(h) = 1 - \dfrac{N^*(h)}{N(h)}$

46

5.6 Kneser-Ney Smoothing

47

Kneser-Ney Smoothing

- Try to find a dedicated backing-off distribution
- Two different variants
  - “Marginal constraint” backing off
  - Singleton backing-off

48

Idea

$$P(w \mid h) = \begin{cases} \dfrac{N(w,h) - d}{N(h)} + \alpha(h)\ \beta(w \mid \hat{h}) & \text{für } N(w,h) > 0 \\ \alpha(h)\ \beta(w \mid \hat{h}) \end{cases}$$

With $\hat{h}$ beeing the history shortened by one word

$$p(w \mid \hat{h}) = \sum_{h'} P(w \mid h) P(h' \mid \hat{h}) \qquad (**)$$

with h' beeing the word removed from the history

$\mapsto$ look for backing off distribution such that (**) is satisfied

49

Solution

- Direct calculation
- Solution

$$\beta(w \mid \hat{h}) = \frac{N_+(w, \hat{h})}{\sum_w N_+(w, \hat{h})}$$

with

$$N_+(w, \hat{h}) = \sum_{h': N(w, \hat{h}, h') > 0} 1$$

Marginal constraint backing-off

50

Alternative Determine Backing-Off Distribution using Leaving-One-Out

$$F = F_0 + \sum_{h'\hat{h}w:N(w\hat{h}h')=1} \log\left(\alpha(h)\beta(w|\hat{h})\right)$$

$$+ \sum_{\hat{h}} \lambda(\hat{h})\left(1 - \sum_{w} \beta(w|\hat{h})\right)$$

F_0 likelihood for events seen more than once

$\lambda(\hat{h})$ Lagange multiplier

51

Solution for Leaving-One-Out Approach to Backing-Off Distribution

- Direct calculation of first derivative
- Solution

$$\beta(w \mid \hat{h}) = \frac{N_1(w, \hat{h})}{\sum_w N_1(w, \hat{h})}$$

with

$$N_1(w, \hat{h}) = \sum_{h': N(w, \hat{h}, h')=1} 1$$

52

Comparison Marginal-Constraint/Singleton Backing-Off

Backing-off Model	Bigram	Trigram	4-gram
Rel. Frequencies	160.1	146.4	147.8
Marginal Constraint	154.8	138.3	137.9
Singleton BO	156.2	142.6	141.9

53

5.7 Pruning of count trees

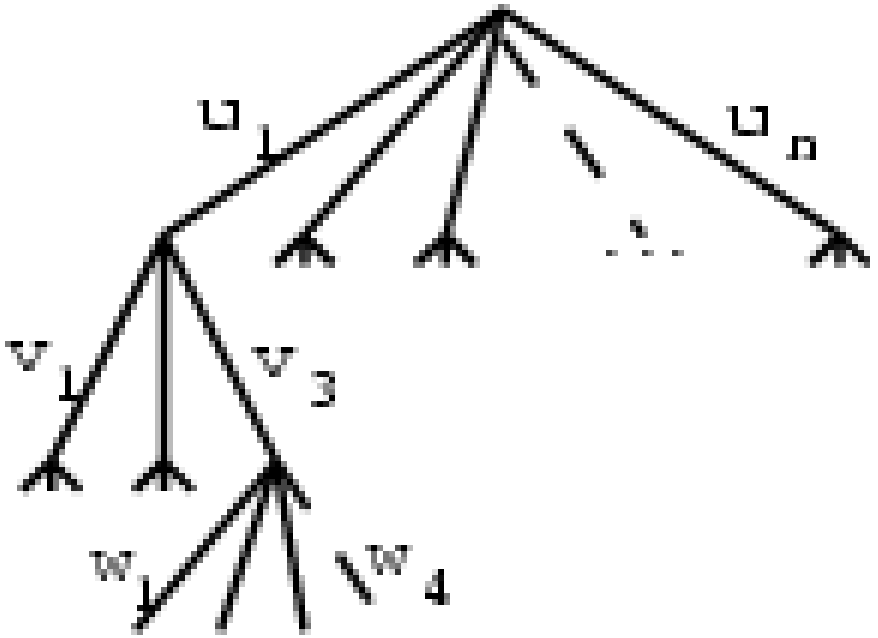
54

Goal

- Method to decrease size of the tree

55

Pruning Count-Trees



Level 1

Level 2

Level 3

Simple criterion for pruning trees:
remove all infrequent nodes

56

Improved Criterion: Change in Likelihood

- Metric inspired by maximum likelihood objective
- In case a leaf is pruned, the backing off probability will be used instead

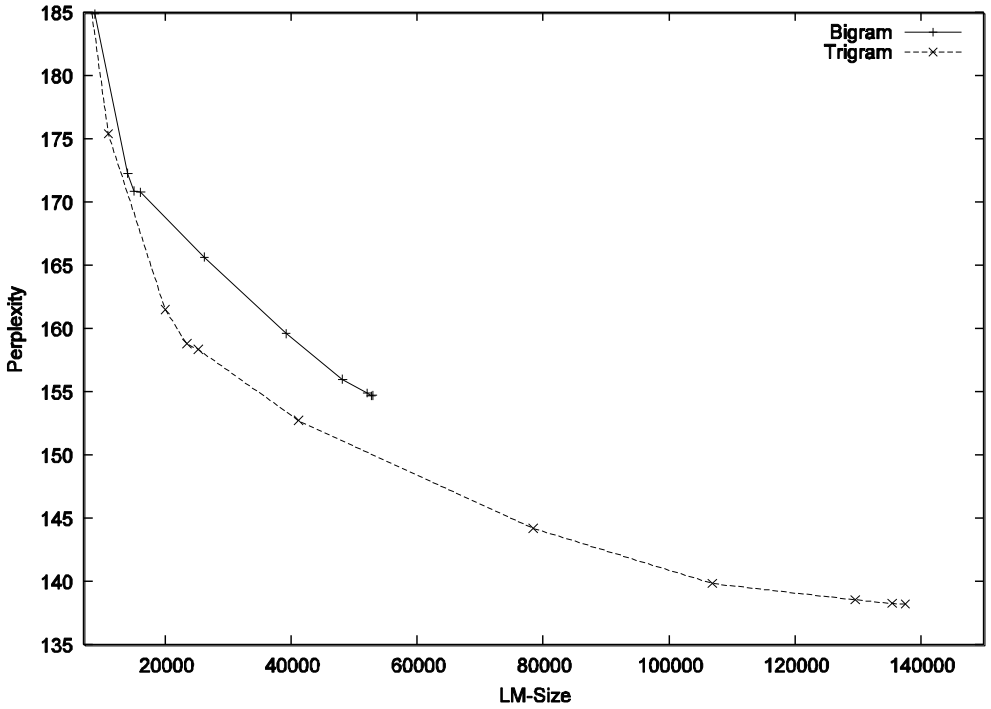
$$\Delta F_{Leaf} = N(w,h)\log P(w|h) - N(wh)\log\left(\alpha(h)\beta(w|\hat{h})\right)$$

$$= N(w,h)\log\left(\frac{P(w|h)}{\alpha(h)\beta(w|\hat{h})}\right)$$

- This is a metric for the leaves of the tree
- Create measure for sub-trees from measures for leaves

57

Effect of language model pruning



- Pruned Trigram better than pruned bigram
- For a given size pruning gives you the best LM
- Results generalize to longer N-grams

58

Summary

- OOV-words
- Maximum likelihood estimator
- Backing-off language models
- Discounting parameter
- Kneser-Ney Smoothing
- Pruning of trees

59