N-gram Language Models

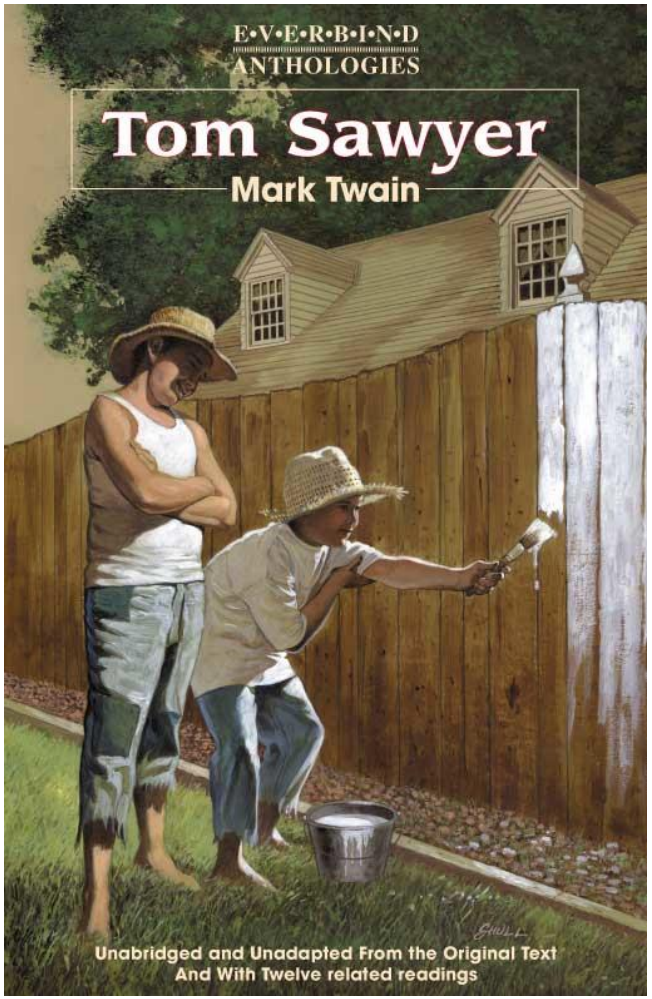
CMSC 723 / LING 723 / INST 725

MARINE CARPUAT
marine@cs.umd.edu

Today

- Counting words
  - Corpora, types, tokens
  - Zipf's law
- N-gram language models
  - Markov assumption
  - Sparsity
  - Smoothing

Let's pick up a book...



How many words are there?

- Size: ~0.5 MB
- Tokens: 71,370
- Types: 8,018
- Average frequency of a word: # tokens / # types = 8.9
  - But averages lie….

Some key terms…

- Corpus (pl. corpora)

- Number of word types vs. word tokens
  - Types: distinct words in the corpus
  - Tokens: total number of running words

What are the most frequent words?

Word	Freq.	Use
the	3332	determiner (article)
and	2972	conjunction
a	1775	determiner
to	1725	preposition, verbal infinitive marker
of	1440	preposition
was	1161	auxiliary verb
it	1027	(personal/expletive) pronoun
in	906	preposition

from Manning and Shütze

And the distribution of frequencies?

Word Freq.	Freq. of Freq.
1	3993
2	1292
3	664
4	410
5	243
6	199
7	172
8	131
9	82
10	91
11-50	540
50-100	99
> 100	102

from Manning and Shütze

Zipf's Law

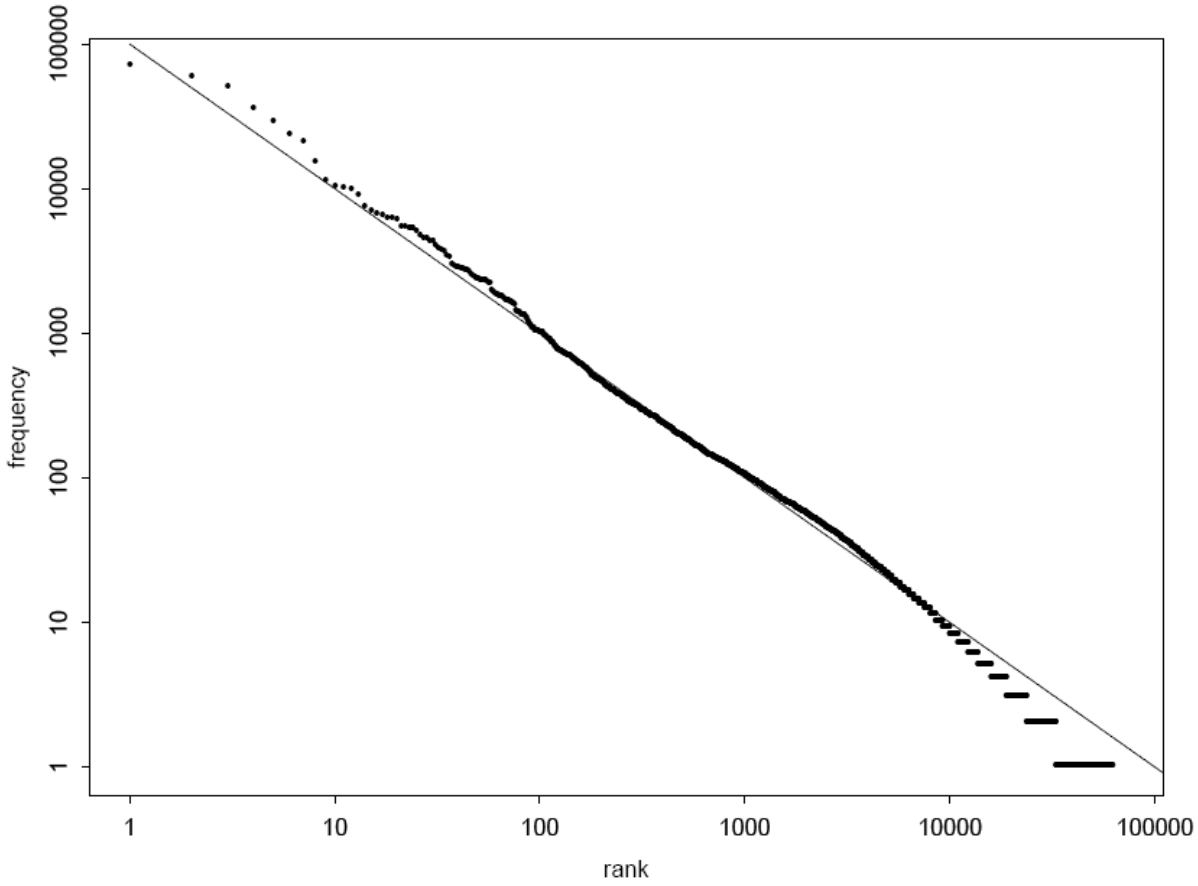
- George Kingsley Zipf (1902-1950) observed the following relation between frequency and rank

$$f \cdot r = c \quad \text{or} \quad f = \frac{c}{r}$$

f = frequency
r = rank
c = constant

  - Example: the 50th most common word should occur three times more often than the 150th most common word
- In other words
  - A few elements occur very frequently
  - Many elements occur very infrequently

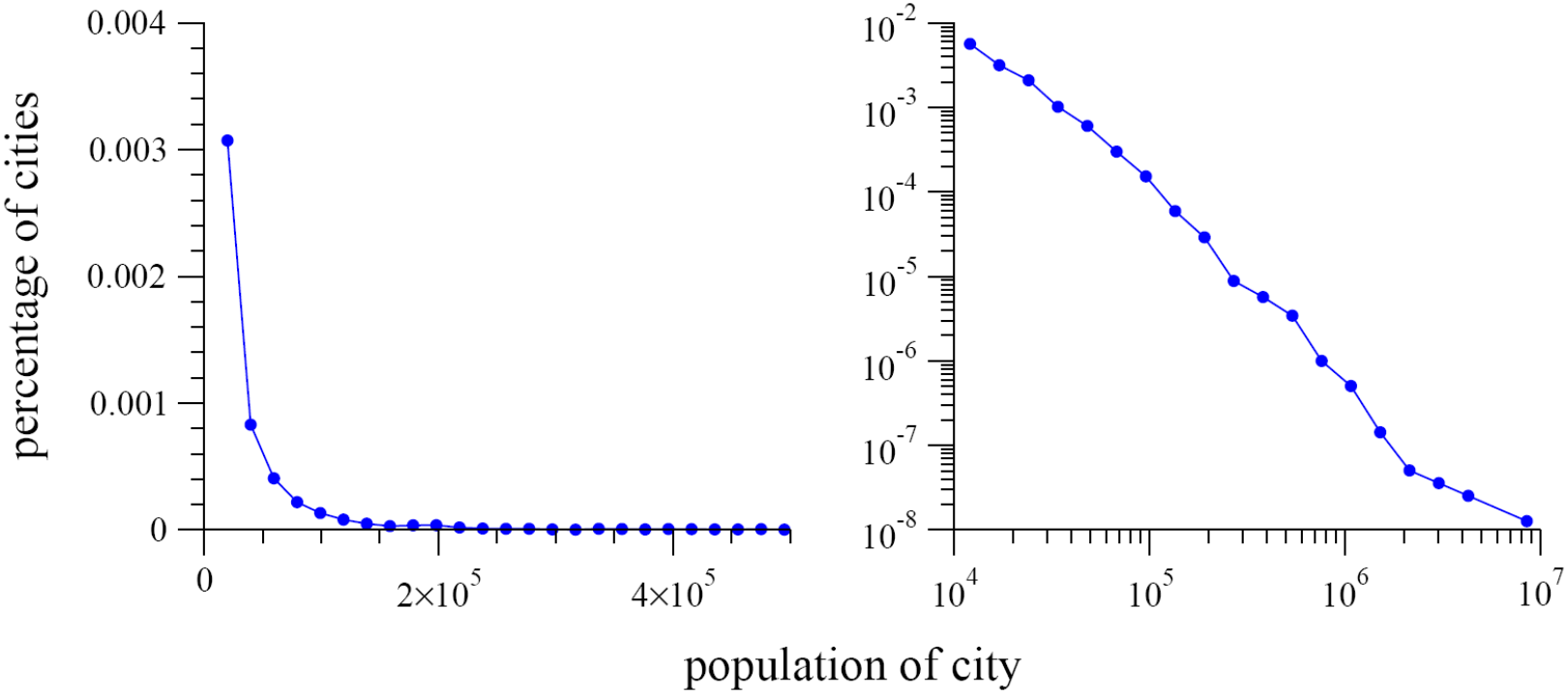
Zipf's Law



Graph illustrating Zipf's Law for the Brown corpus

from Manning and Shütze

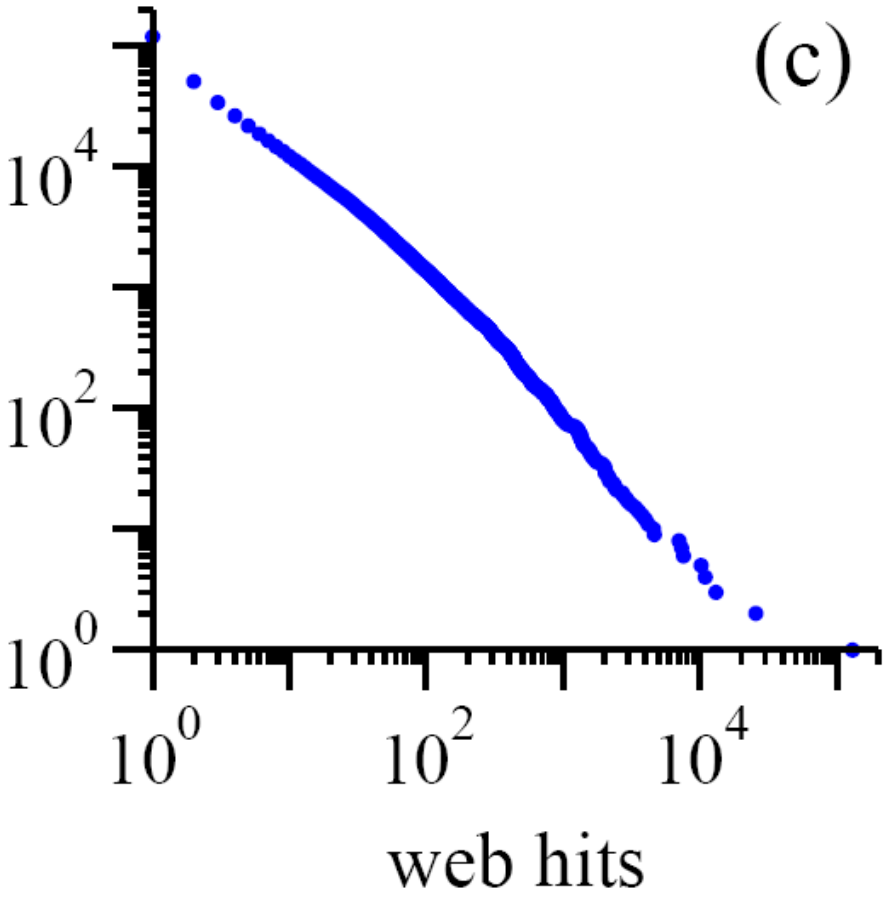
Power Law Distributions: Population



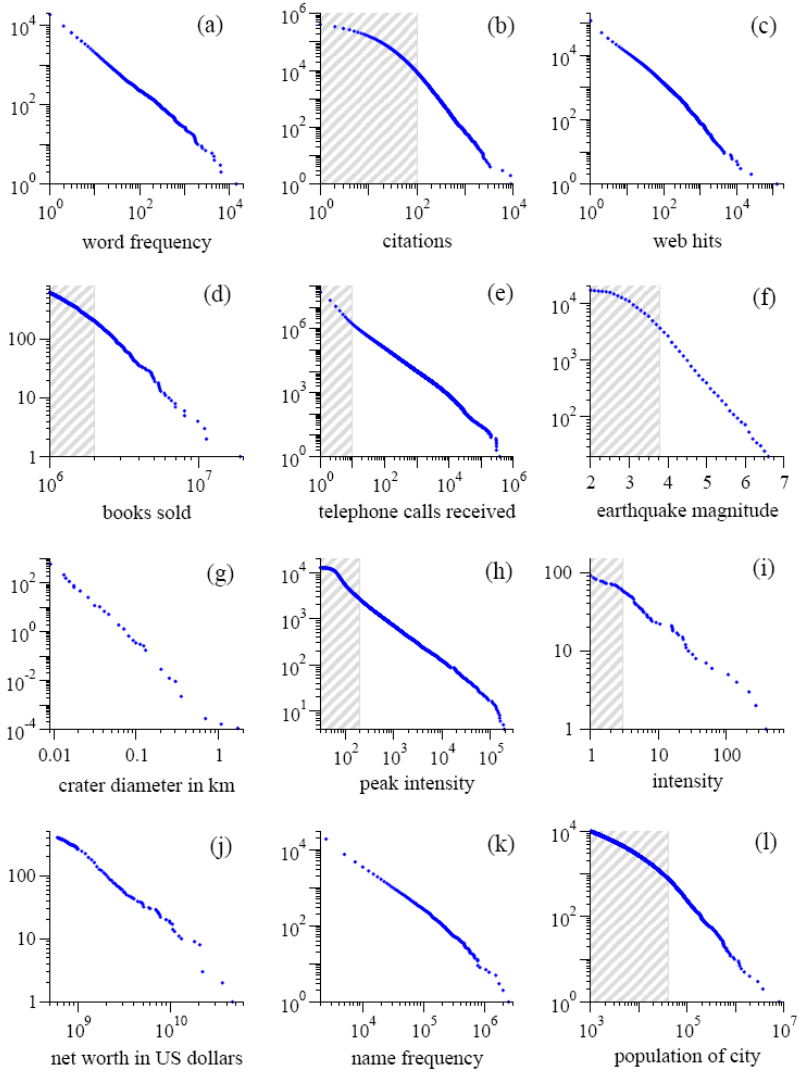
Distribution US cities with population greater than 10,000. Data from 2000 Census.

These and following figures from: Newman, M. E. J. (2005) “Power laws, Pareto distributions and Zipf's law.” Contemporary Physics 46:323–351.

Power Law Distributions: Web Hits



More Power Law Distributions!



What else can we do by counting?

Raw Bigram collocations

Frequency	Word 1	Word 2
80871	of	the
58841	in	the
26430	to	the
21842	on	the
21839	for	the
18568	and	the
16121	that	the
15630	at	the
15494	to	be
13899	in	a
13689	of	a
13361	by	the
13183	with	the
12622	from	the
11428	New	York

Most frequent bigrams collocations in the New York Times, from Manning and Shütze

Filtered Bigram Collocations

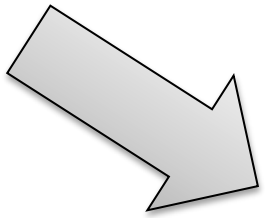
Frequency	Word 1	Word 2	POS
11487	New	York	A N
7261	United	States	A N
5412	Los	Angeles	N N
3301	last	year	A N
3191	Saudi	Arabia	N N
2699	last	week	A N
2514	vice	president	A N
2378	Persian	Gulf	A N
2161	San	Francisco	N N
2106	President	Bush	N N
2001	Middle	East	A N
1942	Saddam	Hussein	N N
1867	Soviet	Union	A N
1850	White	House	A N
1633	United	Nations	A N

Most frequent bigrams collocations in the New York Times filtered by part of speech, from Manning and Shütze

Learning verb "frames"

1	could find a target. The librarian	"showed	off" - running hither and thither w
2	elights in. The young lady teachers	"showed	off" - bending sweetly over pupils
3	ingly. The young gentlemen teachers	"showed	off" with small scoldings and other
4	seeming vexation). The little girls	"showed	off" in various ways, and the littl
5	n various ways, and the little boys	"showed	off" with such diligence that the a
6	t genuwyne?" Tom lifted his lip and	showed	the vacancy. "Well, all right," sai
7	is little finger for a pen. Then he	showed	Huckleberry how to make an H and an
8	ow's face was haggard, and his eyes	showed	the fear that was upon him. When he
9	not overlook the fact that Tom even	showed	a marked aversion to these inquests
10	own. Two or three glimmering lights	showed	where it lay, peacefully sleeping,
11	ird flash turned night into day and	showed	every little grass-blade, separate
12	that grew about their feet. And it	showed	three white, startled faces, too. A
13	he first thing his aunt said to him	showed	him that he had brought his sorrows
14	p from her lethargy of distress and	showed	good interest in the proceedings. S
15	ent a new burst of grief from Becky	showed	Tom that the thing in his mind had
16	shudder quiver all through him. He	showed	Huck the fragment of candle-wick pe

Figure 1.3 Key Word In Context (KWIC) display for the word *showed*.



$$\text{NP}_{agent}\ \text{showed off}\ (\text{PP}[\textit{with/in}]_{manner})$$

$$\text{NP}_{agent}\ \text{showed}\ (\text{NP}_{recipient})\ \left(\left\{\begin{array}{l}\text{NP}_{content}\\ \text{CP}[\textit{that}]_{content}\\ \text{VP[inf]}_{content}\\ \textit{how}\ \text{VP[inf]}_{content}\\ \text{CP}[\textit{where}]_{content}\end{array}\right\}\right)$$

$$\text{NP}_{agent}\ \text{showed NP}[\textit{interest}]\ \text{PP}[\textit{in}]_{content}$$

$$\text{NP}_{agent}\ \text{showed NP}[\textit{aversion}]\ \text{PP}[\textit{to}]_{content}$$

Figure 1.4 Syntactic frames for *showed* in *Tom Sawyer*.

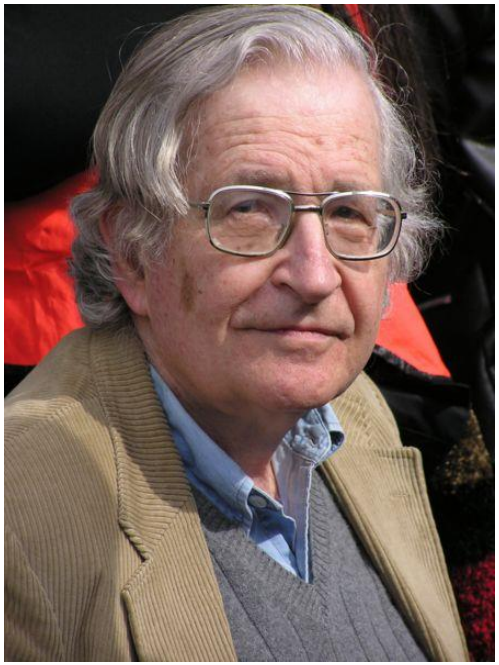
from Manning and Shütze

Today

- Counting words
  - Corpora, types, tokens
  - Zipf's law
- N-gram language models
  - Markov assumption
  - Sparsity
  - Smoothing

N-Gram Language Models

- What?
  - LMs assign probabilities to sequences of tokens
- Why?
  - Autocomplete for phones/websearch
  - Statistical machine translation
  - Speech recognition
  - Handwriting recognition
- How?
  - Based on previous word histories
  - n-gram = consecutive sequences of tokens



Noam Chomsky

But it must be recognized that the notion “probability of a sentence” is an entirely useless one, under any known interpretation of this term. (1969, p. 57)

Fred Jelinek

Anytime a linguist leaves the group the recognition rate goes up. (1988)

N-Gram Language Models

N=1 (unigrams)

Unigrams:
This,
is,
a,
sentence

N-Gram Language Models

N=2 (bigrams)

Bigrams:
This is,
is a,
a sentence

N-Gram Language Models

N=3 (trigrams)

Trigrams:
This is a,
is a sentence

Computing Probabilities

$$P(w_1, w_2, \ldots, w_T)$$

$$= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\ldots P(w_T|w_1, \ldots, w_{T-1})$$

[chain rule]

Approximating Probabilities

Basic idea: limit history to fixed number of words N (Markov Assumption)

$$P(w_k|w_1,\ldots,w_{k-1}) \approx P(w_k|w_{k-N+1},\ldots,w_{k-1})$$

N=1: Unigram Language Model

$$P(w_k|w_1,\ldots,w_{k-1}) \approx P(w_k)$$

$$\Rightarrow P(w_1,w_2,\ldots,w_T) \approx P(w_1)P(w_2)\ldots P(w_T)$$

Approximating Probabilities

Basic idea: limit history to fixed number of words N (Markov Assumption)

$$P(w_k|w_1,\ldots,w_{k-1}) \approx P(w_k|w_{k-N+1},\ldots,w_{k-1})$$

N=2: Bigram Language Model

$$P(w_k|w_1,\ldots,w_{k-1}) \approx P(w_k|w_{k-1})$$

$$\Rightarrow P(w_1,w_2,\ldots,w_T) \approx P(w_1|<\mathrm{S}>)P(w_2|w_1)\ldots P(w_T|w_{T-1})$$

Approximating Probabilities

Basic idea: limit history to fixed number of words N (Markov Assumption)

$$P(w_k|w_1,\ldots,w_{k-1}) \approx P(w_k|w_{k-N+1},\ldots,w_{k-1})$$

N=3: Trigram Language Model

$$P(w_k|w_1,\ldots,w_{k-1}) \approx P(w_k|w_{k-2},w_{k-1})$$

$$\Rightarrow P(w_1,w_2,\ldots,w_T) \approx P(w_1|<\mathrm{S}><\mathrm{S}>)\ldots P(w_T|w_{T-2}w_{T-1})$$

Building N-Gram Language Models

- Use existing sentences to compute n-gram probability estimates (training)
- Terminology:
  - N = total number of words in training data (tokens)
  - V = vocabulary size or number of unique words (types)
  - $C(w_1,...,w_k)$ = frequency of n-gram w_1, ..., w_k in training data
  - $P(w_1, ..., w_k)$ = probability estimate for n-gram w_1 ... w_k
  - $P(w_k|w_1, ..., w_{k-1})$ = conditional probability of producing w_k given the history w_1, ... w_{k-1}

What's the vocabulary size?

Vocabulary Size: Heaps' Law

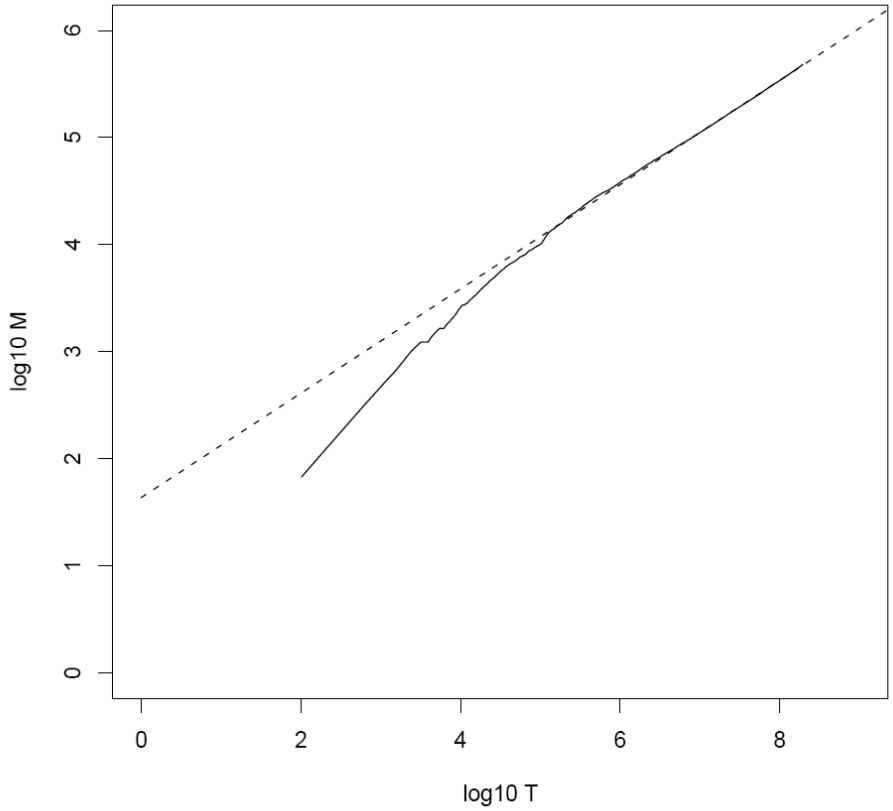
$$M = kT^b$$

M is vocabulary size
T is collection size (number of documents)
k and *b* are constants

Typically, *k* is between 30 and 100, *b* is between 0.4 and 0.6

- Heaps' Law: linear in log-log space
- Vocabulary size grows unbounded!

Heaps' Law for RCV1



First 1,000,020 terms:
Predicted = 38,323
Actual = 38,365

Building N-Gram Models

- Compute maximum likelihood estimates for individual n-gram probabilities
  - Unigram: $P(w_i) = \frac{C(w_i)}{N}$
  - Bigram: $P(w_i, w_j) = \frac{C(w_i, w_j)}{N}$

$$P(w_j|w_i) = \frac{P(w_i, w_j)}{P(w_i)} = \frac{C(w_i, w_j)}{\sum_w C(w_i, w)} = \frac{C(w_i, w_j)}{C(w_i)}$$

- Uses relative frequencies as estimates

Example: Bigram Language Model

<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like green eggs and ham </s>

Training Corpus

P(I | <s>) = 2/3 = 0.67	P(Sam | <s>) = 1/3 = 0.33
P(am | I) = 2/3 = 0.67	P(do | I) = 1/3 = 0.33
P(</s> | Sam)= 1/2 = 0.50	P(Sam | am) = 1/2 = 0.50
...	

Bigram Probability Estimates

Note: We don't ever cross sentence boundaries

More Context, More Work

- Larger N = more context
  - Lexical co-occurrences
  - Local syntactic relations
- More context is better?
- Larger N = more complex model
  - For example, assume a vocabulary of 100,000
  - How many parameters for unigram LM? Bigram? Trigram?
- Larger N has another problem…

Data Sparsity

P(I | <s>) = 2/3 = 0.67	P(Sam | <s>) = 1/3 = 0.33
P(am | I) = 2/3 = 0.67	P(do | I) = 1/3 = 0.33
P(</s> | Sam)= 1/2 = 0.50	P(Sam | am) = 1/2 = 0.50
…	

Bigram Probability Estimates

P(I like ham)

= P(I | <s>) P(like | I) P(ham | like) P(</s> | ham)

= 0

Why is this bad?

Data Sparsity

- Serious problem in language modeling!
- Becomes more severe as N increases
  - What's the tradeoff?
- Solution 1: Use larger training corpora
  - But Zipf's Law
- Solution 2: Assign non-zero probability to unseen n-grams
  - Known as smoothing

Smoothing

- Zeros are bad for any statistical estimator
  - Need better estimators because MLEs give us a lot of zeros
  - A distribution without zeros is “smoother”
- The Robin Hood Philosophy: Take from the rich (seen n-grams) and give to the poor (unseen n-grams)
  - And thus also called discounting
  - Critical: make sure you still have a valid probability distribution!

Laplace's Law

- Simplest and oldest smoothing technique
- Just add 1 to all n-gram counts including the unseen ones
- So, what do the revised estimates look like?

Laplace's Law: Probabilities

$$P_{MLE}(w_i) = \frac{C(w_i)}{N} \longrightarrow P_{LAP}(w_i) = \frac{C(w_i) + 1}{N + V}$$

$$P_{MLE}(w_i, w_j) = \frac{C(w_i, w_j)}{N} \longrightarrow P_{LAP}(w_i, w_j) = \frac{C(w_i, w_j) + 1}{N + V^2}$$

Careful, don't confuse the N's!

Laplace's Law: Frequencies

Expected Frequency Estimates

$$\begin{aligned} C_{LAP}(w_i) &= P_{LAP}(w_i)N \\ C_{LAP}(w_i, w_j) &= P_{LAP}(w_i, w_j)N \end{aligned}$$

Relative Discount

$$\begin{aligned} d_1 &= \frac{C_{LAP}(w_i)}{C(w_i)} \\ d_2 &= \frac{C_{LAP}(w_i, w_j)}{C(w_i, w_j)} \end{aligned}$$

Laplace's Law

- Bayesian estimator with uniform priors
- Moves too much mass over to unseen n-grams
- We can add a fraction of 1 instead
  - add $0 < \gamma < 1$ to each count instead

Also: backoff Models

- Consult different models in order depending on specificity (instead of all at the same time)
- The most detailed model for current context first and, if that doesn't work, back off to a lower model
- Continue backing off until you reach a model that has some counts

- In practice: Kneser-Ney smoothing (J&M 4.9.1)

Explicitly Modeling OOV

- Fix vocabulary at some reasonable number of words

- During training:
  - Consider any words that don't occur in this list as unknown or **out of vocabulary (OOV)** words
  - Replace all OOVs with the special word <UNK>
  - Treat <UNK> as any other word and count and estimate probabilities

- During testing:
  - Replace unknown words with <UNK> and use LM
  - Test set characterized by OOV rate (percentage of OOVs)

Evaluating Language Models

- Information theoretic criteria used
- Most common: Perplexity assigned by the trained LM to a test set
- Perplexity: How surprised are you on average by what comes next ?
  - If the LM is good at knowing what comes next in a sentence ⇒ Low perplexity (lower is better)

Computing Perplexity

- Given test set *W* with words w_1, ...,w_N
- Treat entire test set as one word sequence
- Perplexity is defined as the probability of the entire test set normalized by the number of words

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

- Using the probability chain rule and (say) a bigram LM, we can write this as

$$PP(T) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}}$$

Practical Evaluation

- Use <s> and </s> both in probability computation
- Count </s> but not <s> in *N*
- Typical range of perplexities on English text is 50-1000
- Closed vocabulary testing yields much lower perplexities
- Testing across genres yields higher perplexities
- Can only compare perplexities if the LMs use the same vocabulary

Order	Unigram	Bigram	Trigram
PP	962	170	109

Training: N=38 million, V~20000, open vocabulary, Katz backoff where applicable
Test: 1.5 million words, same genre as training

Typical LMs in practice…

- Training
  - N = 10 billion words, V = 300k words
  - 4-gram model with Kneser-Ney smoothing
- Testing
  - 25 million words, OOV rate 3.8%
  - Perplexity ~50

Take-Away Messages

- Counting words
  - Corpora, types, tokens
  - Zipf's law

- N-gram language models
  - LMs assign probabilities to sequences of tokens
  - N-gram models: consider limited histories
  - Data sparsity is an issue: smoothing to the rescue