text statistics

many slides courtesy James Allan@umass

outline

- Zipf's law
- Heap's Law
- log-log plots
- least squares fitting
- information theory
- collocations
- Markov Models

frequent words

Word	Occurrences	Percentage
the	8,543,794	6.8
of	3,893,790	3.1
to	3,364,653	2.7
and	3,320,687	2.6
in	2,311,785	1.8
is	1,559,147	1.2
for	1,313,561	1.0
that	1,066,503	0.8
said	1,027,713	0.8

Frequencies from 336,310 documents in the 1GB TREC Volume 3 Corpus
125,720,891 total word occurrences; 508,209 unique words

Zipf's law

- A few words occur very often
  - 2 most frequent words can account for 10% of occurrences
  - top 6 words are 20%, top 50 words are 50%
- Many words are infrequent
- "Principle of Least Effort"
  - easier to repeat words rather than coining new ones

- Rank · Frequency ≈ Constant
  - pr = (Number of occurrences of word of rank r)/N
  - N total word occurrences
  - probability that a word chosen randomly from the text will be the word of rank r
  - for D unique words $\Sigma\, p_r = 1$

  - r ·pr = A
  - A ≈ 0.1

George Kingsley Zipf, 1902-1950
Linguistic professor at Harvard

Zipf's law

Word	Freq	r	Pr	r*Pr
the	15659	1	6.422	0.0642
of	7179	2	2.944	0.0589
to	6287	3	2.578	0.0774
a	5830	4	2.391	0.0956
and	5580	5	2.288	0.1144
in	5245	6	2.151	0.1291
that	2494	7	1.023	0.0716
for	2197	8	0.901	0.0721
was	2147	9	0.881	0.0792
with	1824	10	0.748	0.0748
his	1813	11	0.744	0.0818
is	1800	12	0.738	0.0886
he	1687	13	0.692	0.0899
as	1576	14	0.646	0.0905
on	1523	15	0.625	0.0937
by	1443	16	0.592	0.0947
at	1318	17	0.541	0.0919
it	1232	18	0.505	0.0909
from	1217	19	0.499	0.0948
but	1136	20	0.466	0.0932
u	949	21	0.389	0.0817
had	937	22	0.384	0.0845
last	909	23	0.373	0.0857
be	906	24	0.372	0.0892
who	883	25	0.362	0.0905

Word	Freq	r	Pr	r*Pr
has	880	26	0.361	0.0938
not	875	27	0.359	0.0969
an	863	28	0.354	0.0991
s	862	29	0.354	0.1025
have	860	30	0.353	0.1058
were	858	31	0.352	0.1091
their	812	32	0.333	0.1066
are	807	33	0.331	0.1092
one	742	34	0.304	0.1035
they	679	35	0.278	0.0975
its	668	36	0.274	0.0986
all	646	37	0.265	0.098
week	626	38	0.257	0.0976
government	582	39	0.239	0.0931
when	577	40	0.237	0.0947
would	572	41	0.235	0.0962
been	554	42	0.227	0.0954
out	553	43	0.227	0.0975
new	544	44	0.223	0.0982
which	539	45	0.221	0.0995
up	539	45	0.221	0.0995
more	535	47	0.219	0.1031
into	516	48	0.212	0.1016
only	504	49	0.207	0.1013
will	488	50	0.2	0.1001

Top 50 words from 423 short TIME magazine articles

Zipf's law

Word	Freq	r	Pr(%)	r*Pr
the	2,420,778	1	6.488	0.0649
of	1,045,733	2	2.803	0.0561
to	968,882	3	2.597	0.0779
a	892,429	4	2.392	0.0957
and	865,644	5	2.32	0.116
in	847,825	6	2.272	0.1363
said	504,593	7	1.352	0.0947
for	363,865	8	0.975	0.078
that	347,072	9	0.93	0.0837
was	293,027	10	0.785	0.0785
on	291,947	11	0.783	0.0861
he	250,919	12	0.673	0.0807
is	245,843	13	0.659	0.0857
with	223,846	14	0.6	0.084
at	210,064	15	0.563	0.0845
by	209,586	16	0.562	0.0899
it	195,621	17	0.524	0.0891
from	189,451	18	0.508	0.0914
as	181,714	19	0.487	0.0925
be	157,300	20	0.422	0.0843
were	153,913	21	0.413	0.0866
an	152,576	22	0.409	0.09
have	149,749	23	0.401	0.0923
his	142,285	24	0.381	0.0915
but	140,880	25	0.378	0.0944

Word	Freq	r	Pr(%)	r*Pr
has	136,007	26	0.365	0.0948
are	130,322	27	0.349	0.0943
not	127,493	28	0.342	0.0957
who	116,364	29	0.312	0.0904
they	111,024	30	0.298	0.0893
its	111,021	31	0.298	0.0922
had	103,943	32	0.279	0.0892
will	102,949	33	0.276	0.0911
would	99,503	34	0.267	0.0907
about	92,983	35	0.249	0.0872
i	92,005	36	0.247	0.0888
been	88,786	37	0.238	0.0881
this	87,286	38	0.234	0.0889
their	84,638	39	0.227	0.0885
new	83,449	40	0.224	0.0895
or	81,796	41	0.219	0.0899
which	80,385	42	0.215	0.0905
we	80,245	43	0.215	0.0925
more	76,388	44	0.205	0.0901
after	75,165	45	0.201	0.0907
us	72,045	46	0.193	0.0888
percent	71,956	47	0.193	0.0906
up	71,082	48	0.191	0.0915
one	70,266	49	0.188	0.0923
people	68,988	50	0.185	0.0925

Top 50 words from 84,678 Associated Press 1989 articles

Zipf's Law and H.P.Luhn

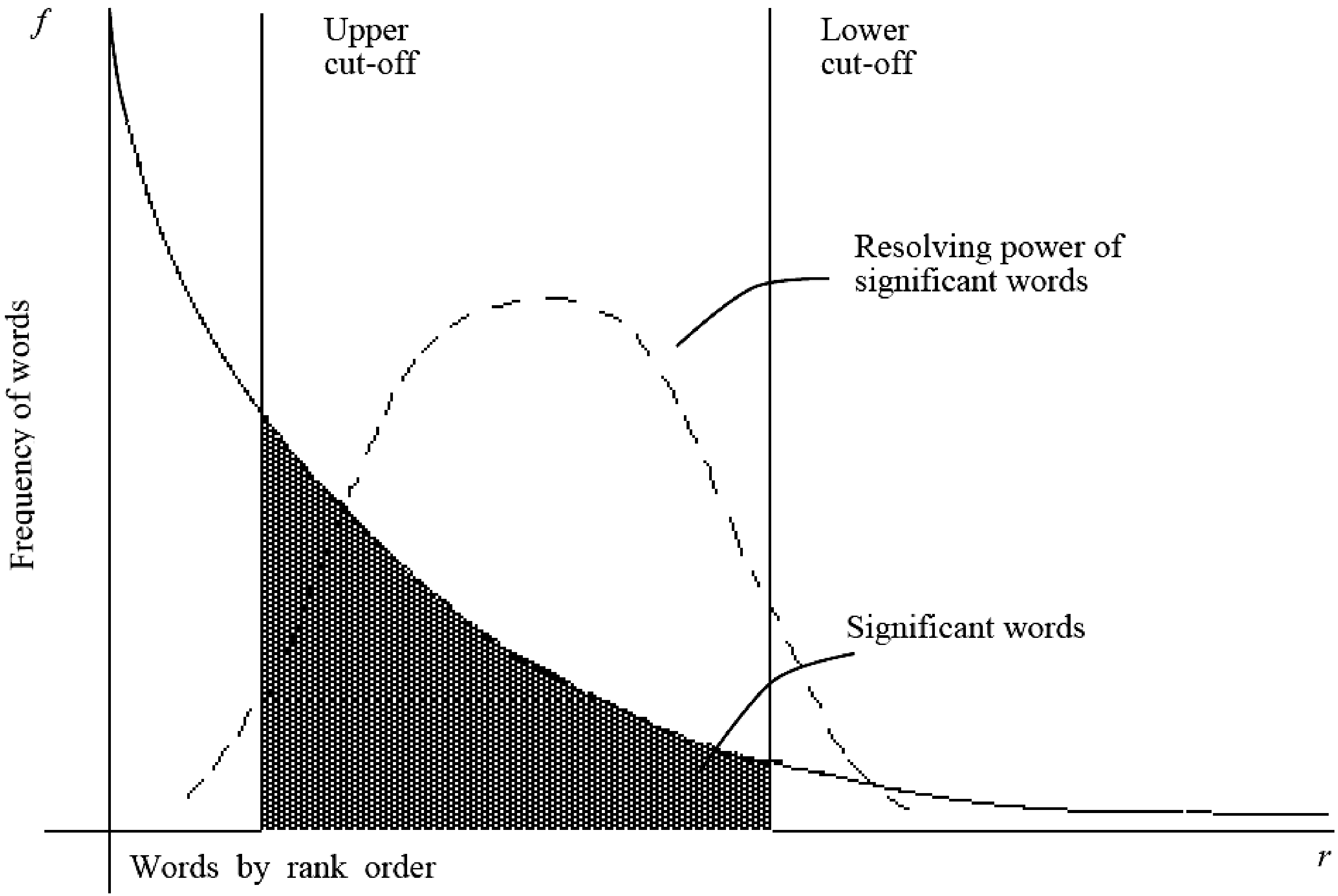


Figure 2.1. A plot of the hyperbolic curve relating f, the frequency of occurrence and r, the rank order (Adaped from Schultz [44] page 120)

Zipf's law: predicting frequencies

$$r \cdot p_r = A$$

- A word that occurs n times has rank $r_n = AN/n$
- Several words may occur n times
- Assume rank given by r_n applies to *last* of the words that occur n times
- r_n words occur n times or more (ranks $1..r_n$)
- r_{n+1} words occur n+1 times or more
  - Note: $r_n > r_{n+1}$ since words that occur frequently are at the start of list (lower rank)

$r_{98}=4$
$r_{90}=7$
$r_{79}=20$

1. the	100
2. of	98
3. to	98
4. a	98
5. and	90
6. in	90
7. said	90
8. for	88
9. that	88
10. was	87
11. on	85
12. he	85
13. is	85
14. with	85
15. at	84
16. by	83
17. it	80
18. from	79
19. as	79
20. be	79
21. were	78
22. an	78
23. have	73
24. his	73
25. but	72

Zipf's law: predicting frequencies

$$r \cdot p_r = A$$

- The number of words that occur exactly n times is
$I_n = r_n - r_{n+1} = AN/n - AN/(n+1) = AN / (n(n+1))$

- Highest ranking term occurs once and has rank
$D = AN/1$

- Proportion of words with frequency n is
$I_n/D = 1/ (n(n+1))$

- Proportion of words occurring once is 1/2

Zipf's law: predicting frequencies

Rank	Predicted Proportion of Occurrences 1/n(n+1)	Actual Proportion occurring n times I_n/D	Actual Number of Words occurring n times
1	.500	.402	204,357
2	.167	.132	67,082
3	.083	.069	35,083
4	.050	.046	23,271
5	.033	.032	16,332
6	.024	.024	12,421
7	.018	.019	9,766
8	.014	.016	8,200
9	.011	.014	6,907
10	.009	.012	5,893

Frequencies from 336,310 documents in the 1GB TREC Volume 3 Corpus

Zipf's law and real data

- A law of the form $y = kxc$ is called a power law.

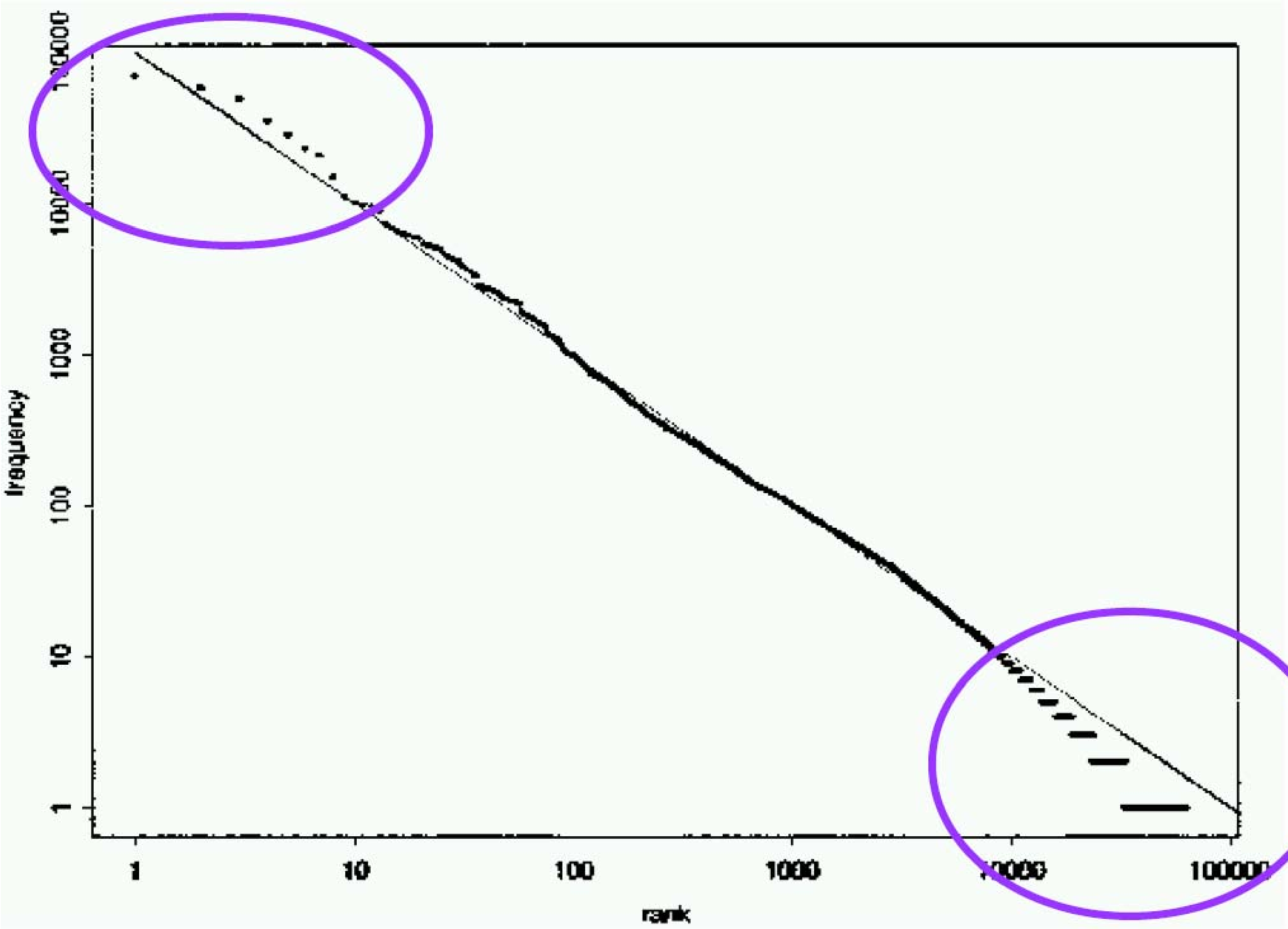
- Zipf's law is a power law with $c = -1$
  - $r = A \cdot n^{-1} \Rightarrow n = A \cdot r^{-1}$
  - A is a constant for a fixed collection

- On a log-log plot, power laws give a straight line with slope c.

- $\log(\ y\ )\ \log(kx^c\ ) = \log(\ k\ ) + c\log(x^c)$
- $\log(n) = \log(Ar^{-1}) = \log(A) - 1 \cdot \log(r)$

- Zipf is quite accurate except for very high and low rank.

high and low ranks

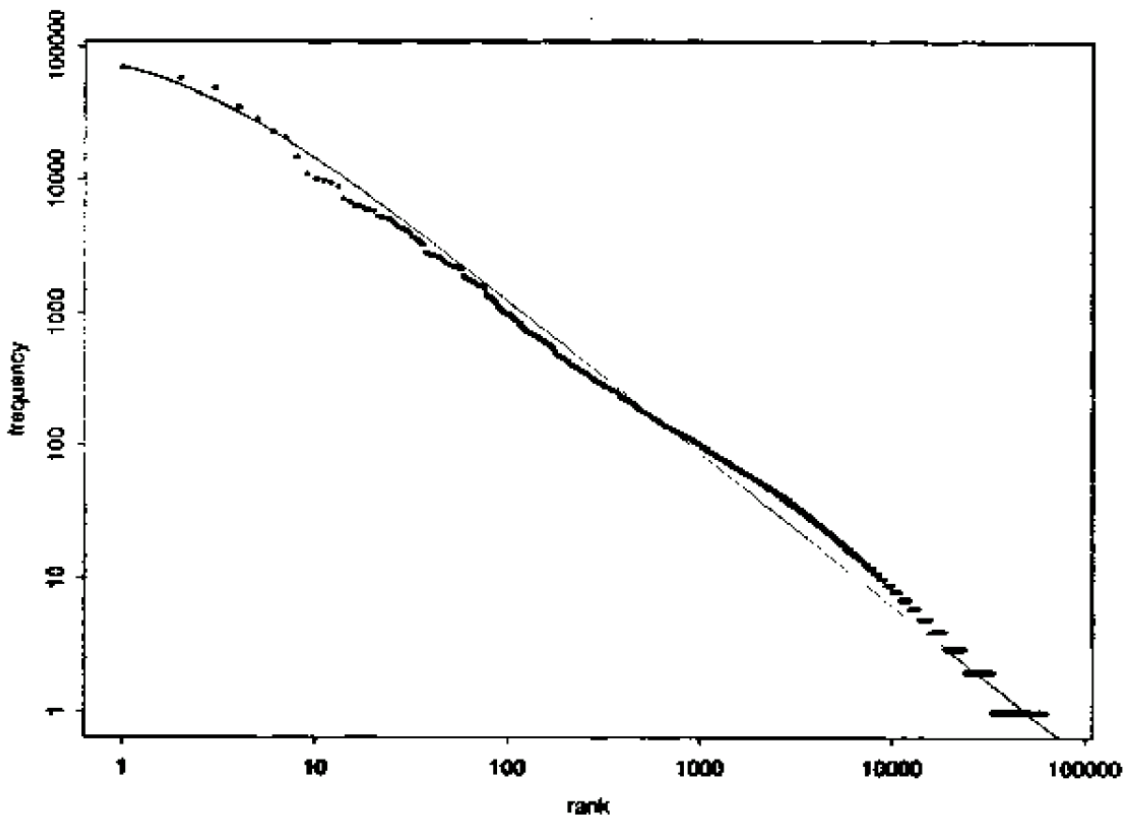


$k = AN = 100{,}000$

Zipf's law: Mandelbrot correction

- The following more general form gives bit better fit
  - Adds a constant to the denominator
  - $y=k(x+t)^c$

- Here,
  $n = A \cdot (r+t)^{-1}$



Zipf's law

- Zipf's explanation was his "principle of least effort."

- Balance between speaker's desire for a small vocabulary and hearer's desire for a large one.

- Debate (1955-61) between Mandelbrot and H. Simon over explanation.

- Li (1992) shows that just random typing of letters including a space will generate "words" with a Zipfian distribution.
  - http://linkage.rockefeller.edu/wli/zipf/
  - Short words more likely to be generated

Explanations for Zipf Law

- Zipf's explanation was his "principle of least effort." Balance between speaker's desire for a small vocabulary and hearer's desire for a large one.

- Debate (1955-61) between Mandelbrot and H. Simon over explanation

- Li (1992) shows that just random typing of letters including a space will generate "words" with a Zipfian distribution.
  - http://linkage.rockefeller.edu/wli/zipf/
  - Short words more likely to be generated

Heap's law

- How does the size of the overall vocabulary (number of unique words) grow with the size of the corpus?
  - Vocabulary has no upper bound due to proper names, typos, etc.
  - New words occur less frequently as vocabulary grows

- If V is the size of the vocabulary and the N is the length of the corpus in words:
  - $V = KN^{\beta}$ ($0 < \beta < 1$)

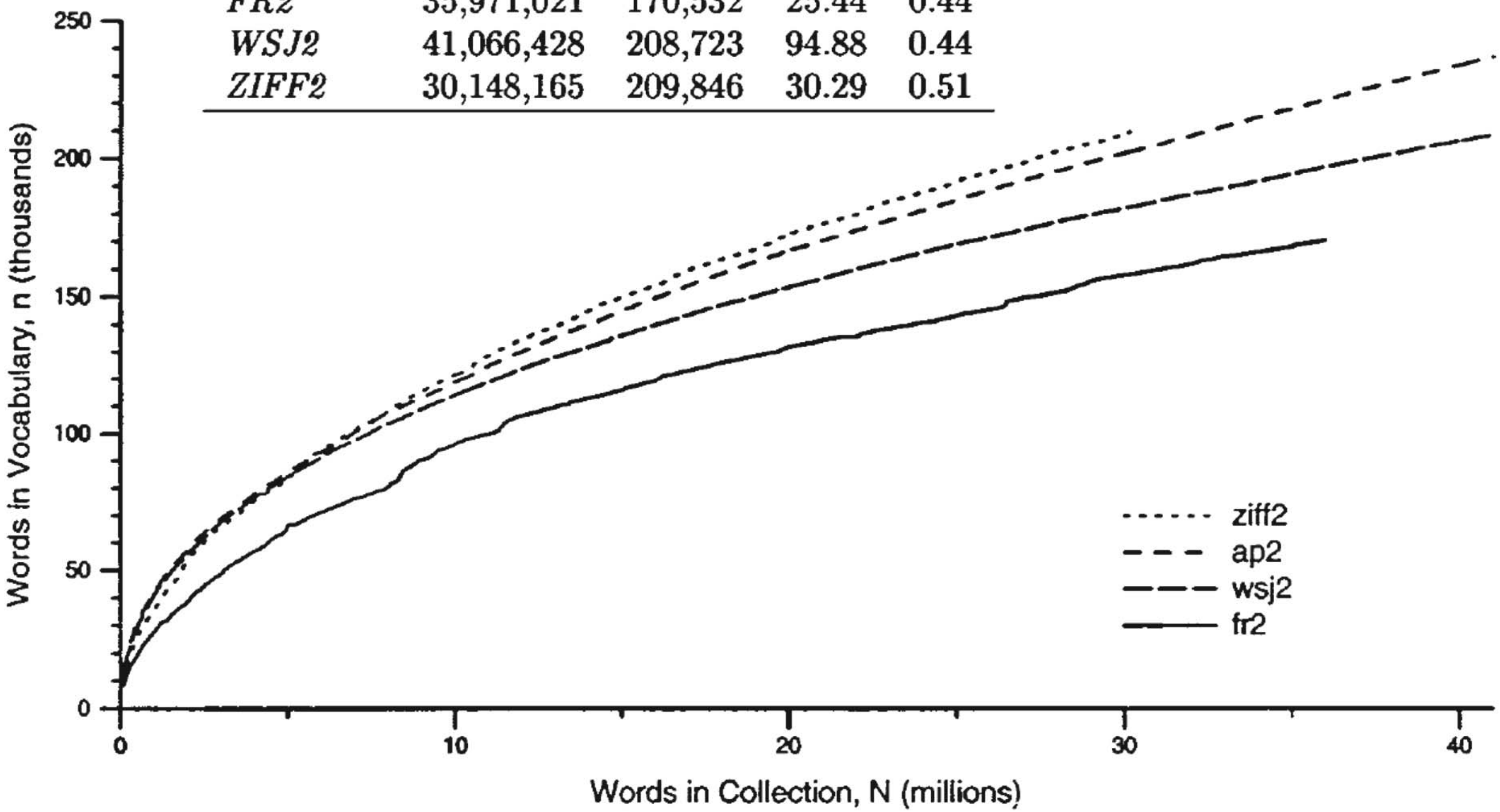
- Typical constants:
  - $K \approx 10-100$
  - $\beta \approx 0.4-0.6$ (approx. square-root of n)

- Can be derived from Zipf's law by assuming documents are generated by randomly sampling words from a Zipfian distribution

Heap's law

$V = n = KN^{\beta}$

Collection	N	n	K	β
AP2	40,998,865	237,160	63.11	0.47
FR2	35,971,021	170,532	25.44	0.44
WSJ2	41,066,428	208,723	94.88	0.44
ZIFF2	30,148,165	209,846	30.29	0.51



outline

- Zipf's law
- Heap's Law
- log-log plots
- least squares fitting
- information theory
- collocations
- Markov Models

on board

outline

- Zipf's law
- Heap's Law
- log-log plots
- least squares fitting
- information theory
- collocations
- Markov Models

information theory

- Shannon studied theoretical limits for data compression and transmission rate

- Compression limits given by Entropy (H)

- Transmission limits given by Channel Capacity (C)

- A number of language tasks have been formulated as a “noisy channel” problem
  - i.e., determine the most likely input given the noisy output
  - OCR
  - Speech recognition
  - Question answering
  - Machine translation
  - …

information theory

- The President of the United States is George W. ...
- The winner of the $10K prize is ...
- Mary had a little ...
- The horse raced past the barn ...
  - Period (end of sentence)
  - "whinnied" (garden path sentence)

information theory

- *Information content* of a message is dependent on the receiver's prior knowledge as well as on the message itself

- How much of the receiver's uncertainty (entropy) is reduced

- How predictable is the message

information theory

$$H = \sum_{r=1}^{n} p_r \log \frac{1}{p_r}$$

- Given n messages, the average or expected information content to be gained through receipt of one of the n possible messages is
- entropy is a maximum when messages are equally probable
  - average entropy associated with characters assuming equal probab
- So for alphabet, entropy is log(26) = 4.7 bits
- Taking actual probabilities into account, entropy is 4.14 bits
- With bigram probabilities, reduces entropy to 3.56 bits
- Experiments with people give values around 1.3 bits
  - Can predict next letter with about 40% chance of accuracy

- Zipf's Law with entropy :

$$r \cdot p_r = A$$

$$H = \sum_{r=1}^{n} \frac{A}{r} \log \frac{r}{A}$$

information theory

- joint entropy $H(X,Y) = \sum_{x,y} p(x,y) \log \frac{1}{p(x,y)}$

- conditionalentropy $H(Y|X) = \sum_{x,y} p(x,y) \log \frac{1}{p(y|x)}$

- mutual information $I(X,Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$

- relative entropy (KL distance)

$$KL(p||q) = \sum_{x} p(x) \log \frac{p(x)}{q(x)}$$

mutual information

- symmetric, non-negative measure of common information between 2 random variabiles
- $I(X,Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$
- $I(X,Y) = KL(p(x,y)||p(x)p(y))$
- $I(X,Y) = H(X) - H(X|Y) = H(X) + H(Y) - H(X,Y)$
- $H(X) + H(Y|X) = H(Y) + H(X|Y) = H(X,Y)$

collocations

- Co-occurrence patterns of words and word classes reveal significant information about how a language is used
  - pragmatics

- Used in building dictionaries (lexicography) and for IR tasks such as phrase detection, query expansion, etc.

- Co-occurrence based on *text windows*
  - typical window may be 100 words
  - smaller windows used for lexicography, e.g. adjacent pairs or 5 words

collocations

Relation	Word x	Word y	Separation	
			mean	variance
fixed	*bread*	*butter*	2.00	0.00
	drink	*drive*	2.00	0.00
compound	*computer*	*scientist*	1.12	0.10
	United	*States*	0.98	0.14
semantic	*man*	*woman*	1.46	8.07
	man	*women*	-0.12	13.08
lexical	*refraining*	*from*	1.11	0.20
	coming	*from*	0.83	2.89
	keeping	*from*	2.14	5.53

Word Pair Statistics from 1988 AP Corpus (Church and Hanks)

collocations

- Typical measure used is the point version of the mutual information measure (compared to the *expected* value of I, sometimes called EMIM)

$$I(x, y) = \log \frac{p(x,y)}{p(x)p(y)}$$

- Paired t test also used to compare collocation probabilities

$$t = \frac{\overline{x} - \overline{y}}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

collocations

Table 1: Some Interesting Associations with *strong* and *powerful* in the 1988 AP Corpus (N = 44.3 million)

I(x;y)	fxy	fx	fy	x	y
10.47	7	7809	28	strong	northerly
9.76	23	7809	151	strong	showings
9.30	7	7809	63	strong	believer
9.22	14	7809	133	strong	second-place
9.17	6	7809	59	strong	runup
9.04	10	7809	108	strong	currents
8.85	62	7809	762	strong	supporter
8.84	8	7809	99	strong	proponent
8.68	15	7809	208	strong	thunderstorm
8.45	7	7809	114	strong	odor
8.66	7	1984	388	powerful	legacy
8.58	7	1984	410	powerful	tool
8.35	8	1984	548	powerful	storms
8.32	31	1984	2169	powerful	minority
8.14	9	1984	714	powerful	neighbor
7.98	9	1984	794	powerful	Tamil
7.93	8	1984	734	powerful	symbol
7.74	32	1984	3336	powerful	figure
7.54	10	1984	1204	powerful	weapon
7.47	24	1984	3029	powerful	post

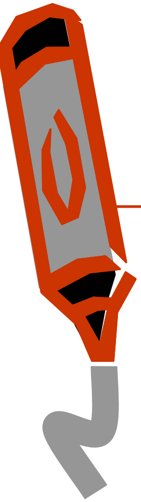
collocations

Table 8: What does a boat do?
(N = 24,677,658; f(x, y) ≥ 3).

I(x;y)	f(x,y)	f(x)	f(y)	x	y	I(x;y)	f(x,y)	f(x)	f(y)	x	y
11.01	16	984	194	boat/S	capsize/V	3.09	4	984	11768	boat/S	fail/V
9.30	51	984	2036	boat/S	sink/V	2.72	4	984	15244	boat/S	stop/V
8.17	3	984	262	boat/S	cruise/V	2.59	5	984	20894	boat/S	accord/V
7.40	6	984	890	boat/S	sail/V	2.54	4	984	17266	boat/S	reach/V
7.27	3	984	488	boat/S	tow/V	2.14	3	984	17074	boat/S	lose/V
7.18	3	984	518	boat/S	turn_in/V	2.09	6	984	35456	boat/S	leave/V
6.83	3	984	660	boat/S	collide/V	2.04	4	984	24410	boat/S	keep/V
6.61	3	984	772	boat/S	drown/V	2.04	6	984	36494	boat/S	kill/V
6.34	4	984	1238	boat/S	drag/V	1.69	6	984	46624	boat/S	be_in/V
6.28	3	984	968	boat/S	escort/V	1.61	3	984	24714	boat/S	put/V
6.04	4	984	1522	boat/S	overturn/V	1.38	8	984	77238	boat/S	take/V
5.90	5	984	2096	boat/S	rescue/V	1.36	3	984	29338	boat/S	hold/V
5.43	5	984	2902	boat/S	approach/V	1.28	4	984	41232	boat/S	use/V
4.64	16	984	16068	boat/S	carry/V	1.26	3	984	31506	boat/S	become/V
4.43	9	984	10470	boat/S	hit/V	0.94	19	984	247542	boat/S	have/V
4.18	4	984	5524	boat/S	travel/V	0.67	3	984	47214	boat/S	begin/V
3.86	6	984	10348	boat/S	pass/V	0.57	3	984	50766	boat/S	get/V
3.71	4	984	7656	boat/S	attack/V	0.17	4	984	89256	boat/S	do/V
3.48	3	984	6748	boat/S	injure/V	−0.35	26	984	830120	boat/S	be/V
3.38	4	984	9614	boat/S	fire/V	−0.35	3	984	95880	boat/S	make/V
3.30	3	984	7634	boat/S	operate/V	−3.38	4	984	1045494	boat/S	say/V

outline

- Zipf's law
- Heap's Law
- log-log plots
- least squares fitting
- information theory
- collocations
- Markov Models



Markov Models

Modeling a sequence of events where probability depends on previous events

- Markov properties
  - Limited Horizon

    $$P(X_{t+1} = k | X_1, X_2, ..., X_t) = P(X_{t+1} = k | X_t)$$

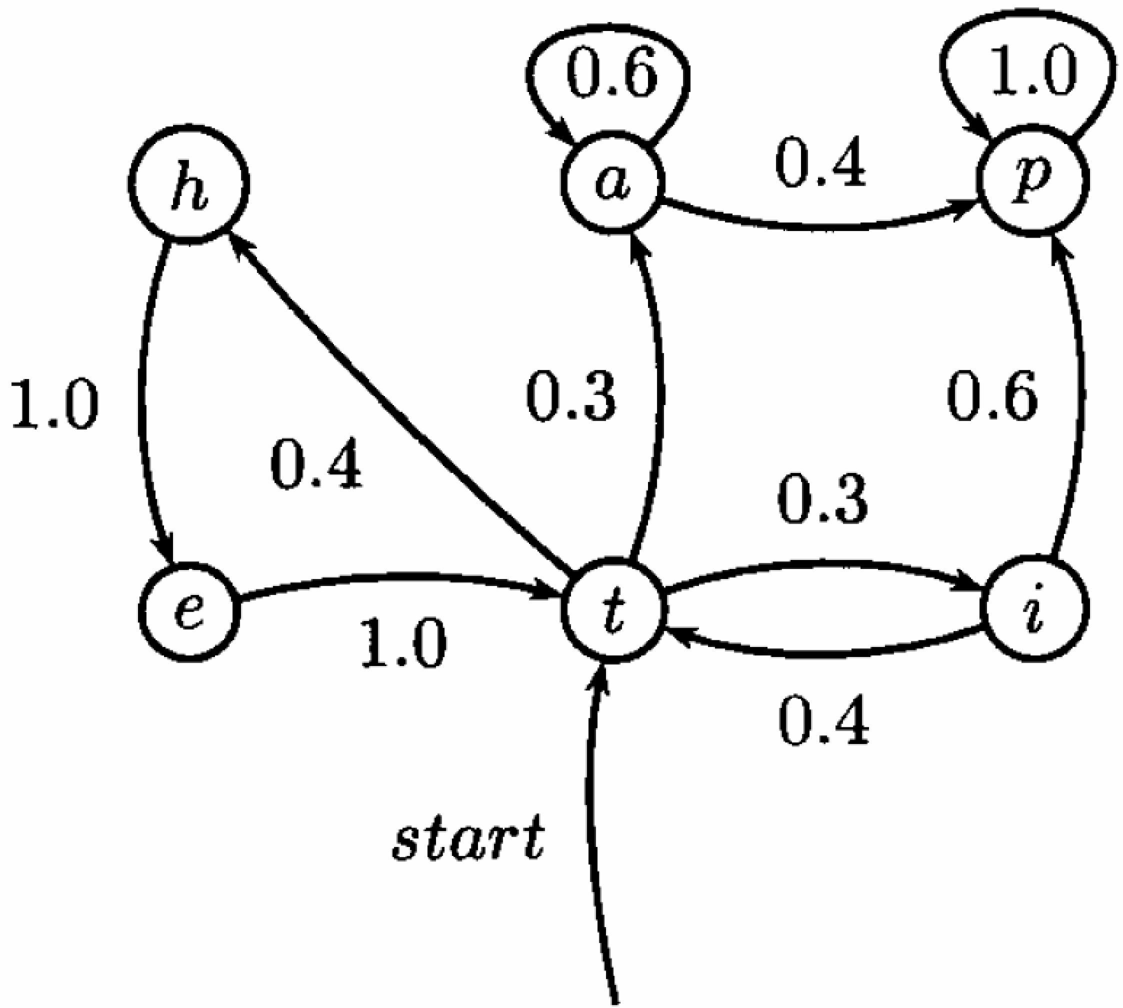
  - Time invariant

    $$= P(X_2 = k | X_1)$$

- A Markov chain is described by a transition probability matrix

$$a_{ij} = p(X_{t+1} = s_j | X_t = s_i)$$

Markov Models



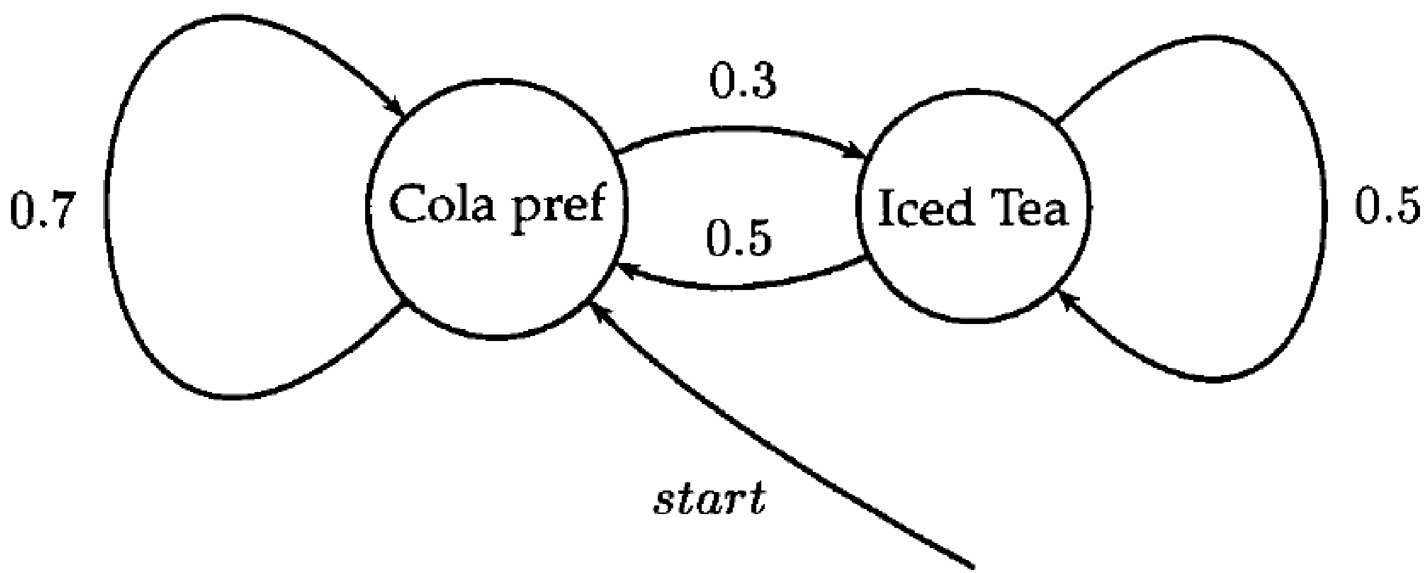
(from Manning and Schutze)

Hidden Markov Models

- Don't know state sequence that the model passes through, only some probabilistic function of it
  - underlying events probabilistically generating surface events

- Both regular and hidden Markov models used for part of speech tagging
  - regular is trained using a tagged corpus
  - HMM approach assumes that an underlying Markov chain of parts of speech generates actual words in the text

Hidden Markov Models



Output probability given From state

	cola	iced tea	lemonade
CP	0.6	0.1	0.3
IP	0.1	0.7	0.2

(From Manning and Schutze)

Hidden Markov Models

- 3 basic questions
  - Given a model, how do we efficiently compute how likely a certain observation is?
  - Given an observation sequence and a model, how do we choose a state sequence that best explains the observations
  - Given an observation sequence and a space of possible models, how do we find the model that best explains the observations

- Viterbi algorithm commonly used for second problem

- Baum-Welch algorithm used for third problem
  - Finding parameters of the model
  - Aka forward-backward algorithm

Hidden Markov Models

- Recall “Shannon game” (guess next word in a text)
- Particularly important for speech recognition, OCR
- n-gram models commonly used to estimate probabilities of words
  - unigram, bigram, trigram
  - n-gram model is equivalent to an (n-1)th order Markov model
- Estimates must be smoothed by, for example, interpolating combinations of n-gram estimates

$$P(w_n \mid w_{n-1}, w_{n-2}) = \lambda_1 P_1(w_n) + \lambda_2 P_2(w_n \mid w_{n-1}) + \lambda_3 P_3(w_n \mid w_{n-1}, w_{n-2})$$

- HMM algorithms can determine the optimal parameter settings