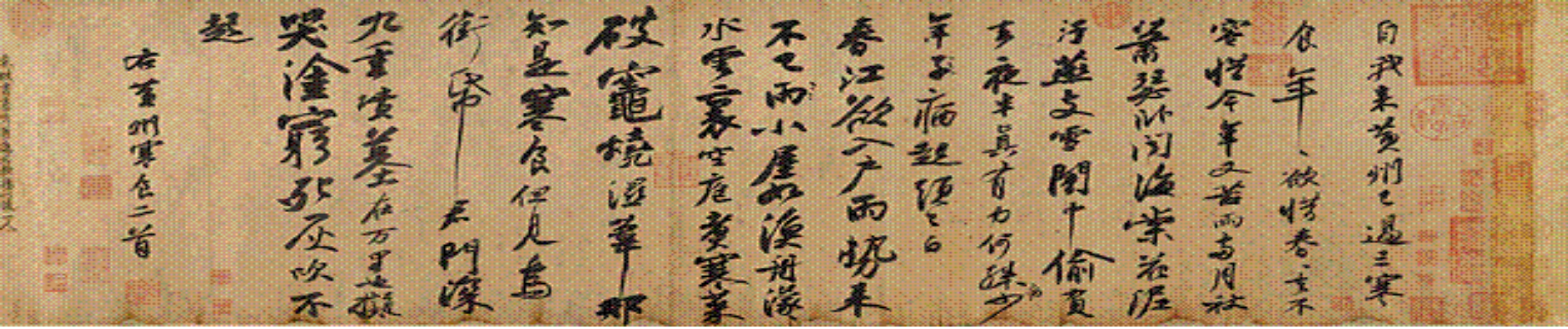


计算语言学
Computational Linguistics

教师：孙茂松

Tel:62781286

Email:sms@tsinghua.edu.cn

TA：**林衍凯**

Email:linyankai423@qq.com

郑重声明

- 此课件仅供选修清华大学计算机系研究生课《计算语言学》(70240052)的学生个人学习使用，所以只允许学生将其下载、存贮在自己的电脑中。未经孙茂松本人同意，任何人不得以任何方式扩散之（包括不得放到任何服务器上）。否则，由此可能引起的一切涉及知识产权的法律责任，概由该人负责。
- 此课件仅限孙茂松本人讲课使用。除孙茂松本人外，凡授课过程中，PTT文件显示此《郑重声明》之情形，即为侵权使用。

第四章
估计句子的概率-Markov模型
（Part 1）

4.1. Rationalism and Empiricism

- *But it must be recognized that the notion "probability of a sentence" is an entirely useless one, under any known interpretation of this term.* Noam Chomsky(1960).

  poverty of the stimulus

  语言的关键部分是天生的

- *Anytime a linguist leaves the group the recognition rate goes up.* Fred Jelinek (IBM speech group, 1988)

18 Nov.1932 - 14 Sept.2010

Frederick Jelinek, Who Gave Machines the Key to Human Speech, Dies at 77

By STEVE LOHR
Published: September 24, 2010

Frederick Jelinek, who survived the Nazi occupation of Czechoslovakia to become a pioneer in computer research in America, helping to make it possible for computers to decipher and translate human speech, died on Sept. 14 in Baltimore. He was 77.

RECOMMEND
TWITTER
LINKEDIN
SIGN IN TO E-MAIL
PRINT
REPRINTS
SHARE

Enlarge This Image

The Johns Hopkins University

Frederick Jelinek

Add to Portfolio

The cause was a heart attack, his son, William, said. Mr. Jelinek was stricken while he was in his office at Johns Hopkins University, where he was a professor. He lived in Baltimore and New York.

Today, computerized speech recognition is becoming a mainstream technology. A few words spoken into a smartphone can summon an Internet search; doctors use voice-transcription software for patient records; drivers talk to speech-recognition systems in cars that reply with driving directions; and customer questions to call centers are increasingly being answered by automated speech systems.

4.1. Rationalism and Empiricism

- Through "statistical magic," computers could be made to distinguish between the almost identical sounds of "wreck a nice beach" and those of "recognize speech."

- linguistic competence vs. linguistic performance
  I-language vs. E-language

- 语言中的非绝对化现象：合乎语法与不合乎语法？
  toy system

4.1. Rationalism and Empiricism

- 歧义难以处理

  List the sales of the products produced in 1973 with the products produced in 1972. (Martin et al. 1987)

  455种可能的parse

- Selectional restriction

  当遇到比喻修辞时，变得非常脆弱

  她跳着蹦着出来了。

  她的心快蹦出来了。

- 语言的统计特性

  我吃了一个红_____。

4.2. Estimating Probability of Sentences

- For any sentence $S = w_1, ..., w_t$

$$PROB(S) = PROB(w_1, ..., w_t)$$

$$= PROB(w_1, ..., w_{t-1}) \times PROB(w_t \mid w_1, ..., w_{t-1})$$

$$= PROB(w_1, ..., w_{t-2}) \times PROB(w_{t-1} \mid w_1, ..., w_{t-2}) \times PROB(w_t \mid w_1, ..., w_{t-1})$$

$$= .....$$

$$= PROB(w_1, w_2) \times PROB(w_3 \mid w_1, w_2) \times ..... \times PROB(w_{t-1} \mid w_1, ..., w_{t-2}) \times PROB(w_t \mid w_1, ..., w_{t-1})$$

$$= PROB(w_1) \times PROB(w_2 \mid w_1) \times PROB(w_3 \mid w_1, w_2) \times ..... \times PROB(w_{t-1} \mid w_1, ..., w_{t-2}) \times PROB(w_t \mid w_1, ..., w_{t-1})$$

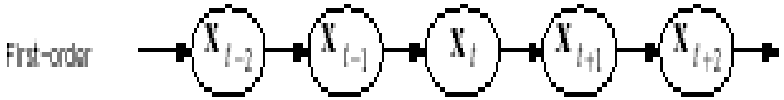
4.2. Estimating Probability of Sentences

n-gram models:

unigram(The 0 order Markov model): $PROB(w_i)$

bigram(The first oder Markov Model): $PROB(w_i \mid w_{i-1})$

trigram(The second order Markov Model): $PROB(w_i \mid w_{i-1}, w_{i-2})$



Using unigram approximation:

$$PROB(w_1,.....,w_t) \cong PROB(w_1) \times PROB(w_2) \times ..... \times PROB(w_t)$$
$$= \prod_{i=1,t} PROB(w_i)$$

Using bigram approximation:

$$PROB(w_1,.....,w_t) \cong PROB(w_1) \times PROB(w_2 \mid w_1) \times ..... \times PROB(w_t \mid w_{t-1})$$
$$= PORB(w_1) \prod_{i=2,t} PROB(w_i \mid w_{i-1})$$

4.2. Estimating Probability of Sentences

1. Unigram approximation to Shakespeare

(a) To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have

(b) Every enter now severally so, let

(c) Hill he late speaks; or! a more to leg less first you enter

(d) Will rash been and by I the me loves gentle me not slavish page, the and hour; ill let

(e) Are where exeunt and sighs have rise excellency took of.. Sleep knave we. near; vile like

2. Bigram approximation to Shakespeare

(a) What means, sir. I confess she? then all sorts, he is trim, captain.

(b) Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow.

(c) What we, hath got so she that I rest and sent to scold and nature bankrupt, nor the first gentleman?

(d) Enter Menenius, if it so many good direction found'st thou art a strong upon command of fear not a liberal largess given away, Falstaff! Exeunt

(e) Thou whoreson chops. Consumption catch your dearest friend, well, and I know where many mouths upon my undoing all but be, how soon, then; we'll execute upon my love's bonds and we do you will?

(f) The world shall- my lord!

4.2. Estimating Probability of Sentences

3. Trigram approximation to Shakespeare

   (a) Sweet prince, Falstaff shall die. Harry of Monmouth's grave.
   (b) This shall forbid it should be branded, if renown made it empty.
   (c) What is't that cried?
   (d) Indeed the duke; and had a very good friend.
   (e) Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.
   (f) The sweet! How many then shall posthumus end his miseries.

4. Quadrigram approximation to Shakespeare

   (a) King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in;
   (b) Will you not tell me who I am?
   (c) It cannot be but so.
   (d) Indeed the short and the long. Marry, 'tis a noble Lepidus.
   (e) They say all lovers swear more performance than they are wont to keep obliged faith unforfeited!
   (f) Enter Leonato's brother Antonio, and the rest, but seek the weary beds of people sick.

4.2. Estimating Probability of Sentences

Example: calculate the probability of the sentence

"prepare for leap in the dark"

? "prepare for lip in the dark" (**Speech Recognition**)

BNC Corpus: 100,106,008 words

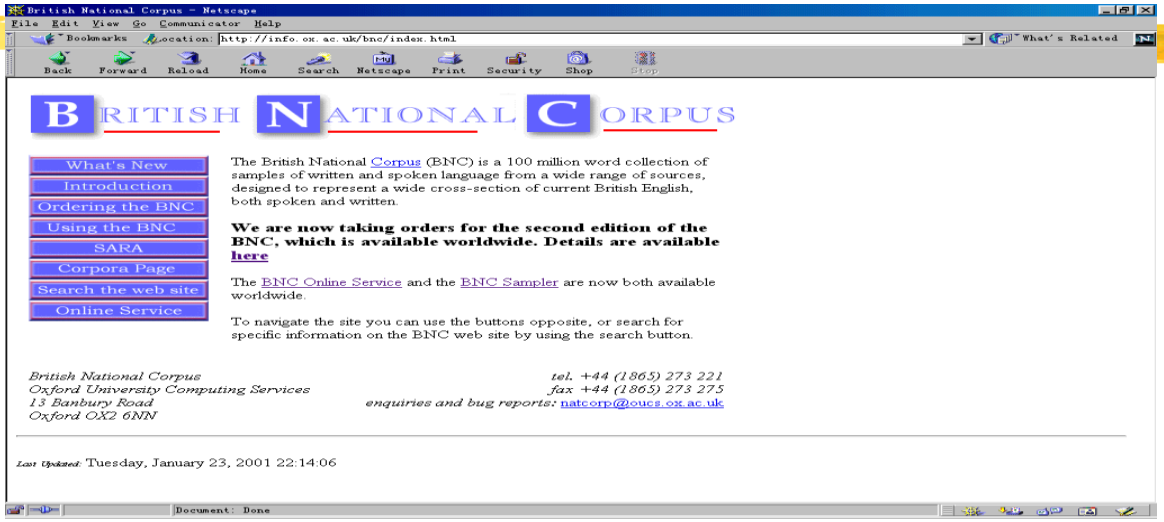
COUNT(prepare)=3023
COUNT(leap)=1045
COUNT(the)=2165569
COUNT(lip)=1592

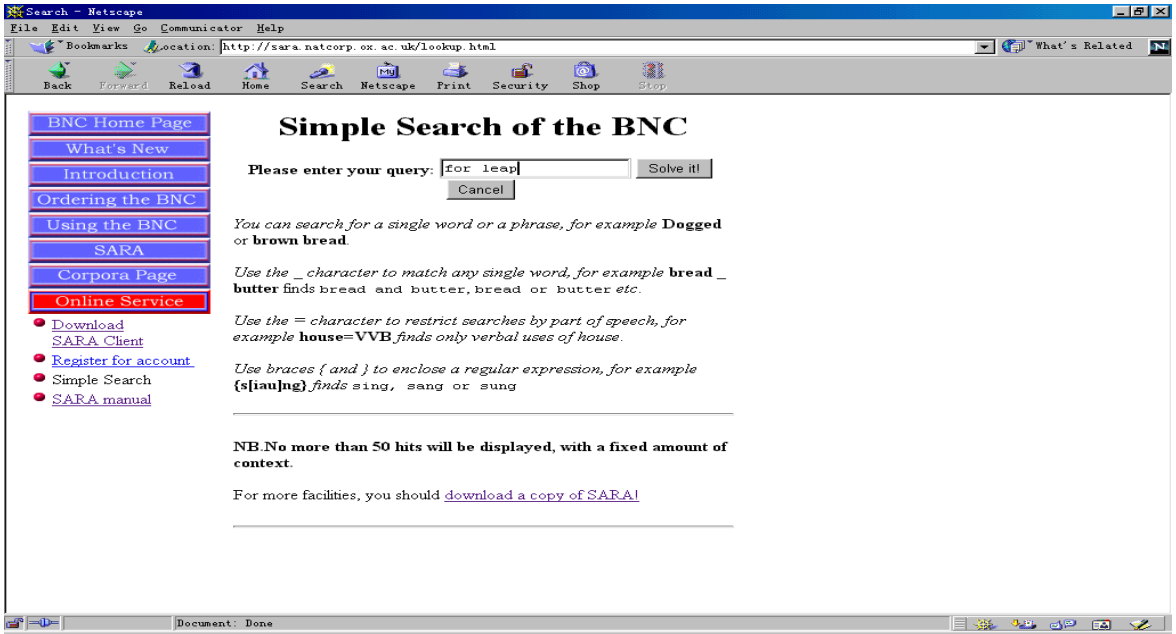
COUNT(for)=899331
COUNT(in)=1970532
COUNT(dark)=13489

COUNT(prepare for)=528
COUNT(leap in)=100
COUNT(the dark)=3668
COUNT(lip in)=25

COUNT(for leap)=1
COUNT(in the)=535036
COUNT(for lip)=2

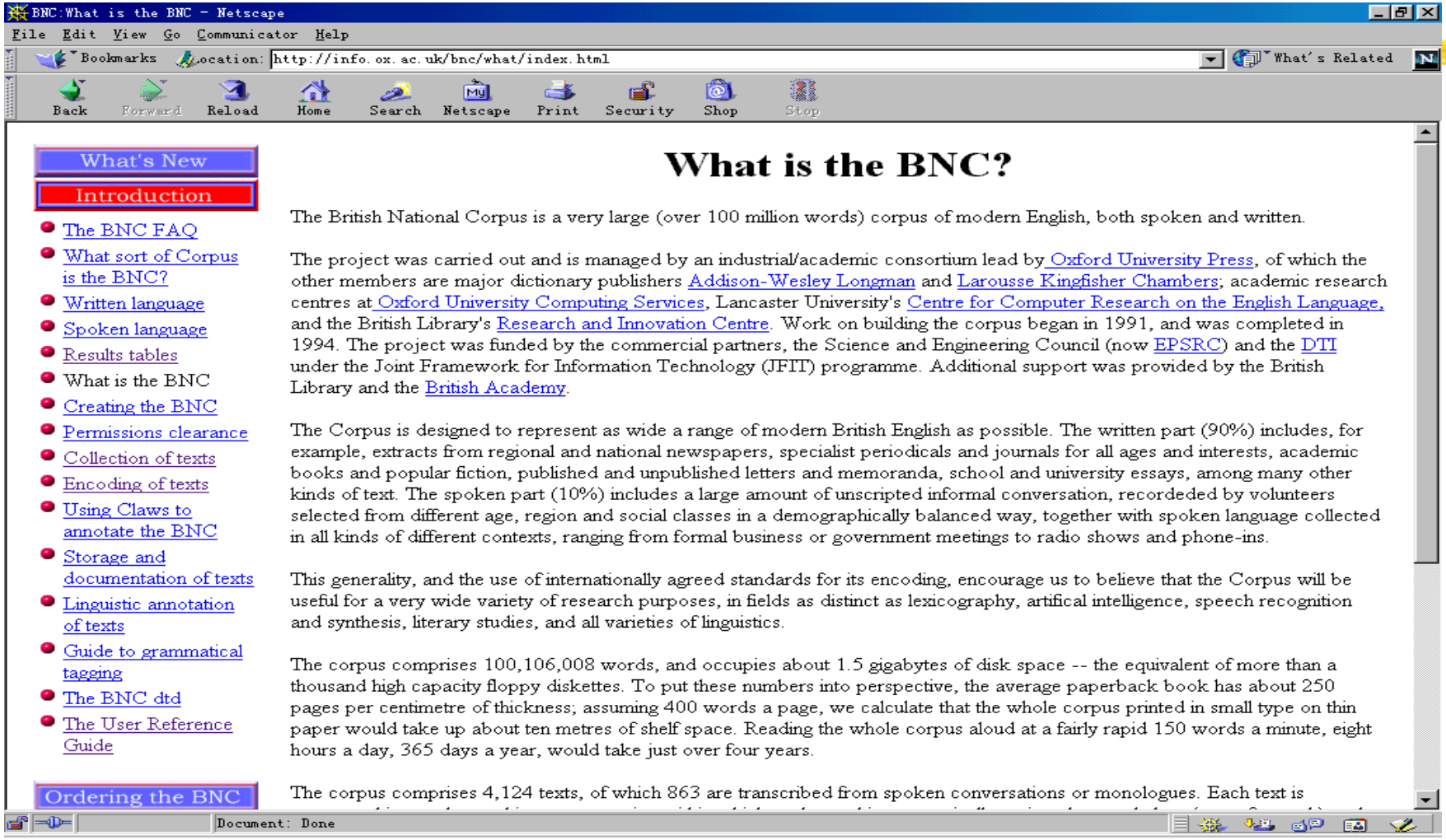
4.2. Estimating Probability of Sentences





4.2. Estimating Probability of Sentences

http://info.ox.ac.uk/bnc/index.html



http://www.natcorp.ox.ac.uk/corpora.html

English Language Corpora and Corpus resources

4.2. Estimating Probability of Sentences

unigram probabilities:

PROB(prepare)= 0.000030 PROB(for)=0.0090
PROB(leap)=0.00001 PROB(in)=0.020
PROB(the)=0.022 PROB(dark)=0.00013
PROB(lip)=0.000016

bigram probabilities:

PROB(prepare for)= 0.0000053 PROB(for leap)=0.00000001
PROB(leap in)=0.000001 PROB(in the)= 0.0053
PROB(the dark)=0.000037 PROB(for lip)= 0.00000002
PROB(lip in)= 0.00000025

$$PROB(for \mid prepare) = \frac{\Pr ob(prepare, for)}{\Pr ob(prepare)} = \frac{Count(prepare, for)/N}{Count(prepare)/N} =$$

$$\frac{Count(prepare, for)}{Count(prepare)} = \frac{528}{3023} = 0.17$$

4.2. Estimating Probability of Sentences

bigram probabilities:

PROB(for|prepare)=0.17
PROB(leap|for)=0.0000011
PROB(in|leap)=0.096
PROB(the|in)=0.27
PROB(dark|the)=0.0017
PROB(lip|for)=0.0000022
PROB(in|lip)=0.016

4.2. Estimating Probability of Sentences

S1= “prepare for leap in the dark”
S2= “prepare for lip in the dark”

unigram:

$$PROB(S_1) = PROB(prepare) \times PROB(for) \times PROB(leap) \times PROB(in)$$
$$PROB(the) \times PROB(dark)$$

=0.000030*0.0090*0.00001*0.02*0.022*0.00013
=1.54*e-19

$$PROB(S_2) = PROB(prepare) \times PROB(for) \times PROB(lip) \times PROB(in)$$
$$PROB(the) \times PROB(dark)$$

=0.000030*0.0090*0.000016*0.02*0.022*0.00013
=2.46*e-19

bigram:

$$PROB(S_1) = PROB(prepare) \times PROB(for \mid prepare) \times PROB(leap \mid for)$$
$$PROB(in \mid leap) \times PROB(the \mid in) \times PROB(dark \mid the)$$

=0.000030* 0.17*0.0000011*0.096*0.27*0.0017
=2.47*e-16

$$PROB(S_2) = PROB(prepare) \times PROB(for \mid prepare) \times PROB(lip \mid for)$$
$$PROB(in \mid leap) \times PROB(the \mid in) \times PROB(dark \mid the)$$

=0.000030* 0.17*0.0000022*0.016*0.27*0.0017
=8.24*e-17

4.2. Estimating Probability of Sentences

Observations:

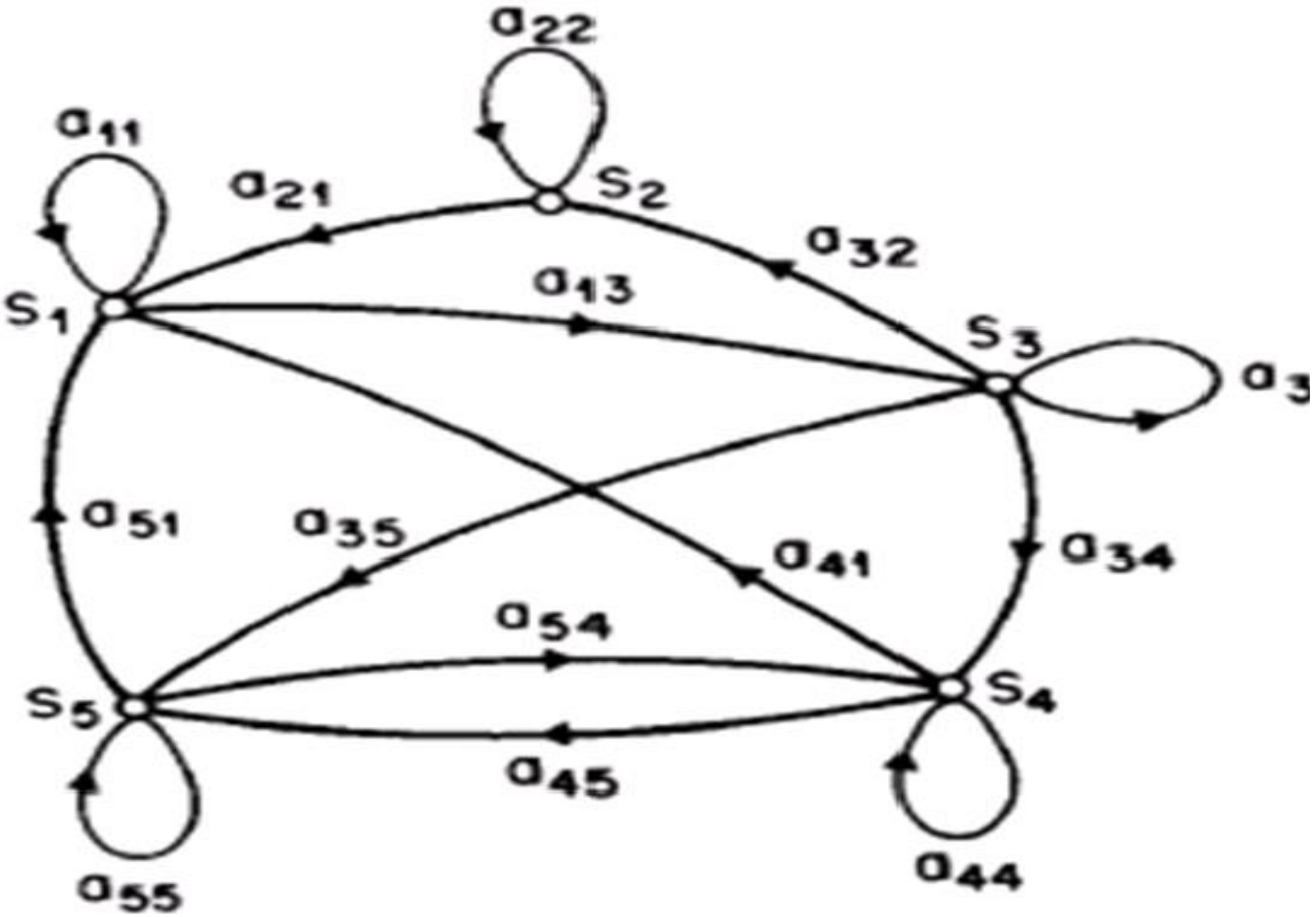
(1) $PROB_{unigram}(S1) < PROB_{unigram}(S2)$
$PROB_{bigram}(S1) > PROB_{bigram}(S2)$

The bigram model is more powerful than unigram model.

(2) $PROB_{bigram}(S1)/PROB_{unigram}(S1)=1604$
$PROB_{bigram}(S2)/PROB_{unigram}(S2)=340$

The estimation of bigram is more accurate than that of unigram.

4.2. Estimating Probability of Sentences



4.2. Estimating Probability of Sentences

Number of parameters to be estimated:

Size of lexicon: L

(1) unigram model: L (假设一个中型词典: 5万词)

word	dark	for	In	leap	lip	prepare	the
count	13489	899331	1970532	1045	1592	3023	2165569

(2) bigram model: L*L (5万*5万=25亿个参数)

co-occurrence count	dark	for	in	leap	lip	prepare	the
dark	6	67	106	0	0	0	17
for	15	397	1880	1	2	1	166570
in	203	2854	1832	3	0	0	535036
leap	0	12	100	0	0	0	9
lip	0	12	25	0	0	0	2
prepare	0	528	13	0	0	1	394
the	3668	122	503	50	133	1	604

(3) trigram model: L*L*L (5万*5万*5万=125万亿个参数)

4.2. Estimating Probability of Sentences

* How big is the lexicon V?
* Heaps' law: $M = kT^b$
* M is the size of the vocabulary, T is the number of tokens in the collection
* Typical values: $30 \leq k \leq 100$ and $b \approx 0.5$
* In a log-log plot of vocabulary size M vs. T, Heaps' law predicts a line with slope about ½

  An empirical finding ("empirical law")

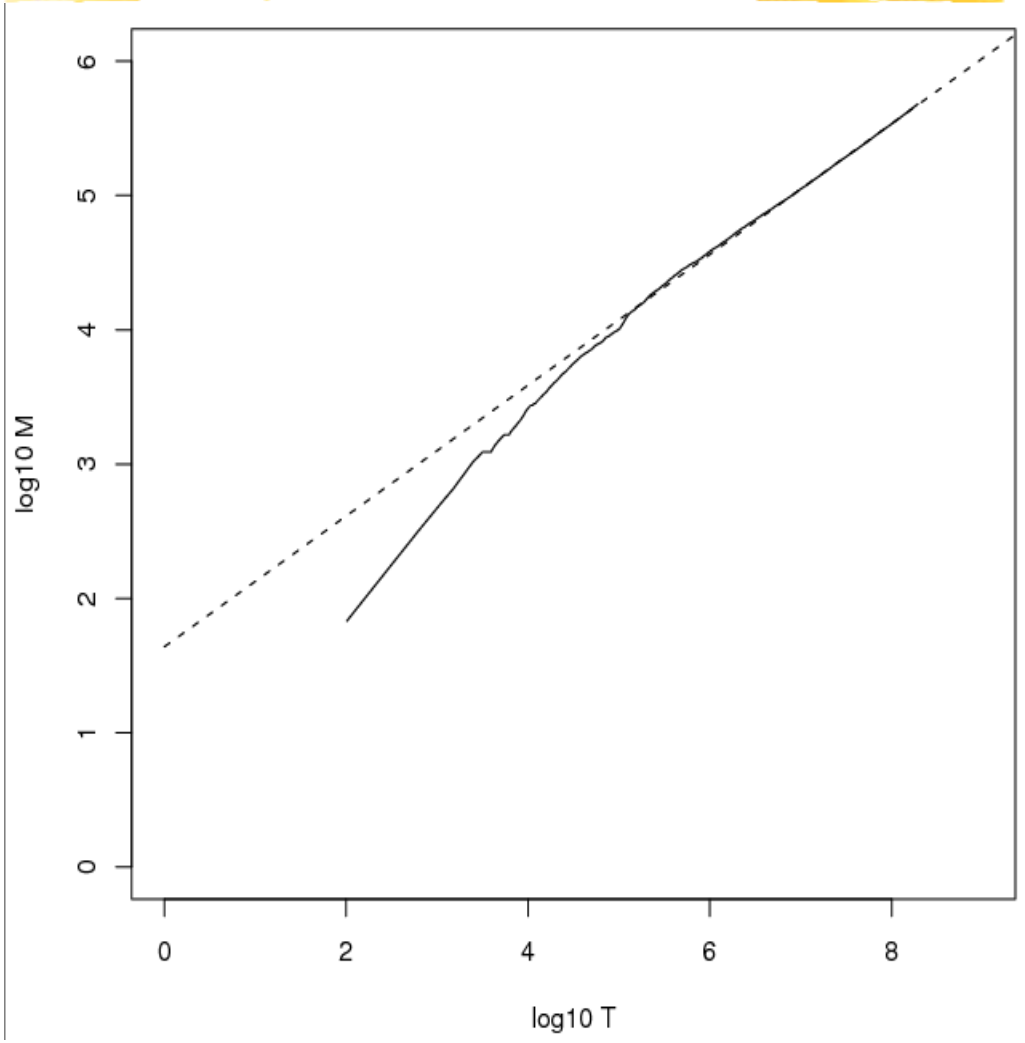
4.2. Estimating Probability of Sentences

For Reuters RCV1, the dashed line

$\log_{10} M = 0.49 \log_{10} T + 1.64$ is the best least squares fit.

Thus, $M = 10^{1.64} T^{0.49}$ so $k = 10^{1.64} \approx 44$ and $b = 0.49$.

For first 1,000,020 tokens, law predicts 38,323 terms; actually, 38,365 terms



4.2. Estimating Probability of Sentences

- Word types and word tokens
- frequency of word types (distinct words)

- Zipf's law: the Harvard linguist

  George Kingsley Zipf (1902-1950)

  Frequency * rank ≈ constant

4.2. Estimating Probability of Sentences

Zipf's law: If the terms in a collection are ranked (*r*) by their frequency (*f_r*), they roughly fit the relation *r* * *f_r* ≈ *C*. Different collections have different constants *C*, but in English text, *C* tends to be about *N / 10*, where *N* is the number of words in the collection.

p_r = *f_r / N* is the probability that a randomly chosen term (with frequency *f_r*) will have rank *r*.

r * *p_r* = *A*, where *A* tends to be about 0.1 in English text.

$$p_r = \frac{f}{N} = \frac{A}{r} \quad \text{for corpus indp. const. } A \approx 0.1$$

4.2. Estimating Probability of Sentences

Statistics from the TIME collection, a 1.6 MB collection of 423 short TIME magazine articles (245,412 term occurrences). Top 50 terms:

Word	f_t	$r_t * p_r$	Word	f_t	$r_t * p_r$	Word	f_t	$r_t * p_r$
the	15861	0.065	it	1290	0.095	week	793	0.113
of	7239	0.059	from	1228	0.095	they	697	0.102
to	6331	0.077	but	1138	0.093	govern	687	0.104
a	5878	0.096	u	955	0.082	all	672	0.104
and	5614	0.114	had	940	0.084	year	672	0.107
in	5294	0.129	last	930	0.087	its	620	0.101
that	2507	0.072	be	915	0.089	britain	89	0.098
for	2228	0.073	have	914	0.093	when	579	0.099
was	2149	0.079	who	894	0.095	out	577	0.101
with	1839	0.075	not	882	0.097	would	577	0.103
his	1815	0.081	has	880	0.100	new	572	0.105
is	1810	0.089	an	873	0.103	up	559	0.105
he	1700	0.090	s	865	0.106	been	554	0.106
as	1581	0.090	were	848	0.107	more	540	0.106
on	1551	0.095	their	815	0.106	which	539	0.108
by	1467	0.096	are	812	0.109	into	518	0.106
at	1333	0.092	one	811	0.112			

4.2. Estimating Probability of Sentences

Statistics from the WSJ87 collection, a 131.6 MB collection of 46,449 newspaper articles (19 million term occurrences). Top 50 terms:

Word	f_t	$r_t * p_r$	Word	f_t	$r_t * p_r$	Word	f_t	$r_t * p_r$
the	1130021	0.059	from	96900	0.092	or	54958	0.101
of	547311	0.058	he	94585	0.095	about	53713	0.102
to	516635	0.082	million	3515	0.098	market	52110	0.101
a	464736	0.098	year	90104	0.100	they	51359	0.103
in	390819	0.103	its	86774	0.100	this	50933	0.105
and	387703	0.122	be	85588	0.104	would	50828	0.107
that	204351	0.075	was	83398	0.105	u	49281	0.106
for	199340	0.084	company	3070	0.109	which	48273	0.107
is	152483	0.072	an	76974	0.105	bank	47940	0.109
said	148302	0.078	has	74405	0.106	stock	47401	0.110
it	134323	0.078	are	74097	0.109	trade	47310	0.112
on	121173	0.077	have	73132	0.112	his	47116	0.114
by	118863	0.081	but	71887	0.114	more	46244	0.114
as	109135	0.080	will	71494	0.117	who	42142	0.106
at	101779	0.080	say	66807	0.113	one	41635	0.107
mr	101679	0.086	new	64456	0.112	their	40910	0.108
with	101210	0.091	share	63925	0.114			

4.2. Estimating Probability of Sentences

Does Real Data Fit Zipf's Law?

A law of the form $y = kx^c$ is called a power law.

Zipf's law is a power law with $c = -1$

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

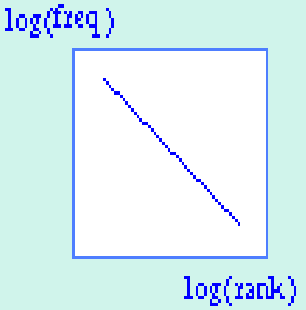
$$\log(y) = \log(kx^c) = \log k + c\log(x)$$

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

4.2. Estimating Probability of Sentences

"Principle of least effort"

ZIPF'S LAW



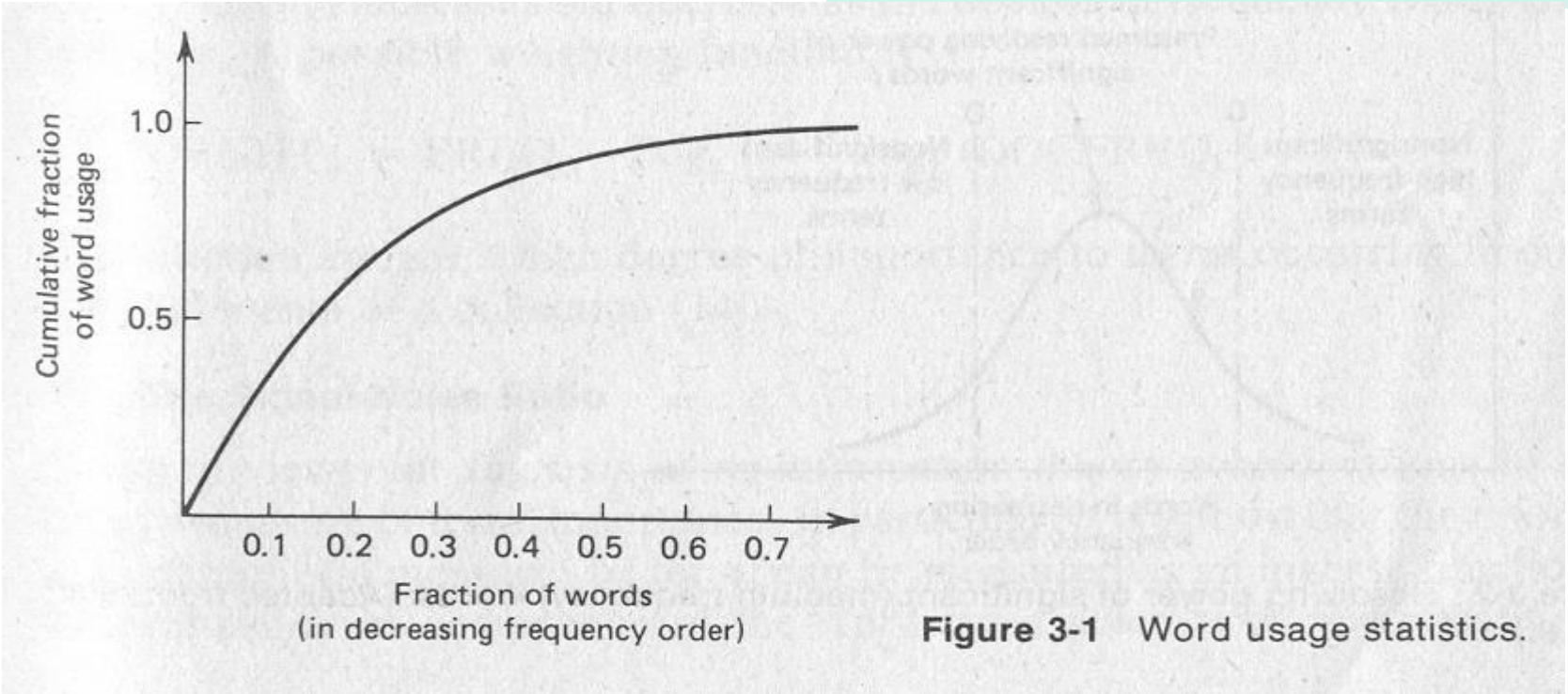
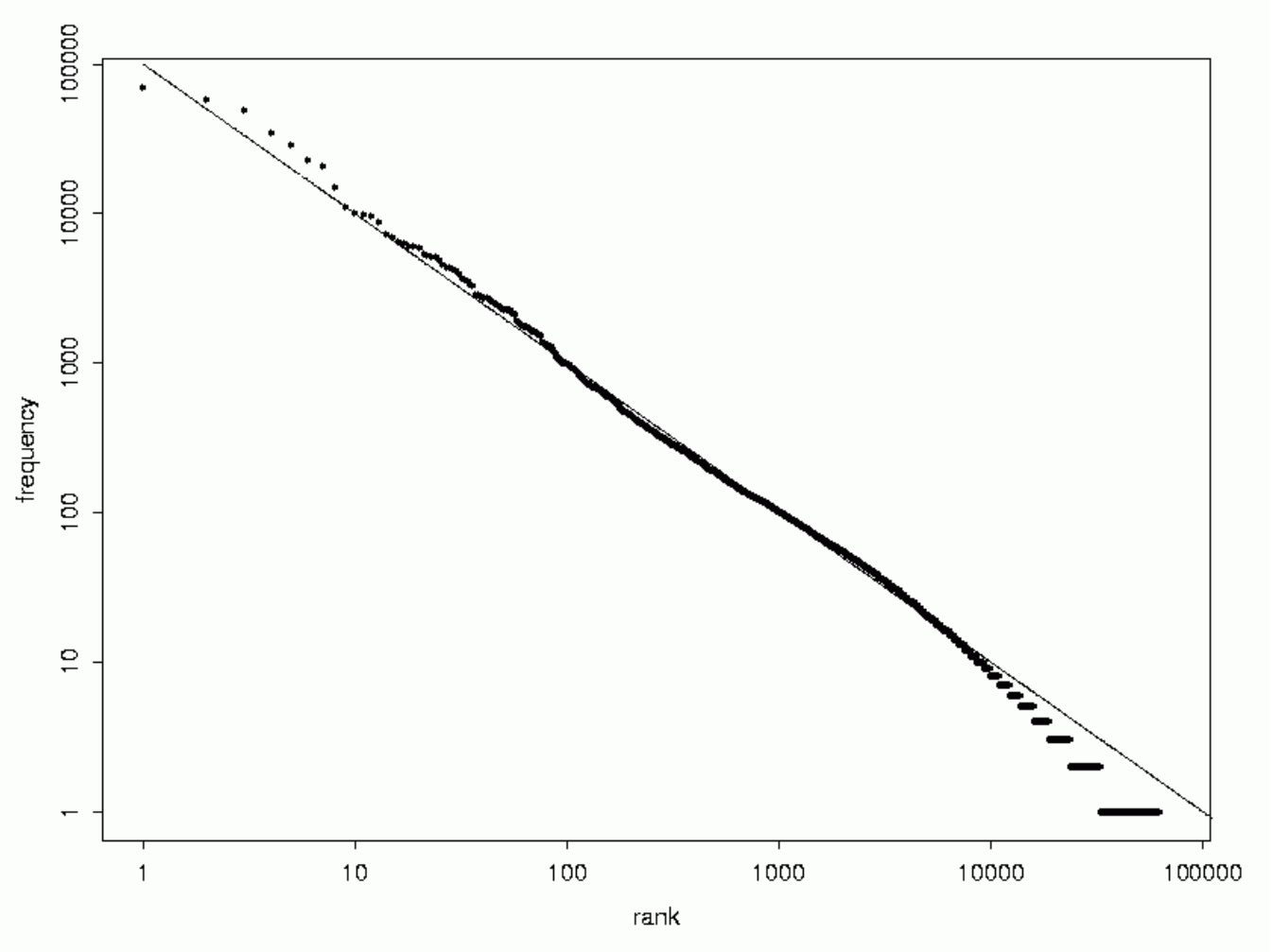


Figure 3-1 Word usage statistics.

4.2. Estimating Probability of Sentences

Zipf's law log-log plot



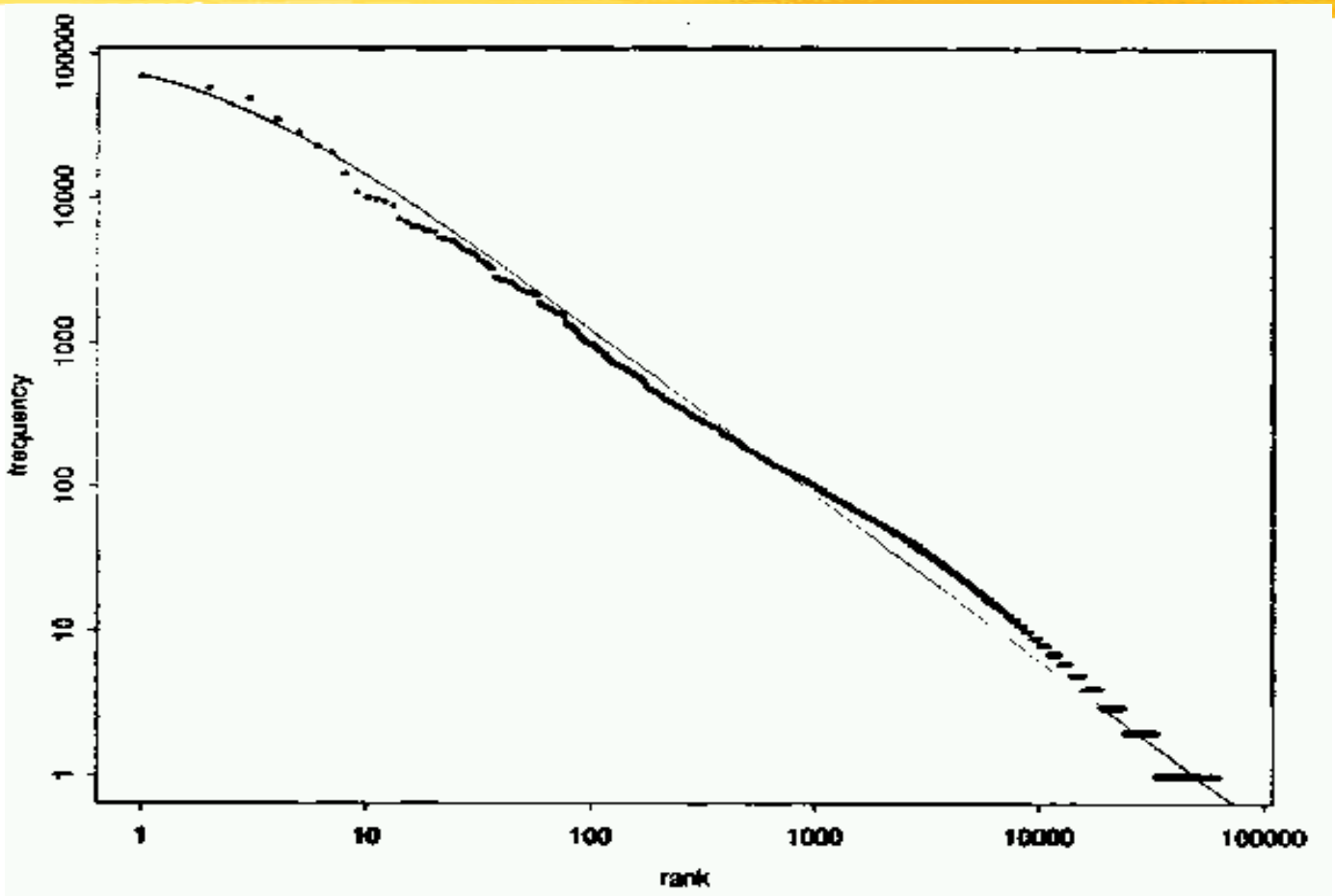
4.2. Estimating Probability of Sentences

Mandelbrot (1954) Correction

The following more general form gives a bit better fit:

$$f = P(r + \rho)^{-B} \qquad \text{For constants } P, B, \rho$$

4.2. Estimating Probability of Sentences



Mandelbrot's function on Brown corpus

$P = 10^{5.4}$, $B = 1.15$, $\rho = 100$

4.2. Estimating Probability of Sentences

Predicting Word Frequencies

By Zipf's Law, a word appearing *n* times has rank $r_n = AN/n$

Several words may occur *n* times, assume rank r_n applies to the last of these.

Therefore, r_n words occur *n* or more times and r_{n+1} words occur $n+1$ or more times.

So, the number of words appearing **exactly** *n* times is:

$$I_n = r_n - r_{n+1} = \frac{AN}{n} - \frac{AN}{n+1} = \frac{AN}{n(n+1)}$$

4.2. Estimating Probability of Sentences

Predicting Word Frequencies

Assume highest ranking term occurs once and therefore has rank $D = AN/1$

Fraction of words with frequency n is:

$$\frac{I_n}{D} = \frac{1}{n(n+1)}$$

Fraction of words appearing only once is therefore ½.

4.2. Estimating Probability of Sentences

Word Frequency Data (from B. Croft, UMass)

Number of Occurrences (n)	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
125,720,891 total word occurrences; 508,209 unique words

4.2. Estimating Probability of Sentences

Chinese Word segmentation:

名人读书法
名人 读 书法
名人 读书 法

名人读书法书
名人 读 书法 书
名人 读书 法 书

<u>失效:</u>
美女飞行员
美女 飞行员
美 女飞行员

SIGHAN

Second International Chinese Word Segmentation Bakeoff

Data

The Second International Chinese Word Segmentation Bakeoff took place over the summer of 2005 and the results were presented at the 4th SIGHAN Workshop, held at IJCNLP'05, October 14-15.

Corpora from the following organizations were used:

- CKIP, Academia Sinica, Taiwan
- City University of Hong Kong, Hong Kong SAR
- Beijing Universty, China
- Microsoft Research, China

The complete training, testing, and gold-standard data sets, as well as the scoring script, are available for research use:

- icwb2-data.rar [40 MB, md5]
- icwb2-data.zip [50 MB, md5]
- icwb2-data.tar.bz2 [37MB, md5]

The Detailed Instructions for the bakeoff are available. Please read them carefully.

Segmentation guidelines for the following corpora are available. These were supplied to SIGHAN by each data provider, and converted into PDF by the organizer:

Corpus	MS Word	PDF
Academia Sinica	516 KB	336 KB
City University of Hong Kong	154 KB	237 KB
Peking University	177 KB	294 KB
Microsoft Research	41 KB	70 KB

http://acl.ldc.upenn.edu/I/I05/I05-3017.pdf

Corpus	Abbrev.	Encodings	Training Size (Words/Types)	Test Size (Words/Types)
Academia Sinica (Taipei)	AS	Big Five Plus, Unicode	5.45M / 141K	122K / 19K
Beijing University	PK	CP936, Unicode	1.1M / 55K	104K / 13K
City University of Hong Kong	CityU	Big Five/HKSCS, Unicode	1.46M / 69K	41K / 9K
Microsoft Research (Beijing)	MSR	CP936, Unicode	2.37M / 88K	107K / 13K

4.3. Case Study: 对联

http://couplet.msra.cn/

上联：苏堤春晓秀
下联：平湖秋月明

上联：沧海一声笑
下联：碧天万里行

上联：预防禽流感
下联：戒备艾滋病

上联：西子盛装迎贵客
下联：南国新月照上宾

上联：月落乌啼霜满天
下联：风吹雁过雨连宵

4.3. Case Study: 对联

上联：西湖论剑吾迎盛世
下联：东海看云何待春风

以上是本在杭州举行的“21世纪的计算”（2005年）学术研讨会上展示时所对

上联：网上购物红红火火
下联：电子商务热热闹闹
横批：质量第一

上联：春夏秋冬风光各异
下联：东西南北气象不同
横批：江山似锦

上联：一号文件迎春到
下联：八方钱财贺岁来
横批：九州同乐

上联：犬护祥和宅
下联：人传平安书
横批：吉星悬宇