Text Statistics

Lecture 6

Text Statistics

- Huge variety of words used in text <u>but</u>
- Many statistical characteristics of word occurrences are predictable
  – e.g., distribution of word counts
- Retrieval models and ranking algorithms depend heavily on statistical properties of words
  – e.g., important words occur often in documents but are not high frequency in collection

Zipf's law tells us

- Head words may take large portion of occurrence, but they are semantically meaningless
  - E.g., the, a, an, we, do, to
- Tail words take major portion of vocabulary, but they rarely occur in documents
  - E.g., dextrosinistral
- The rest is most representative
  - To be included in the controlled vocabulary

Statistical property of language

- Zipf's law

discrete version of power law

  - Frequency of any word is inversely proportional to its rank in the frequency table
  - Formally
    - $f = \frac{C}{k}$

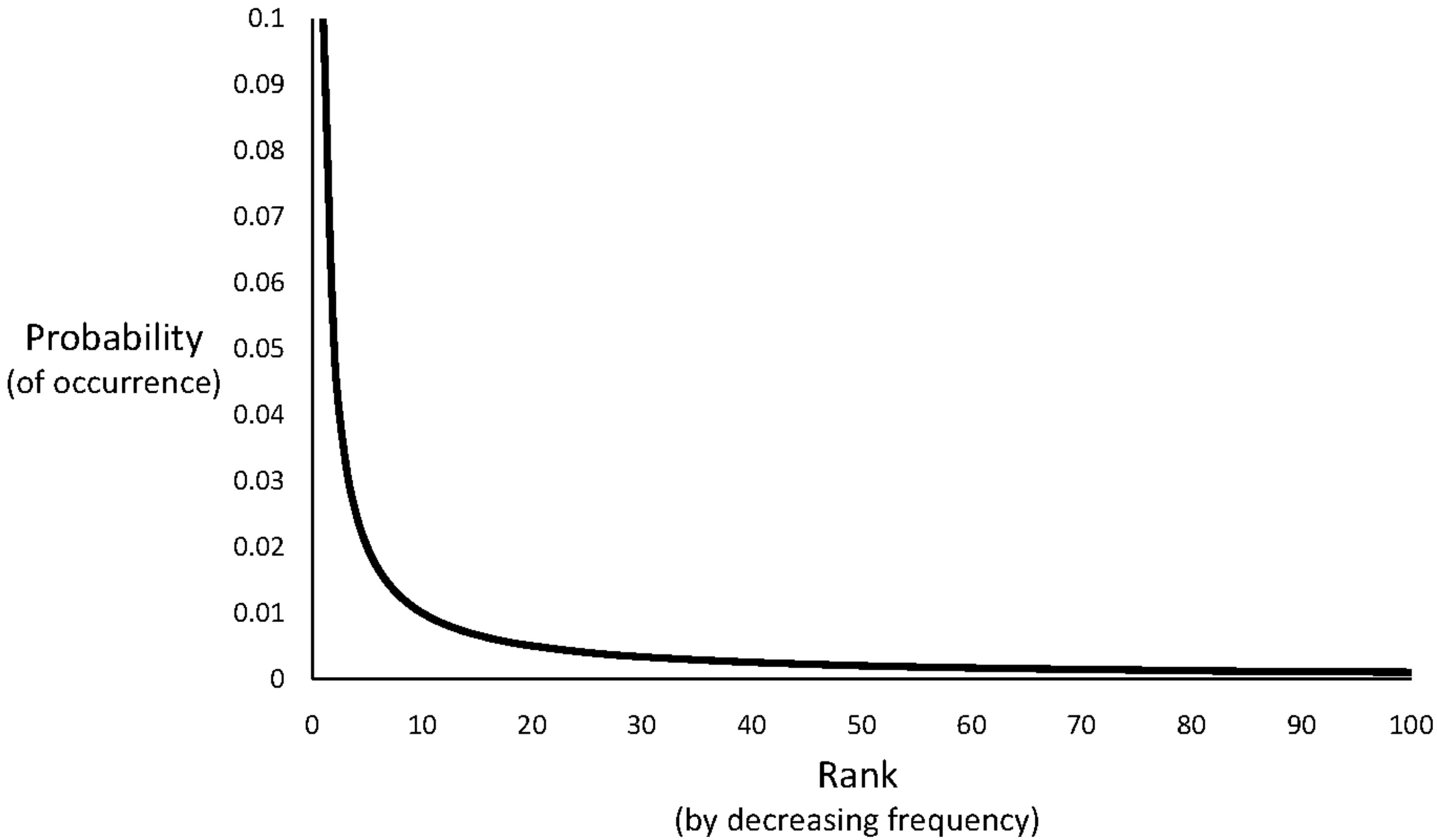
      where k is rank of the word; C is corpus specific constant

*In the Brown Corpus of American English text, the word "the" is the most frequently occurring word, and by itself accounts for nearly **7%** of all word occurrences; the second-place word "of" accounts for slightly over **3.5%** of words.*

Zipf's Law

- Distribution of word frequencies is very *skewed*
  - a few words occur very often, many words hardly ever occur
  - e.g., two most common words ("the", "of") make up about 10% of all word occurrences in text documents
- Zipf's "law":
  - observation that rank (r) of a word times its frequency (f) is approximately a constant (k)
    - assuming words are ranked in order of decreasing frequency
  - i.e., $r.f \approx$ k or $r.P_r \approx c$, where P_r is probability of word occurrence and $c \approx 0.1$ for English

Zipf's Law



0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
Probability
(of occurrence)
0
10
20
30
40
50
60
70
80
90
100
Rank
(by decreasing frequency)

Automatic text indexing

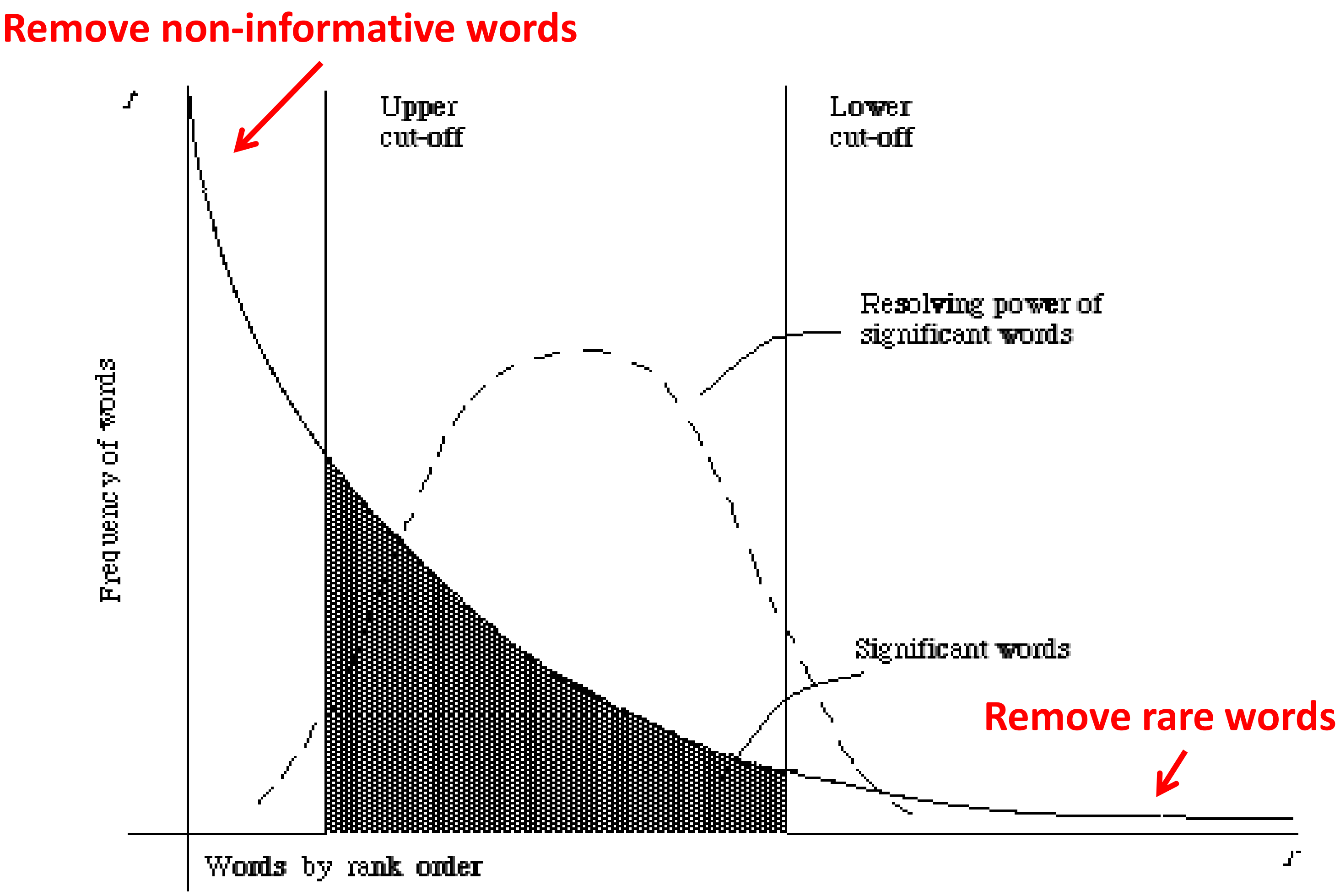


Figure 2.1. A plot of the hyperbolic curve relating f, the frequency of occurrence and r, the rank order (Adaped from Schultz[44] page 120)

News Collection (AP89) Statistics

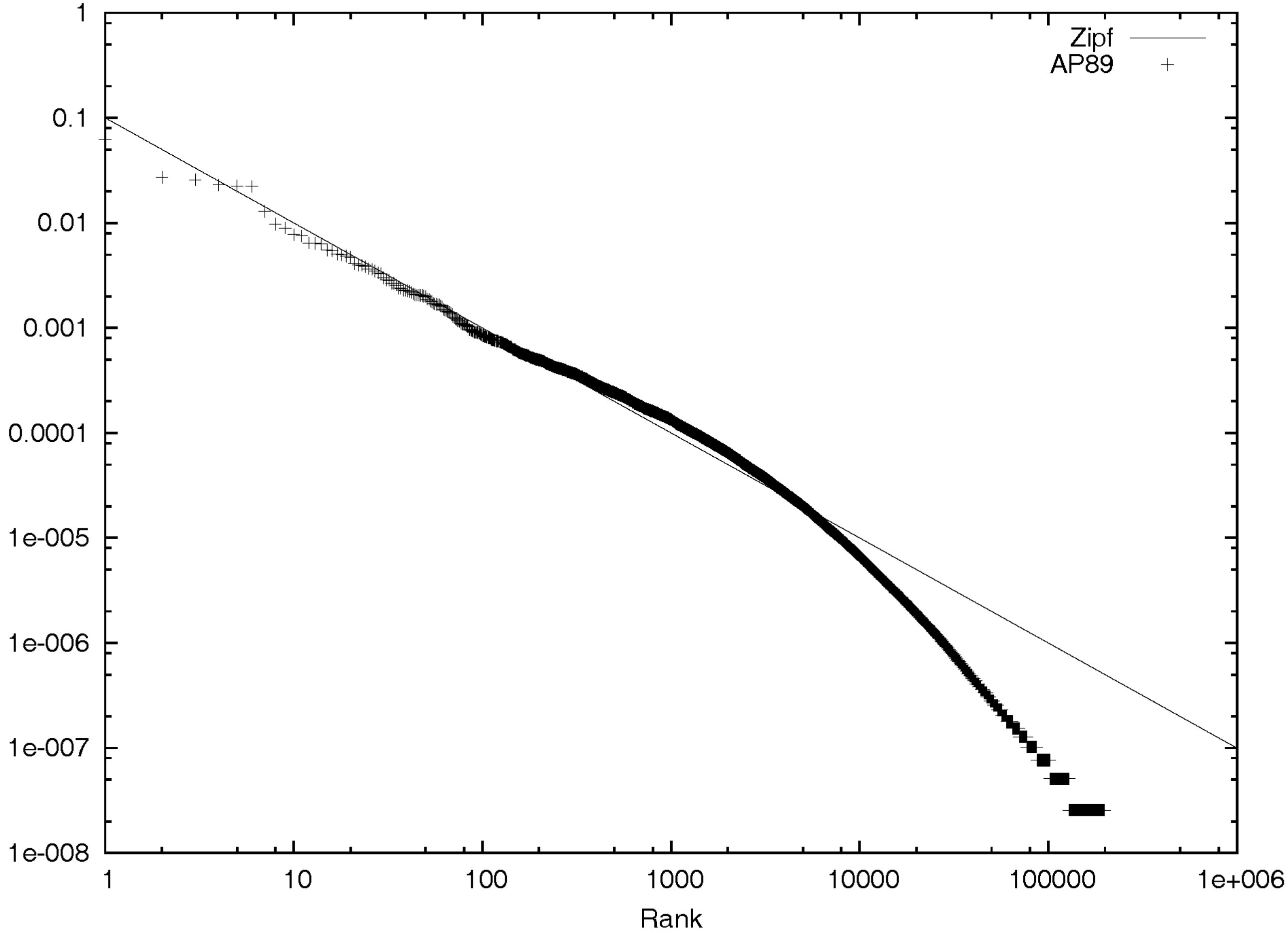
Total documents	84,678
Total word occurrences	39,749,179
Vocabulary size	198,763
Words occurring > 1000 times	4,169
Words occurring once	70,064

Word	*Freq.*	*r*	*Pr(%)*	*r.Pr*
assistant	5,095	1,021	.013	0.13
sewers	100	17,110	$2.56 \times 10-4$	0.04
toothbrush	10	51,555	$2.56 \times 10-5$	0.01
hazmat	1	166,945	$2.56 \times 10-6$	0.04

Top 50 Words from AP89

Word	*Freq.*	r	P_r(%)	$r.P_r$	*Word*	*Freq*	r	P_r(%)	$r.P_r$
the	2,420,778	1	6.49	0.065	has	136,007	26	0.37	0.095
of	1,045,733	2	2.80	0.056	are	130,322	27	0.35	0.094
to	968,882	3	2.60	0.078	not	127,493	28	0.34	0.096
a	892,429	4	2.39	0.096	who	116,364	29	0.31	0.090
and	865,644	5	2.32	0.120	they	111,024	30	0.30	0.089
in	847,825	6	2.27	0.140	its	111,021	31	0.30	0.092
said	504,593	7	1.35	0.095	had	103,943	32	0.28	0.089
for	363,865	8	0.98	0.078	will	102,949	33	0.28	0.091
that	347,072	9	0.93	0.084	would	99,503	34	0.27	0.091
was	293,027	10	0.79	0.079	about	92,983	35	0.25	0.087
on	291,947	11	0.78	0.086	i	92,005	36	0.25	0.089
he	250,919	12	0.67	0.081	been	88,786	37	0.24	0.088
is	245,843	13	0.65	0.086	this	87,286	38	0.23	0.089
with	223,846	14	0.60	0.084	their	84,638	39	0.23	0.089
at	210,064	15	0.56	0.085	new	83,449	40	0.22	0.090
by	209,586	16	0.56	0.090	or	81,796	41	0.22	0.090
it	195,621	17	0.52	0.089	which	80,385	42	0.22	0.091
from	189,451	18	0.51	0.091	we	80,245	43	0.22	0.093
as	181,714	19	0.49	0.093	more	76,388	44	0.21	0.090
be	157,300	20	0.42	0.084	after	75,165	45	0.20	0.091
were	153,913	21	0.41	0.087	us	72,045	46	0.19	0.089
an	152,576	22	0.41	0.090	percent	71,956	47	0.19	0.091
have	149,749	23	0.40	0.092	up	71,082	48	0.19	0.092
his	142,285	24	0.38	0.092	one	70,266	49	0.19	0.092
but	140,880	25	0.38	0.094	people	68,988	50	0.19	0.093

Zipf's Law for AP89



- Note problems at high and low frequencies

Vocabulary Growth

- As corpus grows, so does vocabulary size
  - Fewer new words when corpus is already large
- Observed relationship (*Heaps' Law*):

$$v = k.n^{\beta}$$

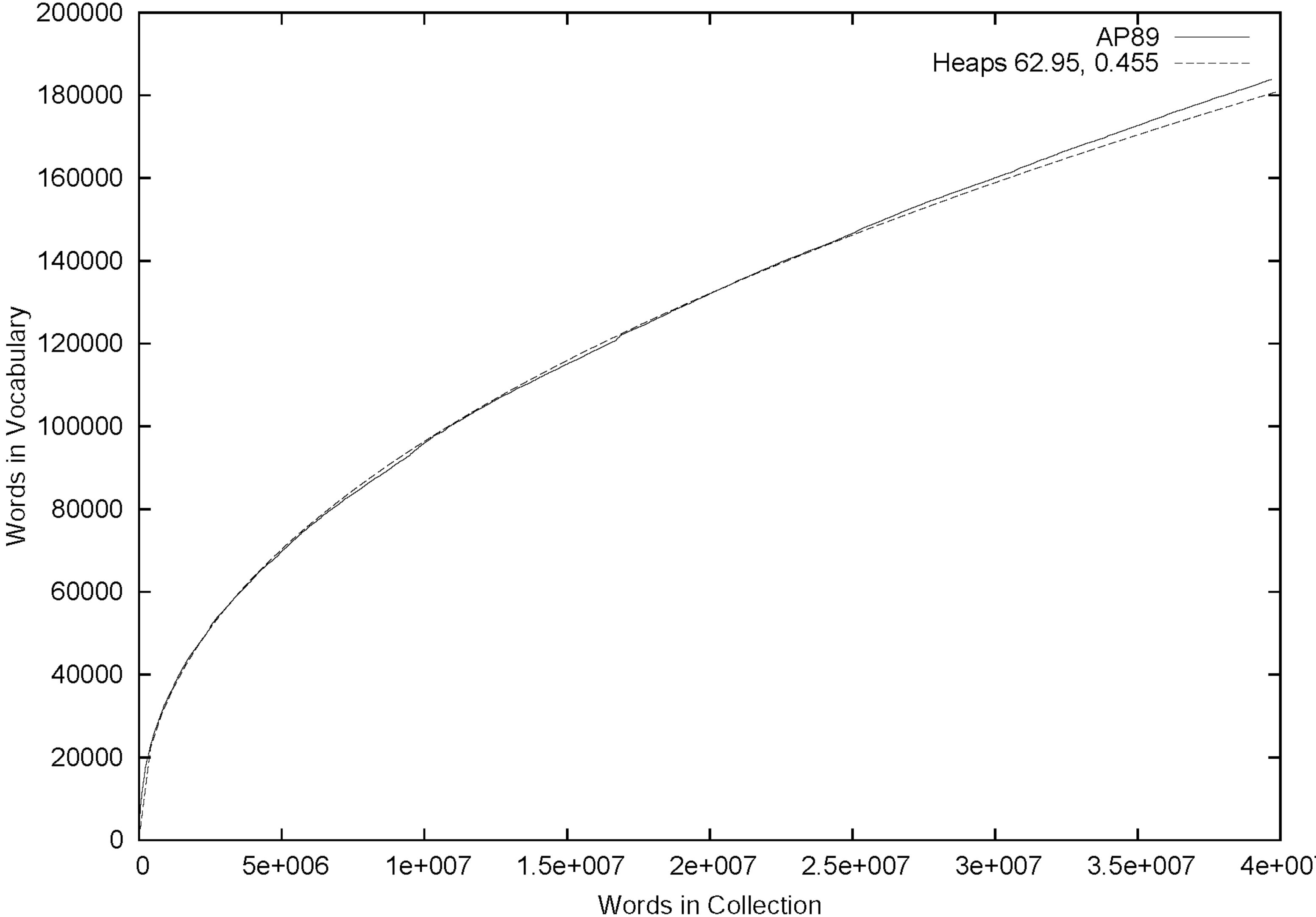
where v is vocabulary size (number of unique words),

n is the number of words in corpus,

k, β are parameters that vary for each corpus

(typical values given are $10 \le k \le 100$ and $\beta \approx 0.5$)

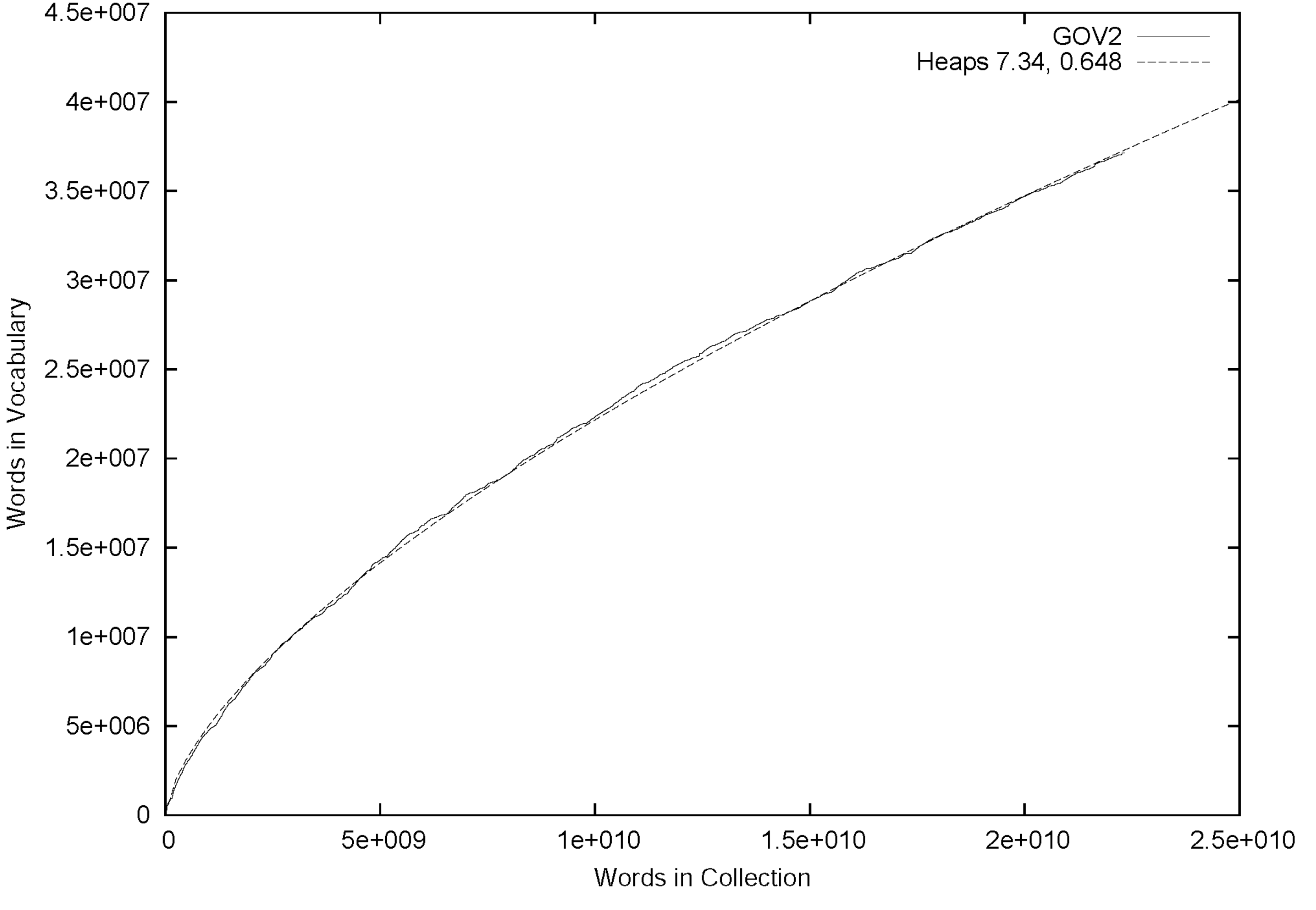
AP89 Example



Heaps' Law Predictions

- Predictions for TREC collections are accurate for large numbers of words
  - e.g., first 10,879,522 words of the AP89 collection scanned
  - prediction is 100,151 unique words
  - actual number is 100,024
- Predictions for small numbers of words (i.e. < 1000) are much worse

GOV2 (Web) Example



Text Normalization

- Every NLP task needs to do text normalization:
  1. Segmenting/tokenizing words in running text
  2. Normalizing word formats
  3. Segmenting sentences in running text

How many words?

- I do uh main- mainly business data processing
  - Fragments, filled pauses
- Seuss's cat in the hat is different from other cats!
  - **Lemma**: same stem, part of speech, rough word sense
    - cat and cats = same lemma
  - **Wordform**: the full inflected surface form
    - cat and cats = different wordforms

Issues in Tokenization

- Finland’s capital → Finland Finlands Finland’s ?
- what’re, I’m, isn’t → What are, I am, is not
- Hewlett-Packard → Hewlett Packard ?
- state-of-the-art → state of the art ?
- Lowercase → lower-case lowercase lower case ?
- San Francisco → one token or two?
- m.p.h., PhD. → ??

Tokenization: language issues

- French
  - ***L'ensemble*** $\rightarrow$ one token or two?
    - ***L*** ? ***L'*** ? ***Le*** ?
    - Want ***l'ensemble*** to match with ***un ensemble***
- German noun compounds are not segmented
  - ***Lebensversicherungsgesellschaftsangestellter***
  - 'life insurance company employee'
  - German information retrieval needs **compound splitter**

Tokenization: language issues

- Chinese and Japanese no spaces between words:
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
  - Sharapova now lives in US southeastern Florida
- Further complicated in Japanese, with multiple alphabets intermingled
  - Dates/amounts in multiple formats

Word Tokenization in Chinese

- Also called **Word Segmentation**
- Chinese words are composed of characters
  - Characters are generally 1 syllable and 1 morpheme.
  - Average word is 2.4 characters long.
- Standard baseline segmentation algorithm:
  - Maximum Matching (also called Greedy)

Max-match segmentation illustration

- Thecatinthehat the cat in the hat
- Thetabledownthere the table down there

- Doesn't generally work in English! theta bled own there

- But works astonishingly well in Chinese
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
- Modern probabilistic segmentation algorithms even better

Stopping

- Function words (determiners, prepositions) have little meaning on their own
- High occurrence frequencies
- Treated as *stopwords* (i.e. removed)
  - reduce index space, improve response time, improve effectiveness
- Can be important in combinations
  - e.g., “to be or not to be”

Stopwords

Nouns

1. time
2. person
3. year
4. way
5. day
6. thing
7. man
8. world
9. life
10. hand
11. part
12. child
13. eye
14. woman
15. place
16. work
17. week
18. case
19. point
20. government
21. company
22. number
23. group
24. problem
25. fact

Verbs

1. be
2. have
3. do
4. say
5. get
6. make
7. go
8. know
9. take
10. see
11. come
12. think
13. look
14. want
15. give
16. use
17. find
18. tell
19. ask
20. work
21. seem
22. feel
23. try
24. leave
25. call

Adjectives

1. good
2. new
3. first
4. last
5. long
6. great
7. little
8. own
9. other
10. old
11. right
12. big
13. high
14. different
15. small
16. large
17. next
18. early
19. young
20. important
21. few
22. public
23. bad
24. same
25. able

Prepositions

1. to
2. of
3. in
4. for
5. on
6. with
7. at
8. by
9. from
10. up
11. about
12. into
13. over
14. after
15. beneath
16. under
17. above

Others

1. the
2. and
3. a
4. that
5. I
6. it
7. not
8. he
9. as
10. you
11. this
12. but
13. his
14. they
15. her
16. she
17. or
18. an
19. will
20. my
21. one
22. all
23. would
24. there
25. their

Stopping

- Stopword list can be created from high-frequency words or based on a standard list
- Lists are customized for applications, domains, and even parts of documents
  - e.g., “click” is a good stopword for anchor text
- Best policy is to index all words in documents, make decisions about which words to use at query time

Normalization

- Convert different forms of a word to normalized form in the vocabulary
  - U.S.A -> USA, St. Louis -> Saint Louis
- Solution
  - Rule-based
    - Delete periods and hyphens
    - All in lower case
  - Dictionary-based
    - Construct equivalent class
      - Car -> “automobile, vehicle”
      - Mobile phone -> “cellphone”

Case folding

- Applications like IR: reduce all letters to lower case
  - Since users tend to use lower case
  - Possible exception: upper case in mid-sentence?
    - e.g., ***General Motors***
    - ***Fed*** vs. ***fed***
    - ***SAIL*** vs. ***sail***
- For sentiment analysis, MT, Information extraction
  - Case is helpful (***US*** versus ***us*** is important)

Morphology

- **Morphemes**:
  - The small meaningful units that make up words
  - **Stems**: The core meaning-bearing units
  - **Affixes**: Bits and pieces that adhere to stems
    - Often with grammatical functions

Stemming

- Stemming is crude chopping of affixes
  - Language dependant
  - E.g., automate(s), automatic, automation all reduce to automat

Stemming

- Many morphological variations of words
  - *Inflectional* (e.g. eats, called, marking, written)
  - *Derivational* (e.g. portable, natural, passage)

Porter Stemmer

- Algorithmic stemmer used in IR experiments since the 70s
- Consists of a series of rules designed to the longest possible suffix at each step
- Produces *stems* not *words*
- Makes a number of errors and difficult to modify

Porter's algorithm
The most common English stemmer

Step 1a

```
sses  →  ss     caresses  →  caress
ies   →  i      ponies    →  poni
ss    →  ss     caress    →  caress
s     →  ø      cats      →  cat
```

Step 1b

```
(*v*)ing  →  ø  walking    →  walk
                sing       →  sing
(*v*)ed   →  ø  plastered  →  plaster
```

Porter's algorithm

Step 2 (for long stems)

```
ational→ ate relational→ relate
izer→ ize       digitizer → digitize
ator→ ate       operator  → operate
```

Step 3 (for longer stems)

```
al     → ø   revival    → reviv
able   → ø   adjustable → adjust
ate    → ∅  activate   → activ
```

Viewing morphology in a corpus

- Given the description you saw on earlier slides, the Porter stemmer would stem the word 'aching' as

A. aching
B. ach
C. ache
D. aches

Viewing morphology in a corpus

- Given the description you saw on earlier slides, the Porter stemmer would stem the word 'aching' as

A. aching
B. ach
C. ache
D. aches

Answer: B