TOKENIZATION AND LANGUAGE MODELING

Manish Shrivastava

Lecture #2

TOKENIZATION

- The task of separating ‘Tokens’ in a given input sentence
- Example
  - “This is my brother’s cat”
  - Should be
  - “ This is my brother’s cat ”
  - Or should it be
  - “ This is my brother ‘ s cat ”

TERMS

- Token : a ‘Token’ is a single surface form word
  - Example :
    - My cat is afraid of other cats .
  - Here, ‘cat’ and ‘cats’ are both tokens
    - Giving a total of 8 tokens in the above sentence
    - ‘ . ’ or any punctuation mark is counted as a token

TERMS

- Type : Type is a vocabulary word
  - A word which might be present in its root form in the dictionary
  - ‘cat’ is the type for both ‘cat’ and ‘cats’
- The set of all types is the vocabulary of a language
- Ques: What would a high token to type ratio tell you about a language ?

TOKENIZATION

- Identifying individual tokens in a given input
  - Types are not of concern at this point
- Ques: Is the task of tokenization difficult?
  - ??

CHALLENGES

- Hyphenation
  - I am a hard-working student .
  - We would deal with the state-of-the-art .
  - I am not going to sho-

CHALLENGES

- Number
  - The value of gravity is 9.8 m/s/s
  - He got 1,000,000 dollars in VC funding .
  - My number is +918451990342
  - Take ½ cups of milk

CHALLENGES

- Dates
  - My birth date is 03/09/1982
  - I joined on 22nd July

CHALLENGES

- Abbreviation
  - Dr. S. P. Kishore is the primary faculty of this course
  - We are in IIIT-H campus

CHALLENGES

- Punctuations
  - This, hands-on experience, is rare in pedagogy .
  - I … uh … not sure how to proceed :-)

CHALLENGES

- URLs
  - The site for course materials for this class is http://tts.iiit.ac.in/~kishore/mediawiki/index.php/NLP:Tokens_and_Words

CHALLENGES

- Sentencification
  - “This is a presentation. It could also be a video”

REGULAR EXPRESSIONS

- Example
  - $ls *.txt
    - All files ending in “.txt”
- Abbreviation
  - “[A-Za-z]+.”
  - “[A-Za-z][A-Za-z]*\.”
    - “ . “ matches any character , hence needs to be escaped to match character “ . “
    - [A-Za-z] matches a single upper- or lower-case character

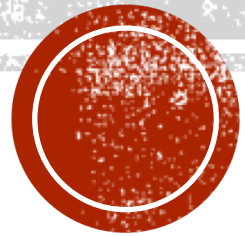
REGULAR EXPRESSIONS

- \s :matches any white-space eg. Tab,newline or space
- \t :tab
- \n :newline
- “^a” :matches strings beginning with letter ‘a’
- [^A-Z] : (NOTE the square braces) any character NOT in the sequence within the braces
- [aeiou] : will match one of the vowels (Think of this as the OR operator within the square braces)
- A||B : matches ‘A’ or ‘B’

REGULAR EXPRESSIONS

- Each programming language implements slightly differently
  - Java : Pattern and Matcher classes
  - C : include <regex.h>
  - Perl : inbuilt
  - Python : import re;

N-GRAMS AND ZIPF'S LAW



BASIC IDEA:

- **Examine short sequences of words**
- **How likely is each sequence?**
- **“Markov Assumption” – word is affected only by its “prior local context” (last few words)**

EXAMPLE

- The boy ate a chocolate
- The girl bought a chocolate
- The girl then ate a chocolate
- The boy bought a horse

- Can we figure out how likely is the following sentence
  - The boy bought a chocolate

"SHANNON GAME"

- **Claude E. Shannon. "Prediction and Entropy of Printed English", *Bell System Technical Journal* 30:50-64. 1951.**
- **Predict the next word, given (*n-1)* previous words**
- **Determine probability of different sequences by examining training corpus**

FORMING EQUIVALENCE CLASSES (BINS)

- **"*n-gram*" = sequence of n words**
  - **bigram**
  - **trigram**
  - **four-gram or quadrigram**
- **Probabilities of n-grams**
  - **Unigram**

$$p(w) = \frac{c(w)}{N}$$

  - **Bigram**

$$P(w_i \mid w_{i-1}) = \frac{c(w_i, w_{i-1})}{c(w_{i-1})}$$

  - **Trigram**

$$P(w_i \mid w_{i-1}, w_{i-2}) = \frac{c(w_i, w_{i-1}, w_{i-2})}{c(w_{i-1}, w_{i-2})}$$

MAXIMUM LIKELIHOOD ESTIMATION

- The boy bought a chocolate
  - Unigram Probabilities
    - (4/16)*(2/16)*(2/16)*(4/16)*(3/16)
    - (4*2*2*4*3)/21^5 = 0.000047
  - Bi-gram Probabilities
    - <The boy> <boy bought> <bought a> <a chocolate>
    - (2/4)*(2/4)*(2/2)*(3/4) = 0.1875
- Data
  - The boy ate a chocolate
  - The girl bought a chocolate
  - The girl then ate a chocolate
  - The boy bought a horse

RELIABILITY VS. DISCRIMINATION

“large green _____”

tree? mountain? frog? car?

“swallowed the large green _____”

pill? candy?

RELIABILITY VS. DISCRIMINATION

- **larger n: more information about the context of the specific instance (greater discrimination)**

- **smaller n: more instances in training data, better statistical estimates (more reliability)**

SELECTING AN *N*

VOCABULARY (V) = 20,000 WORDS

n	Number of bins
2 (bigrams)	400,000,000
3 (trigrams)	8,000,000,000,000
4 (4-grams)	1.6×10^{17}

STATISTICAL ESTIMATORS

- **Given the observed training data …**
  - **How do you develop a model (probability distribution) to predict future events?**
  - **Language Modeling**
    - **Predict Likelihood of sequences**

MAXIMUM LIKELIHOOD ESTIMATION

- $P_{MLE}(w_n|w_1 \ldots w_{n-1}) = \frac{C(w_1 \ldots w_n)}{C(w_1 \ldots w_{n-1})}$

- Estimate sequence probabilities using “counts” or frequencies of sequences
- Problems
  - Sparseness
  - What do you do when unknown words are seen??

EXAMPLE

- Data
  - The boy ate a chocolate
  - The girl bought a chocolate
  - The girl then ate a chocolate
  - The horse bought a boy

- The boy bought a chocolate
  - Unigram Probabilities
    - (4/16)*(2/16)*(2/16)*(4/16)*(3/16)
    - (4*2*2*4*3)/21^5 = 0.000047
  - Bi-gram Probabilities
    - <The boy> <boy bought> <bought a> <a chocolate>
    - (2/4)*(0/4)*(2/2)*(3/4) = **0**

APPROXIMATING SHAKESPEARE

- Generating sentences with random unigrams…
  - Every enter now severally so, let
  - Hill he late speaks; or! a more to leg less first you enter
- With bigrams…
  - What means, sir. I confess she? then all sorts, he is trim, captain.
  - Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry.
- Trigrams
  - Sweet prince, Falstaff shall die.
  - This shall forbid it should be branded, if renown made it empty.

From Julia Hirschberg

- Quadrigrams
  - What! I will go seek the traitor Gloucester.
  - Will you not tell me who I am?
  - What's coming out here looks like Shakespeare because it ***is*** Shakespeare
- Note: ***As we increase the value of N, the accuracy of an n-gram model increases, since choice of next word becomes increasingly constrained***

From Julia Hirschberg

N-GRAM TRAINING SENSITIVITY

- If we repeated the Shakespeare experiment but trained our n-grams on a Wall Street Journal corpus, what would we get?
- Note: ***This question has major implications for corpus selection or design***

From Julia Hirschberg

WSJ IS *NOT* SHAKESPEARE: SENTENCES GENERATED FROM WSJ

unigram: Months the my and issue of year foreign new exchange's september were recession exchange new endorsed a acquire to six executives

bigram: Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her

trigram: They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions

From Julia Hirschberg

EVALUATION AND DATA SPARSITY QUESTIONS

- Perplexity and entropy: how do you ***estimate*** how well your language model fits a corpus once you're done?
- Smoothing and Backoff : how do you handle unseen n-grams?

From Julia Hirschberg

PERPLEXITY AND ENTROPY

- Information theoretic metrics
  - Useful in measuring how well a grammar or language model (LM) models a natural language or a corpus
- Entropy: With 2 LMs and a corpus, which LM is the better match for the corpus? How much information is there (in e.g. a grammar or LM) about what the next word will be? More is better!
  - For a random variable ***X*** ranging over e.g. bigrams and a probability function ***p(x),*** the entropy of X is the expected negative log probability

$$H(X) = -\sum_{x=1}^{x=n} p(x)\log_2 p(x)$$

From Julia Hirschberg

- Entropy is the lower bound on the # of bits it takes to encode information e.g. about bigram likelihood

- Cross Entropy
  - An upper bound on entropy derived from estimating true entropy by a subset of possible strings – we don't know the real probability distribution

- Perplexity
  - At each choice point in a grammar
    - What are the average number of choices that can be made, weighted by their probabilities of occurrence?
    - I.e., Weighted average branching factor
  - How much probability does a grammar or language model (LM) assign to the sentences of a corpus, compared to another LM? The more information, the lower perplexity

$$PP(W) = 2^{H(W)}$$

From Julia Hirschberg

SOME USEFUL OBSERVATIONS

- There are 884,647 tokens, with 29,066 word form types, in an approximately one million word Shakespeare corpus
  - Shakespeare produced 300,000 bigram types out of 844 million possible bigrams: so, ***99.96% of the possible bigrams were never seen (have zero entries in the table)***
- A small number of events occur with high frequency
- A large number of events occur with low frequency
- You can quickly collect statistics on the high frequency events
- You might have to wait an arbitrarily long time to get valid statistics on low frequency events
- Some zeroes in the table are really zeros But others are simply low frequency events you haven't seen yet. How to address?

From Julia Hirschberg

ZIPF'S LAW

George Kingsley Zipf
1902-1950

- Frequency of occurrence of words is inversely proportional to the rank in this frequency of occurrence.
- When both are plotted on a log scale, the graph is a straight line.

ZIPF DISTRIBUTION

- The Important Points:
  - a few elements occur *very frequently*
  - a medium number of elements have medium frequency
  - many elements occur *very infrequently*

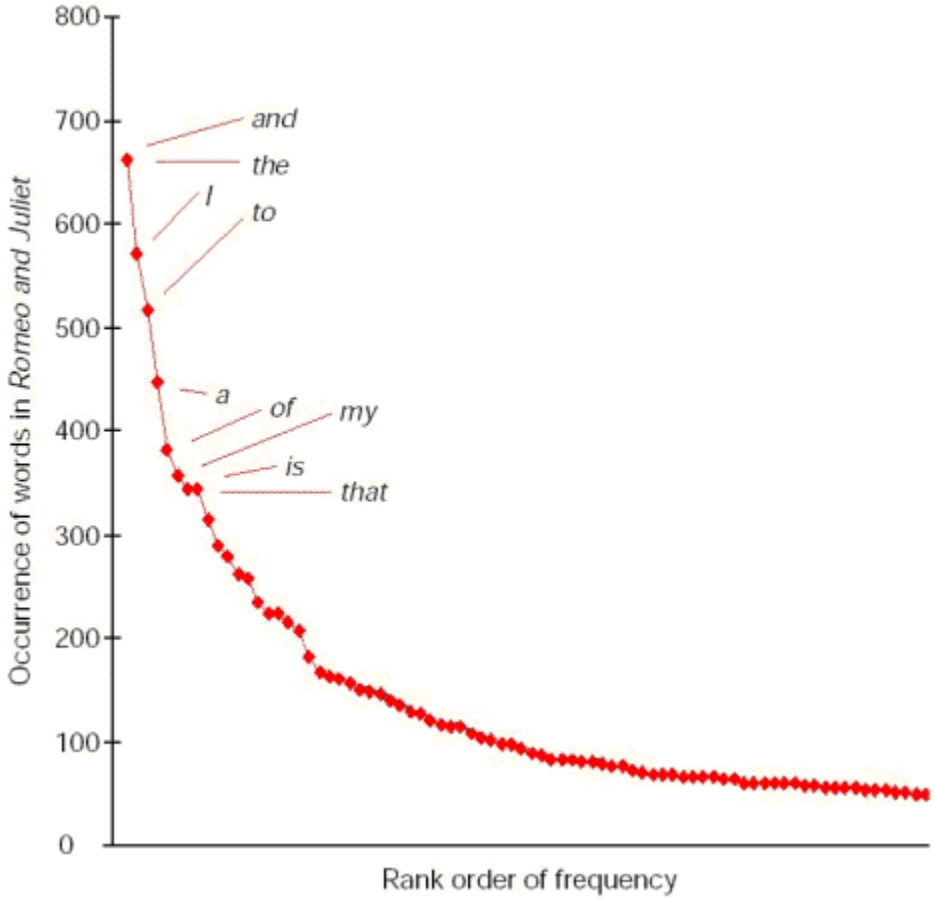
ZIPF DISTRIBUTION

The product of the frequency of words (f) and their rank (r) is approximately constant

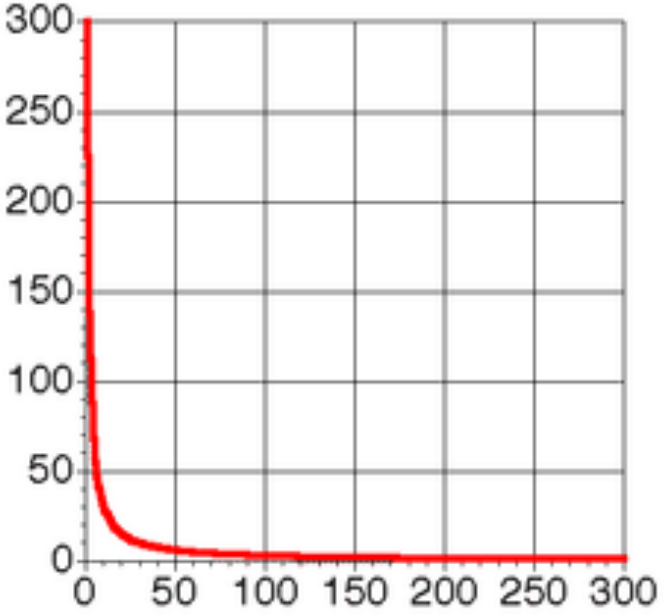
Rank = order of words' frequency of occurrence

$$f = C * 1/r$$

$$C \cong N/10$$



ZIPF DISTRIBUTION (SAME CURVE ON LINEAR AND LOG SCALE)



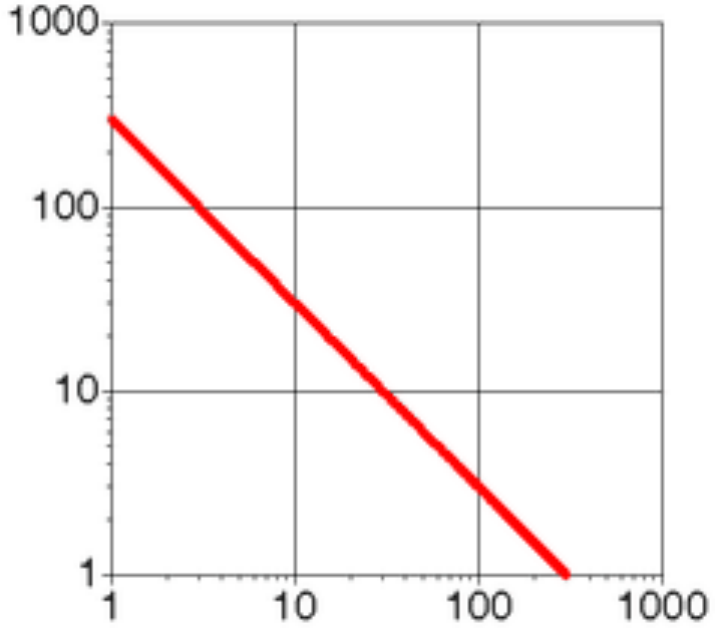
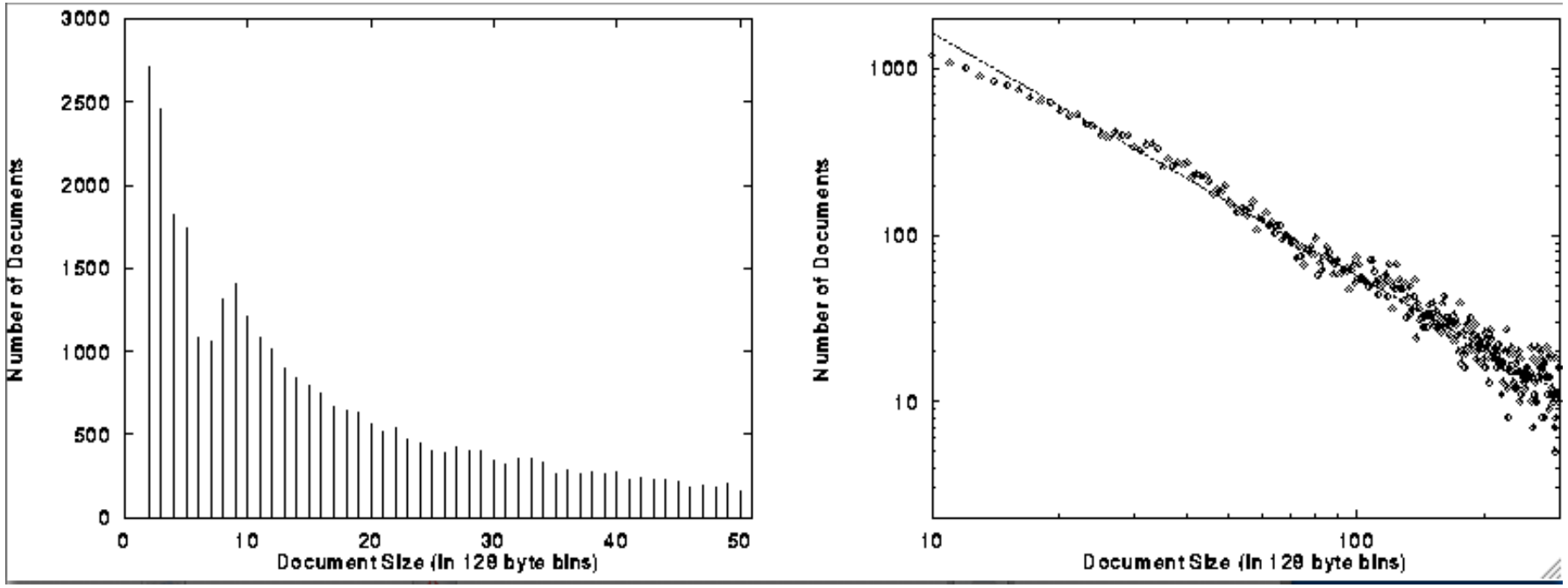


Illustration by Jacob Nielsen

WHAT KINDS OF DATA EXHIBIT A ZIPF DISTRIBUTION?

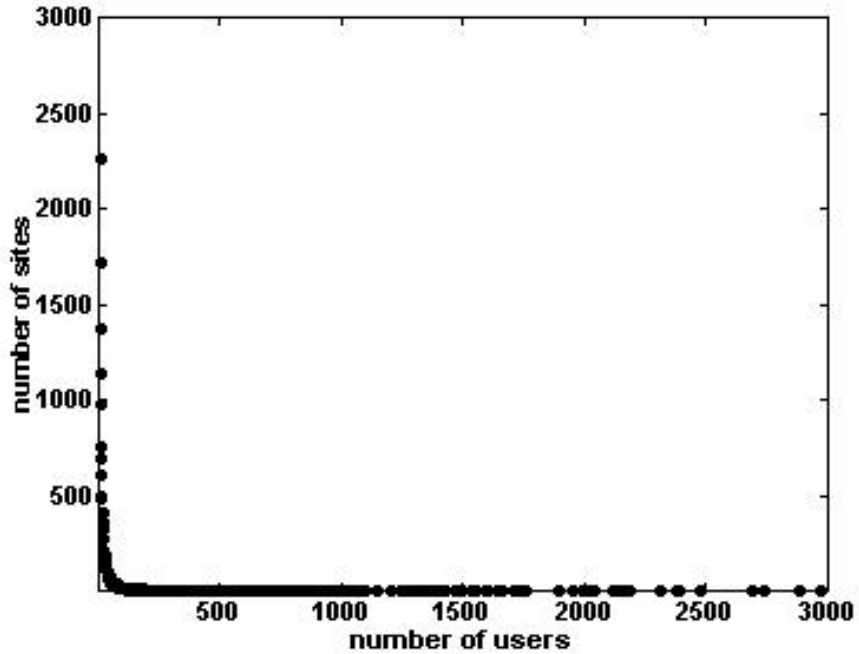
- Words in a text collection
  - Virtually any language usage
- Library book checkout patterns
- Incoming Web Page Requests (Nielsen)
- Outgoing Web Page Requests (Cunha & Crovella)
- Document Size on Web (Cunha & Crovella)

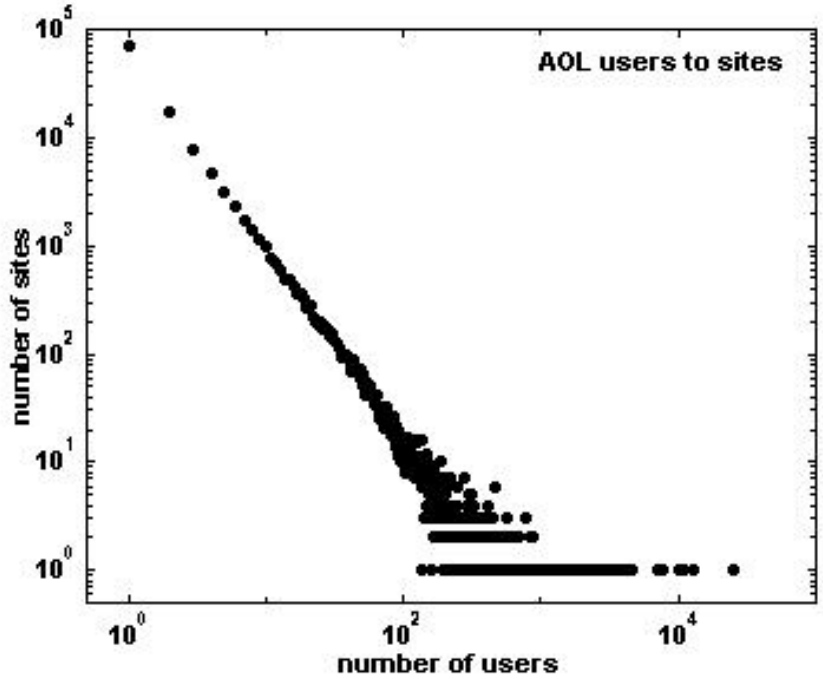
CHARACTERISTICS OF WWW CLIENT-BASED TRACES



Zipf's Law Applied To WWW Documents

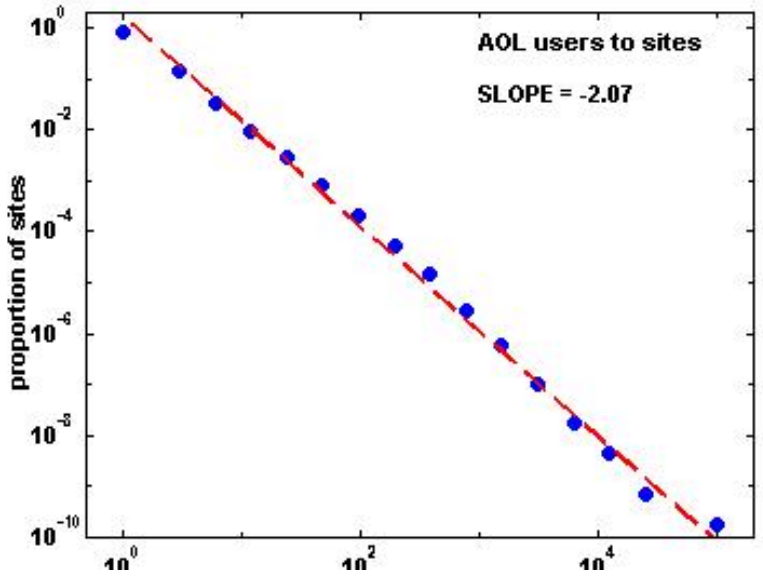
DISTRIBUTION OF USERS AMONG WEB SITES





Binned distribution of users to sites

Exponentially increasing bins



Cumulative distribution of users to sites