Some Basic Concepts in NLP

Some terms

- Corpus: Collection of meaningful sentences or documents
- Lexicon: Collection of unique words
- Pre-processing
- Corpus frequency
- Bag-of-word
- Zipfs law

Preprocessing

- Text Normalization

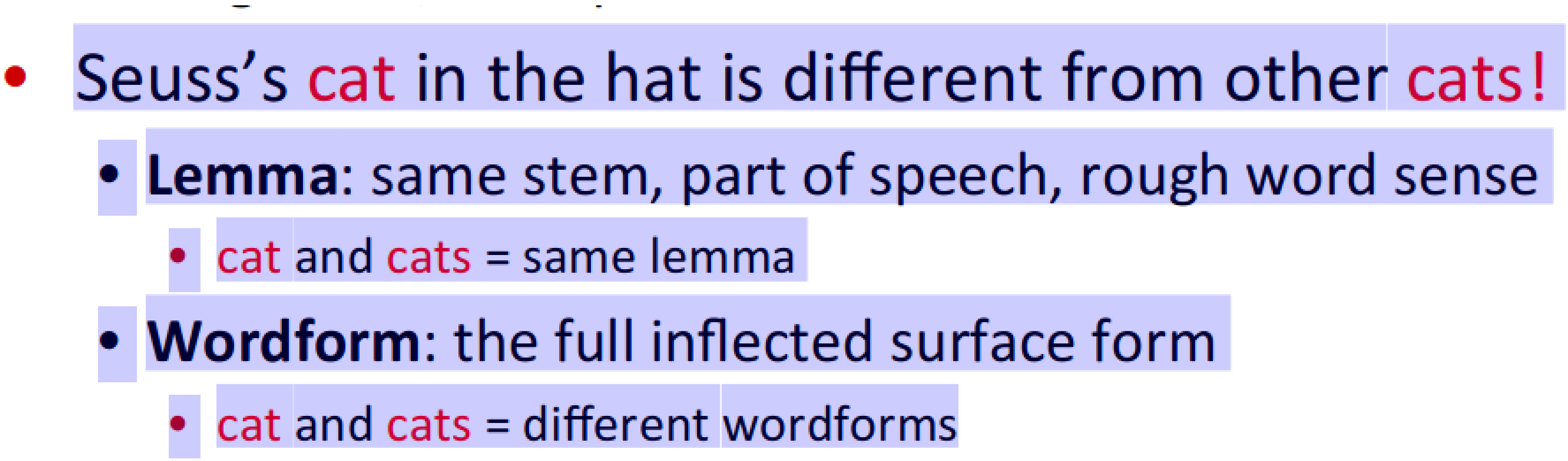
- Every NLP task needs to do text normalization:

  1.Segmenting/tokenizing words in running text

  2.Normalizing word formats

  3.Segmenting sentences in running text

How many words?



- 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

How many words?

they lay back on the San Francisco grass and looked at the stars and their

- **Type**: an element of the vocabulary.
- **Token**: an instance of that type in running text.
- How many?
  - 15 tokens (or 14)
  - 13 types (or 12) (or 11?)

How many words?

N = number of tokens

V = vocabulary = set of types

|*V*| is the size of the vocabulary

Church and Gale (1990): $|V| > O(N^{\frac{1}{2}})$

	Tokens = N	Types = |V|
Switchboard phone conversations	2.4 million	20 thousand
Shakespeare	884,000	31 thousand
Google N-grams	1 trillion	13 million

Tokenization

- (Inspired by Ken Church's UNIX for Poets.)
- Given a text file, output the word tokens and their frequencies

```
tr -sc 'A-Za-z' '\n' < shakes.txt
      | sort
      | uniq –c
```

Change all non-alpha to newlines

Sort in alphabetical order

Merge and count each type

```
1945 A          25 Aaron
  72 AARON       6 Abate
  19 ABBESS      1 Abates
   5 ABBOT       5 Abbess
 ... ...         6 Abbey
                 3 Abbot
               …. …
```

Issues in Tokenization

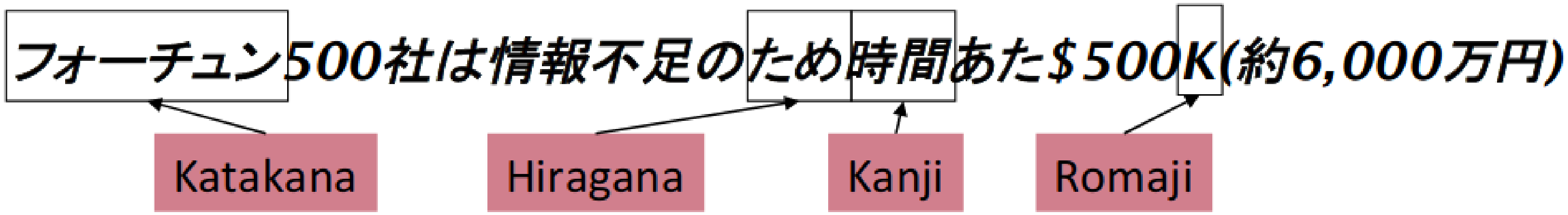
- Finland's capital → Finland Finlands Finland's ?
- what're, I'm, isn't → What are, I am, is not
- Hewlett-Packard → Hewlett Packard ?
- state-of-the-art → state of the art ?
- Lowercase → lower-case lowercase lower case ?
- San Francisco → one token or two?
- m.p.h., PhD. → ??

Tokenization: language issues

- French
  - ***L'ensemble*** → 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



フォーチュン500社は情報不足のため時間あた$500K(約6,000万円)

End-user can express query entirely in hiragana!

Normalization

- Need to “normalize” terms
  - Information Retrieval: indexed text & query terms must have same form.
    - We want to match ***U.S.A.*** and ***USA***
- We implicitly define equivalence classes of terms
  - e.g., deleting periods in a term
- Alternative: asymmetric expansion:
  - Enter: ***window*** Search: ***window, windows***
  - Enter: ***windows*** Search: ***Windows, windows, window***
  - Enter: ***Windows*** Search: ***Windows***
- Potentially more powerful, but less efficient

Bag of word model

Do you have enough money? Do you have everything you've been dreaming of? Are you happy with how your life is going? If the answer is 'Yes', then just don't waste your time and close this page.

Enough you have   have everything you've been dreaming of? Are you with how your is going? If the is 'Yes', then just don't your time and close money? this page Do everything Do you happy life answer waste

Corpus frequency

Zipf’s Law

George Kingsley Zipf
1902-1950

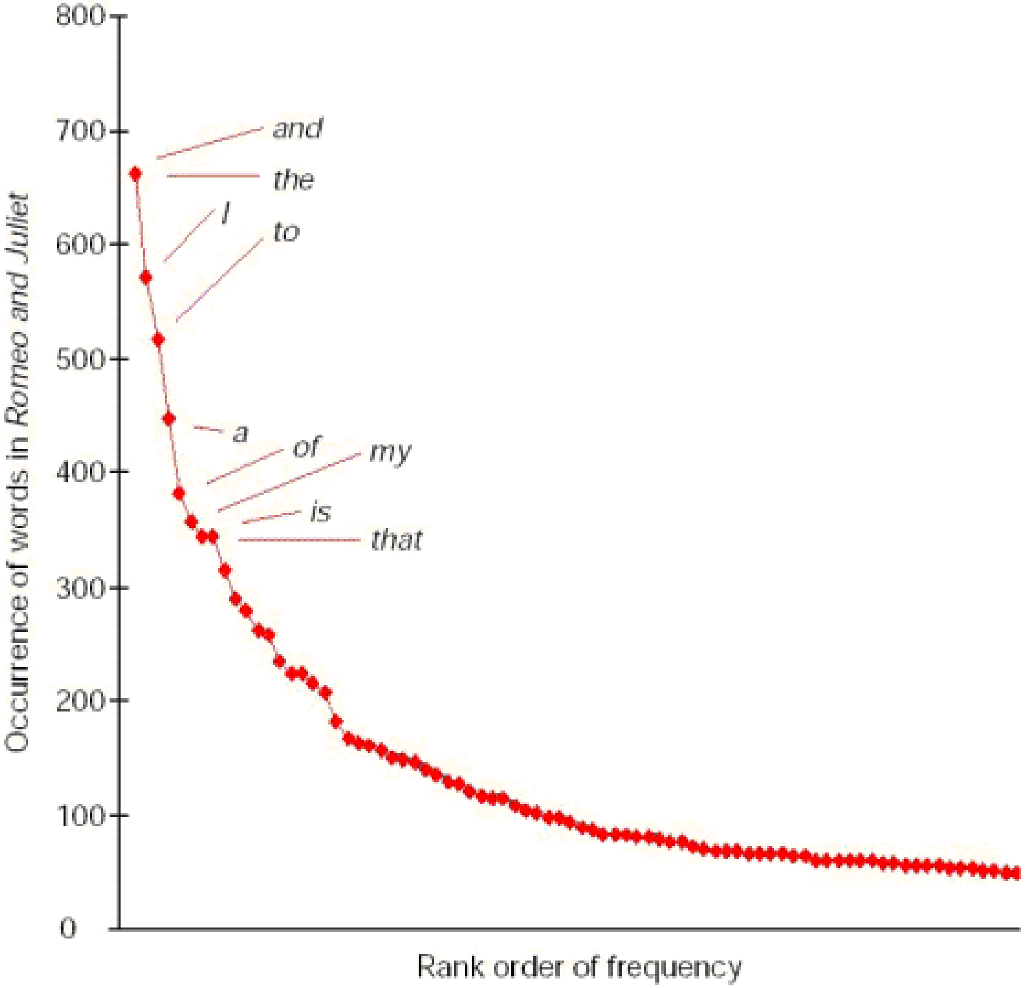
- Frequency of occurrence of words is inversely proportional to the rank in this frequency of occurrence.
- When both are plotted on a log scale, the graph is a straight line.

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

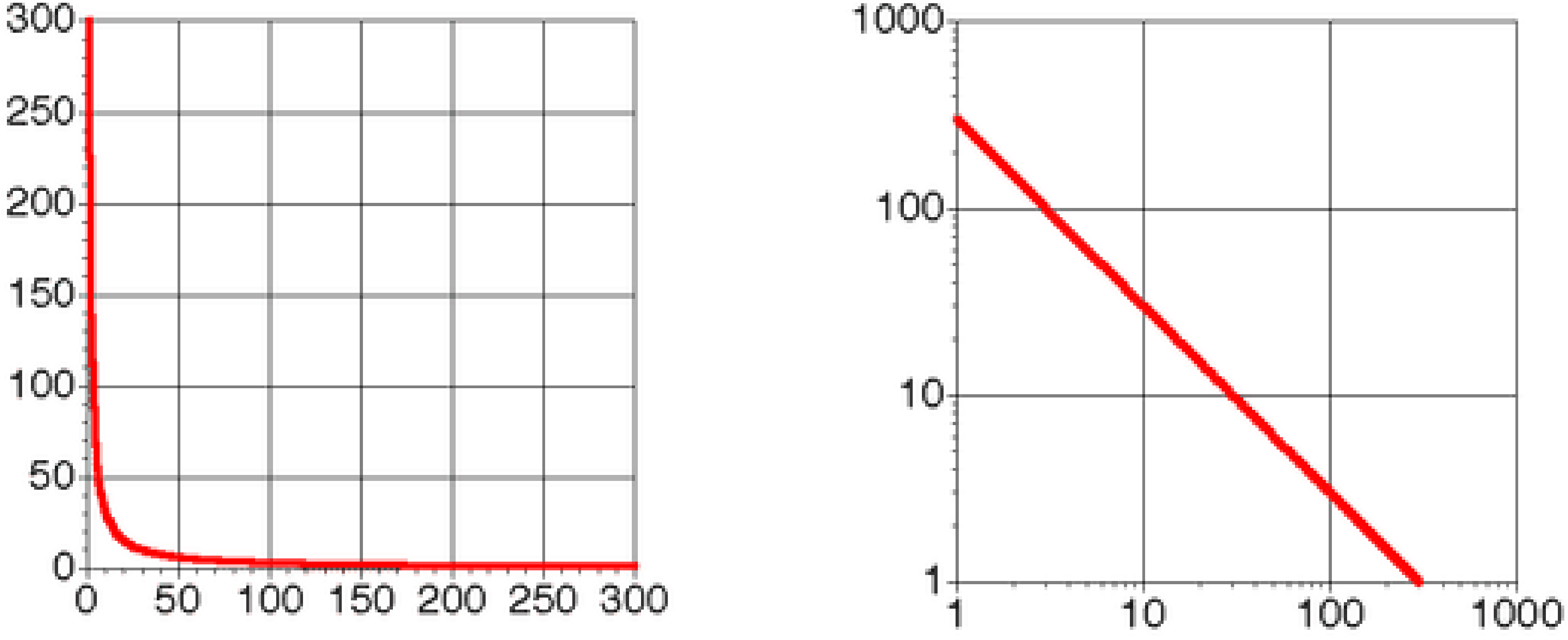


Rank = order of words' frequency of occurrence

$$f = C * 1/r$$

$$C \cong N/10$$

Zipf Distribution
(Same curve on linear and log scale)



300
250
200
150
100
50
0
0
50
100
150
200
250
300
1000
100
10
1
1
10
100
1000