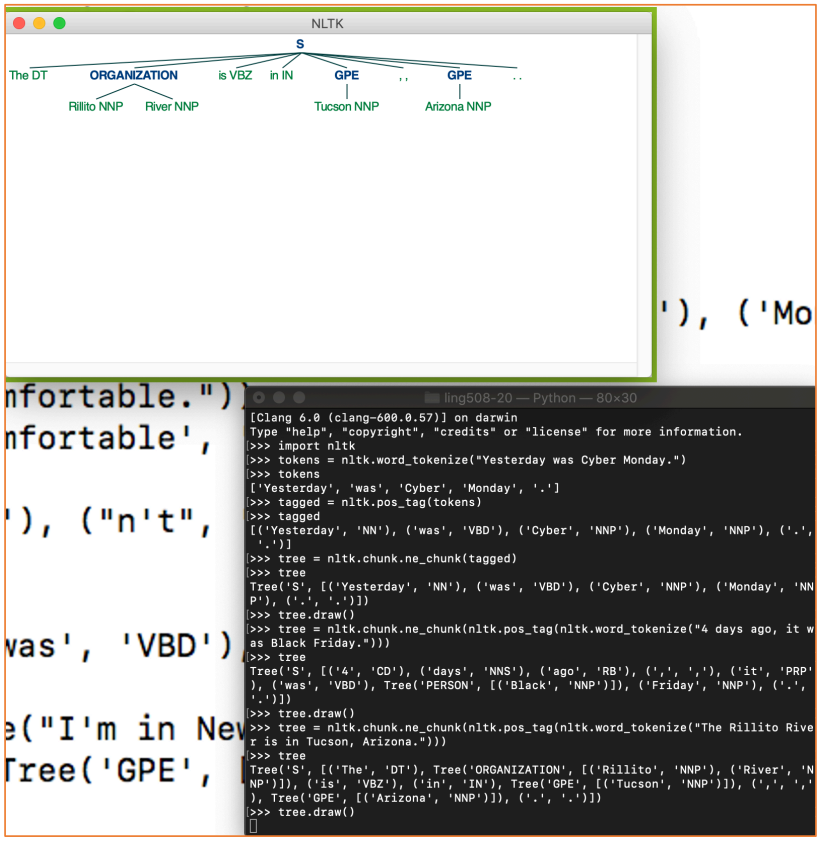
LING 408/508: Computational Techniques for Linguists

Lecture 29

Terminal from Last Time



- Steps:
  - draw tree
  - NE (Named Entity) chunk
  - POS (Part of Speech) tag
  - word tokenize
  - input is the raw string "*The Rillito River is in Tucson, Arizona.*"

Today's Topics

- Importing your own corpus: example
- More on things to do with corpora

nltk book: chapter 3

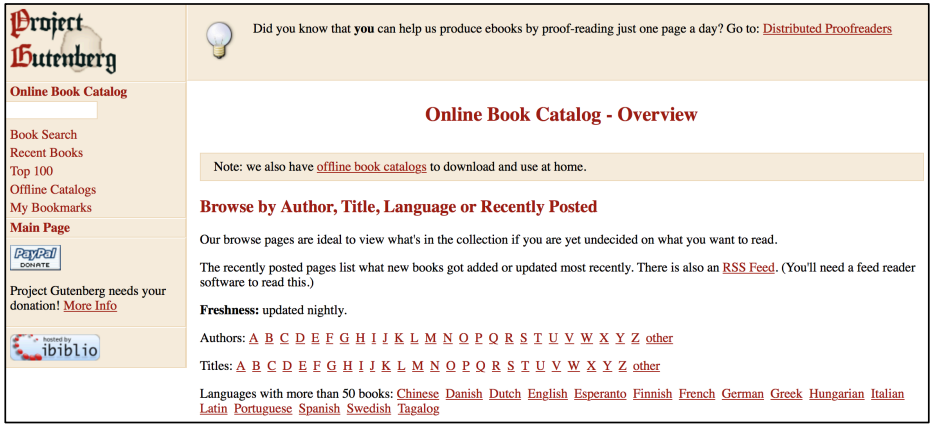
3 Processing Raw Text

Learning to import your own texts

Assume

```
import nltk, re, pprint
from nltk import word_tokenize
```

http://www.gutenberg.org/catalog/



nltk book: chapter 3

Project Gutenberg Australia
- (not indexed by www.gutenberg.org)

- `http://gutenberg.net.au/ebooks02/0200991.txt`
- *Mrs. Dalloway* by Virginia Woolf (1925)
- Code to read plaintext:

```
from urllib import request
url = "http://gutenberg.net.au/ebooks02/0200991.txt"
response = request.urlopen(url)
raw = response.read().decode('latin-1') # utf-8 common
```

ï»¿

Project Gutenberg Australia

a treasure-trove of literature

treasure found hidden with no evidence of ownership

Title: Mrs. Dalloway (1925)
Author: Virginia Woolf
* A Project Gutenberg of Australia eBook *
eBook No.: 0200991.txt
Edition: 1
Language: English
Character set encoding: Latin-1(ISO-8859-1)--8 bit
Date first posted: November 2002
Date most recently updated: November 2002

This eBook was produced by: Don Lainson dlainson@sympatico.ca

Project Gutenberg of Australia eBooks are created from printed editions which are in the public domain in Australia, unless a copyright notice is included. We do NOT keep any eBooks in compliance with a particular paper edition.

Copyright laws are changing all over the world. Be sure to check the copyright laws for your country before downloading or redistributing this file.

This eBook is made available at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg of Australia License which may be viewed online at http://gutenberg.net.au/licence.html

To contact Project Gutenberg of Australia go to http://gutenberg.net.au

---

A Project Gutenberg of Australia eBook

Title: Mrs. Dalloway
Author: Virginia Woolf

nltk book: chapter 3

```
>>> raw[:150]
'\r\n\r\nï»¿<table width="45%" border ="0">\r\n<tr>\r\n<td
bgcolor="#FFE4E1"><font color="#800000" size="5"><p style="text-
align:center"><b><a href="http://gut'
```

```
1
2
3  ï»¿<table width="45%" border ="0">
4  <tr>
5  <td bgcolor="#FFE4E1"><font color="#800000" size="5"><p style="text-align:center"><b><a href="http://gutenberg.net.au" tar
6  Gutenberg Australia</a><br>
7  </b></font><font color="#800000" size="4"><i>a treasure-trove of literature</i><br>
8  </font>treasure found hidden with no evidence of ownership</p></td>
9  </tr>
10 </table>
11 <!-- ad goes here -->
12 <pre>
13
14
15
16 Title:      Mrs. Dalloway (1925)
17 Author:     Virginia Woolf
18 * A Project Gutenberg of Australia eBook *
19 eBook No.:  0200991.txt
20 Edition:    1
21 Language:   English
22 Character set encoding:     Latin-1(ISO-8859-1)--8 bit
23 Date first posted:          November 2002
24 Date most recently updated: November 2002
25
26 This eBook was produced by: Don Lainson dlainson@sympatico.ca
```

https://docs.microsoft.com/en-us/windows/win32/intl/using-byte-order-marks

nltk book: chapter 3

```
30 is included. We do NOT keep any eBooks in compliance with a particular
31 paper edition.
32
33 Copyright laws are changing all over the world. Be sure to check the
34 copyright laws for your country before downloading or redistributing this
35 file.
36
37 This eBook is made available at no cost and with almost no restrictions
38 whatsoever. You may copy it, give it away or re-use it under the terms
39 of the Project Gutenberg of Australia License which may be viewed online at
40 http://gutenberg.net.au/licence.html
41
42 To contact Project Gutenberg of Australia go to http://gutenberg.net.au
43
44 ------------------------------------------------------------------------
45
46 A Project Gutenberg of Australia eBook
47
48 Title:      Mrs. Dalloway
49 Author:     Virginia Woolf
50
51
52
53
54 Mrs. Dalloway said she would buy the flowers herself.
55
56 For Lucy had her work cut out for her.  The doors would be taken
57 off their hinges; Rumpelmayer's men were coming.  And then, thought
58 Clarissa Dalloway, what a morning--fresh as if issued to children
```

```
>>> response = request.urlopen(url)
>>> raw = response.read().decode('latin-1')
>>> m = re.search('Title',raw)
>>> m
<_sre.SRE_Match object; span=(426, 431),
match='Title'>
>>> raw = raw[431:]
>>> m = re.search('Title',raw)
>>> m
<_sre.SRE_Match object; span=(1217, 1222),
match='Title'>
>>> raw = raw[1217:]
>>> raw[:200]
'Title:      Mrs.
Dalloway\r\nAuthor:     Virginia
Woolf\r\n\r\n\r\n\r\n\r\nMrs. Dalloway said she
would buy the flowers herself.\r\n\r\nFor Lucy
had her work cut out for her.  The doors would be
taken\r\noff their hing'
```

nltk book: chapter 3

```
6683 he had wondered, Who is that lovely girl? and it was his daughter!
6684 That did make her happy.  But her poor dog was howling.
6685
6686 "Richard has improved.  You are right," said Sally.  "I shall go
6687 and talk to him.  I shall say goodnight.  What does the brain
6688 matter," said Lady Rosseter, getting up, "compared with the heart?"
6689
6690 "I will come," said Peter, but he sat on for a moment.  What is
6691 this terror? what is this ecstasy? he thought to himself.  What is
6692 it that fills me with extraordinary excitement?
6693
6694 It is Clarissa, he said.
6695
6696 For there she was.
6697
6698
6699
6700 THE END
6701
6702
6703
6704
6705
6706 </pre>
6707 <p style="margin-left:10%"><img src="/pga-australia.jpg" width="80" height="75" alt=""> </p>
6708
6709 <p><b>This site is full of FREE ebooks - <a href="http://gutenberg.net.au" target="_blank">Project
6710 <!-- ad goes here -->
```

```
>>> raw[-400:]

'What is\r\nit that fills me with extraordinary
excitement?\r\n\r\nIt is Clarissa, he said.\r\n\r\nFor
there she was.\r\n\r\n\r\n\r\nTHE
END\r\n\r\n\r\n\r\n\r\n\r\n</pre>\r\n<p style="margin-
left:10%"><img src="/pga-australia.jpg" width="80"
height="75" alt=""> </p>\r\n\r\n<p><b>This site is full of
FREE ebooks - <a href="http://gutenberg.net.au"
target="_blank">Project Gutenberg
Australia</a></b></p>\r\n<!-- ad goes here --
>\r\n\r\n\r\n\r\n\r\n'

>>> m = re.search('THE END',raw)

>>> m

<_sre.SRE_Match object; span=(368969, 368976), match='THE
END'>

>>> raw = raw[:368976]

>>> raw[-400:]

'Sally.  "I shall go\r\nand talk to him.  I shall say
goodnight.  What does the brain\r\nmatter," said Lady
Rosseter, getting up, "compared with the heart?"\r\n\r\n"I
will come," said Peter, but he sat on for a moment.  What
is\r\nthis terror? what is this ecstasy? he thought to
himself.  What is\r\nit that fills me with extraordinary
excitement?\r\n\r\nIt is Clarissa, he said.\r\n\r\nFor
there she was.\r\n\r\n\r\n\r\nTHE END'
```

nltk book: chapter 3

```
>>> tokens = word_tokenize(raw)
>>> type(tokens)
<class 'list'>
>>> len(tokens)
77718
>>> tokens[:100]
['Title', ':', 'Mrs.', 'Dalloway', 'Author', ':', 'Virginia', 'Woolf',
'Mrs.', 'Dalloway', 'said', 'she', 'would', 'buy', 'the', 'flowers',
'herself', ',', 'For', 'Lucy', 'had', 'her', 'work', 'cut', 'out', 'for',
'her', '.', 'The', 'doors', 'would', 'be', 'taken', 'off', 'their',
'hinges', ';', 'Rumpelmayer', "'s", 'men', 'were', 'coming', '.', 'And',
'then', ',', 'thought', 'Clarissa', 'Dalloway', ',', 'what', 'a',
'morning', ',--', 'fresh', 'as', 'if', 'issued', 'to', 'children', 'on',
'a', 'beach', '.', 'What', 'a', 'lark', '!', 'What', 'a', 'plunge', '!',
'For', 'so', 'it', 'had', 'always', 'seemed', 'to', 'her', ',', 'when',
',', 'with', 'a', 'little', 'squeak', 'of', 'the', 'hinges', ',', 'which',
'she', 'could', 'hear', 'now', ',', 'she', 'had', 'burst']
```

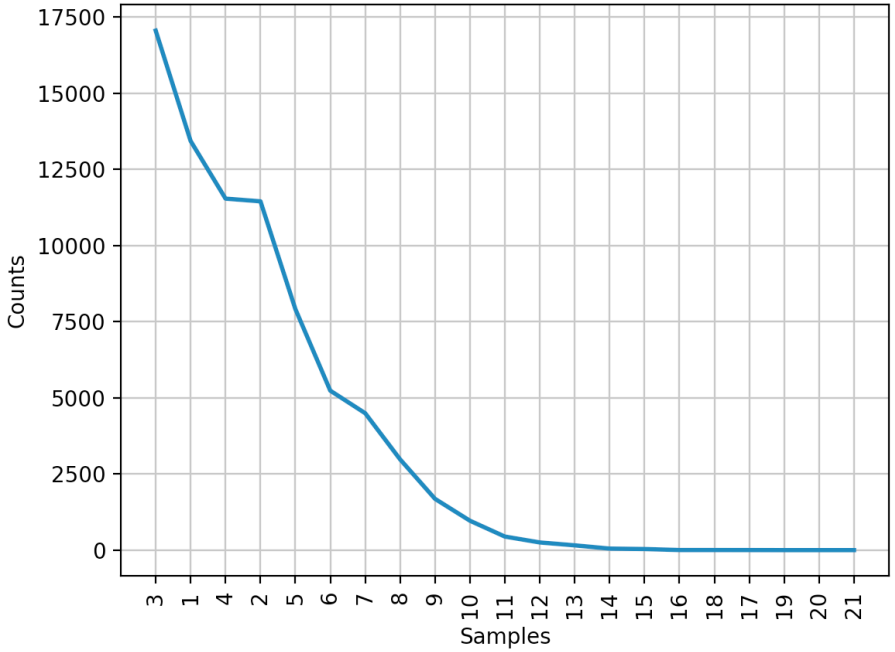
nltk book: chapter 3

```
>>> text = nltk.Text(tokens)
>>> len(text)
77718
>>> len(set(text))
7623
>>> len(set(text)) / len(text)
0.09808538562495175
>>> text.count('Dalloway')
104
>>> text.count('Mrs.')
118
```

```
>>> fd = nltk.FreqDist(text)
>>> fd
FreqDist({',': 6098, '.': 3017, 'the': 3015, 'and': 1625, 'of': 1525, ';': 1473, 'to': 1447, 'a': 1328, 'was': 1254, 'her': 1227, ...})
>>> print(fd)
<FreqDist with 7623 samples and 77718 outcomes>
>>> fd['Dalloway']
104
>>> text.collocations()
Peter Walsh; Sir William; Lady Bruton; Miss Kilman; Dr. Holmes; Prime
Minister; Ellie Henderson; Mrs. Filmer; Mrs. Dalloway; Hugh Whitbread;
Warren Smith; Sally Seton; Aunt Helena; Big Ben; Richard Dalloway;
motor car; Miss Parry; motor cars; years ago; Bond Street
```

nltk book: chapter 3

```
>>> fd2 = nltk.FreqDist(len(word) for word in text)
>>> fd2.most_common()
[(3, 17056), (1, 13433), (4, 11541), (2, 11452), (5, 7915), (6, 5236), (7,
4496), (8, 2980), (9, 1684), (10, 966), (11, 446), (12, 253), (13, 158),
(14, 51), (15, 37), (16, 4), (18, 4), (17, 3), (19, 1), (20, 1), (21, 1)]
>>> fd2.plot()
```



nltk book: chapter 3

```
ï»¿<table width="45%" border ="0">
<tr>
<td bgcolor="#FFE4E1"><font color="#800000" size="5"><p style="text-align:center"><b><a href="http://gutenberg.n
Gutenberg Australia</a><br>
</b></font><font color="#800000" size="4"><i>a treasure-trove of literature</i><br>
</font>treasure found hidden with no evidence of ownership</p></td>
</tr>
</table>
<!-- ad goes here -->
<pre>



Title:      Mrs. Dalloway (1925)
Author:     Virginia Woolf



It is Clarissa, he said.

For there she was.


THE END




</pre>
<p style="margin-left:10%"><img src="/pga-australia.jpg" width="80" height="75" alt=""> </p>

<p><b>This site is full of FREE ebooks - <a href="http://gutenberg.net.au" target="_blank">Project Gutenberg
<!-- ad goes here -->
```

nltk book: chapter 3

- Dealing with html

```
>>> html = request.urlopen(url).read().decode('latin-1')
>>> html[:60]
'\r\n\r\nï»¿<table width="45%" border ="0">\r\n<tr>\r\n<td bgcolor="#'
>>> from bs4 import BeautifulSoup
>>> raw = BeautifulSoup(html).get_text()
>>> tokens = word_tokenize(raw)
>>> len(tokens)
77977
```

```
>>> tokens[:100]

['ï', '»', '¿', 'Project', 'Gutenberg', 'Australia', 'a', 'treasure-trove', 'of', 'literature', 'treasure', 'found', 'hidden',
'with', 'no', 'evidence', 'of', 'ownership', 'Title', ':', 'Mrs.', 'Dalloway', '(', '1925', ')', 'Author', ':', 'Virginia', 'Woolf',
'*', 'A', 'Project', 'Gutenberg', 'of', 'Australia', 'eBook', '*', 'eBook', 'No', '.', ':', '0200991.txt', 'Edition', ':', '1',
'Language', ':', 'English', 'Character', 'set', 'encoding', ':', 'Latin-1', '(', 'ISO-8859-1', ')', '--', '8', 'bit', 'Date', 'first',
'posted', ':', 'November', '2002', 'Date', 'most', 'recently', 'updated', ':', 'November', '2002', 'This', 'eBook', 'was',
'produced', 'by', ':', 'Don', 'Lainson', 'dlainson', '@', 'sympatico.ca', 'Project', 'Gutenberg', 'of', 'Australia', 'eBooks',
'are', 'created', 'from', 'printed', 'editions', 'which', 'are', 'in', 'the', 'public', 'domain', 'in']

>>> tokens[-100:]

['shall', 'go', 'and', 'talk', 'to', 'him', '.', 'I', 'shall', 'say', 'goodnight', '.', 'What', 'does', 'the', 'brain', 'matter', ',', '"', "'",
'said', 'Lady', 'Rosseter', ',', 'getting', 'up', ',', '`', 'compared', 'with', 'the', 'heart', '?', "'", '"', '`', 'I', 'will', 'come', ',',
'"', "'", 'said', 'Peter', ',', 'but', 'he', 'sat', 'on', 'for', 'a', 'moment', '.', 'What', 'is', 'this', 'terror', '?', 'what', 'is', 'this',
'ecstasy', '?', 'he', 'thought', 'to', 'himself', '.', 'What', 'is', 'it', 'that', 'fills', 'me', 'with', 'extraordinary', 'excitement',
'?', 'It', 'is', 'Clarissa', ',', 'he', 'said', '.', 'For', 'there', 'she', 'was', '.', 'THE', 'END', 'This', 'site', 'is', 'full', 'of', 'FREE',
'ebooks', '-', 'Project', 'Gutenberg', 'Australia']
```

nltk book: chapter 3

- Reading local files

```
>>> f = open('document.txt')
>>> raw = f.read()
```

```
>>> f = open(path, encoding='latin2')
>>> for line in f:
...     line = line.strip()
...     print(line)
Pruska Biblioteka Państwowa. Jej dawne zbiory znane pod nazwą
"Berlinka" to skarb kultury i sztuki niemieckiej. Przewiezione przez
Niemców pod koniec II wojny światowej na Dolny Śląsk, zostały
odnalezione po 1945 r. na terytorium Polski. Trafiły do Biblioteki
Jagiellońskiej w Krakowie, obejmują ponad 500 tys. zabytkowych
archiwaliów, m.in. manuskrypty Goethego, Mozarta, Beethovena, Bacha.
```

nltk book: chapter 3

- Searching Tokenized Text in nltk
  angle brackets <…> mark token boundaries

```
>>> text[:20]
['Title', ':', 'Mrs.', 'Dalloway', 'Author', ':', 'Virginia', 'Woolf',
'the', 'flowers', 'herself', '.', 'For', 'Lucy']
```

```
>>> text.findall(r"<Mrs\.> (<\w+>)")
Dalloway; Dalloway; Foxcroft; Dalloway; Asquith; Dalloway; Richard;
Dalloway; Dalloway; Dalloway; Coates; Coates; Bletchley; Bletchley;
Dempster; Dempster; Dempster; Dempster; Dempster; Dempster; Dempster;
Dalloway; Walker; Dalloway; Walker; Dalloway; Dalloway; Dalloway;
Dalloway; Turner; Filmer; Hugh; Septimus; Filmer; Filmer; Warren;
Smith; Filmer; Smith; Warren; Dalloway; Whitbread; Marsham; Marsham;
Marsham; Marsham; Hilbery; Dalloway; Dalloway; Dalloway; Dalloway;
Dalloway; Dalloway; Marsham; Marsham; Dalloway; Dalloway; Gorham;
Dalloway; Filmer; Peters; Peters; Filmer; Peters; Peters; Filmer;
Peters; Peters; Peters; Peters; Filmer; Peters; Peters; Peters;
Filmer; Filmer; Filmer; Williams; Filmer; Filmer; Filmer; Filmer;
Filmer; Filmer; Filmer; Filmer; Burgess; Burgess; Burgess; Morris;
Morris; Walker; Walker; Dalloway; Walker; Walker; Walker; Parkinson;
Barnet; Barnet; Barnet; Barnet; Barnet; Garrod; Hilbery; Mount;
Dakers; Durrant; Hilbery; Hilbery; Dalloway; Dalloway; Dalloway;
Dalloway; Hilbery; Hilbery
```

nltk book: chapter 3

```
>>> from nltk.corpus import gutenberg, nps_chat
>>> moby = nltk.Text(gutenberg.words('melville-moby_dick.txt'))
>>> moby.findall(r"<a> (<.*>) <man>")  ❶
monied; nervous; dangerous; white; white; white; pious; queer; good;
mature; white; Cape; great; wise; wise; butterless; white; fiendish;
pale; furious; better; certain; complete; dismasted; younger; brave;
brave; brave; brave
>>> chat = nltk.Text(nps_chat.words())
>>> chat.findall(r"<.*> <.*> <bro>")  ❷
you rule bro; telling you bro; u twizted bro
>>> chat.findall(r"<l.*>{3,}")  ❸
lol lol lol; lmao lol lol; lol lol lol; la la la la la; la la la; la
la la; lovely lol lol love; lol lol lol.; la la la; la la la
```

nltk book: chapter 3

```
>>> from nltk.corpus import brown
>>> hobbies_learned = nltk.Text(brown.words(categories=['hobbies',
'learned']))
>>> hobbies_learned.findall(r"<\w*> <and> <other> <\w*s>")
```

speed and other activities; water and other liquids; tomb and other landmarks; Statues and other monuments; pearls and other jewels; charts and other items; roads and other features; figures and other objects; military and other areas; demands and other factors; abstracts and other compilations; iron and other metals

NLTK Book

- Chapter 1: section 4. http://www.nltk.org/book/ch01.html
- Highlights:

Table 4.2:

Some Word Comparison Operators

Function	Meaning
`s.startswith(t)`	test if `s` starts with `t`
`s.endswith(t)`	test if `s` ends with `t`
`t in s`	test if `t` is a substring of `s`
`s.islower()`	test if `s` contains cased characters and all are lowercase
`s.isupper()`	test if `s` contains cased characters and all are uppercase
`s.isalpha()`	test if `s` is non-empty and all characters in `s` are alphabetic
`s.isalnum()`	test if `s` is non-empty and all characters in `s` are alphanumeric
`s.isdigit()`	test if `s` is non-empty and all characters in `s` are digits
`s.istitle()`	test if `s` contains cased characters and is titlecased (i.e. all words in `s` have initial capitals)

NLTK Book

```
dhcp-10-142-130-43:~ sandiway$ python3
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 26 2016, 10:47:25)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from nltk.book import *
*** Introductory Examples for the NLTK Book ***
Loading text1, ..., text9 and sent1, ..., sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
text7: Wall Street Journal
text8: Personals Corpus
text9: The Man Who Was Thursday by G . K . Chesterton 1908
>>>
```

Recall startup procedure …

NLTK Book

text1: Moby Dick

```
>>> sorted(w for w in set(text1) if w.endswith('ableness'))
['comfortableness', 'honourableness', 'immutableness', 'indispensableness',
'indomitableness', 'intolerableness', 'palpableness', 'reasonableness',
'uncomfortableness']
```

text7: Wall Street Journal

```
>>> sorted(w for w in set(text7) if w.istitle() and '.' in w)
['A.', 'A.C.', 'A.D.', 'A.L.', 'Ala.', 'Ariz.', 'Aug.', 'B.', 'B.A.T', 'C.', 'C.J.B.',
'Calif.', 'Co.', 'Colo.', 'Conn.', 'Corp.', 'Cos.', 'D.', 'D.C.', 'Dec.', 'Del.', 'Dr.',
'E.', 'E.C.', 'E.W.', 'F.', 'F.H.', 'F.W.', 'Feb.', 'Fla.', 'G.', 'Ga.', 'Gov.', 'H.',
'H.N.', 'I.', 'Ill.', 'Inc.', 'Ind.', 'J.', 'J.L.', 'J.P.', 'Jan.', 'Jr.', 'K.', 'Ky.',
'L.', 'L.A.', 'L.P.', 'La.', 'Lt.', 'Ltd.', 'M.', 'M.D.', 'Mass.', 'Md.', 'Messrs.',
'Mich.', 'Minn.', 'Miss.', 'Mo.', 'Mr.', 'Mrs.', 'Ms.', 'N.', 'N.C.', 'N.C.', 'N.H.',
'N.J.', 'N.J.', 'N.M.', 'N.V.', 'N.V.', 'N.Y.', 'N.Y.', 'Nev.', 'No.', 'Nov.', 'O.', 'Oct.',
'Ore.', 'P.', 'Pa.', 'Prof.', 'Pty.', 'R.', 'R.D.', 'R.I.', 'R.P.', 'Rep.', 'Rev.', 'S.',
'S.A.', 'S.I.', 'Sen.', 'Sept.', 'Sept.30', 'Sino-U.S.', 'Sr.', 'St.', 'T.', 'Tenn.',
'U.K.', 'U.S.', 'U.S.-Japan', 'U.S.-Japanese', 'U.S.A', 'U.S.A.', 'U.S.S.R.', 'Va.', 'W.',
'W.D.', 'W.N.', 'W.R.', 'Wash.', 'Wis.', 'Z.']
>>>
```

NLTK Book

- Different genres

```
>>> sorted(w for w in set(text1) if w.istitle() and '.' in w)
[]
>>> sorted(w for w in set(text7) if w.endswith('ableness'))
[]
```

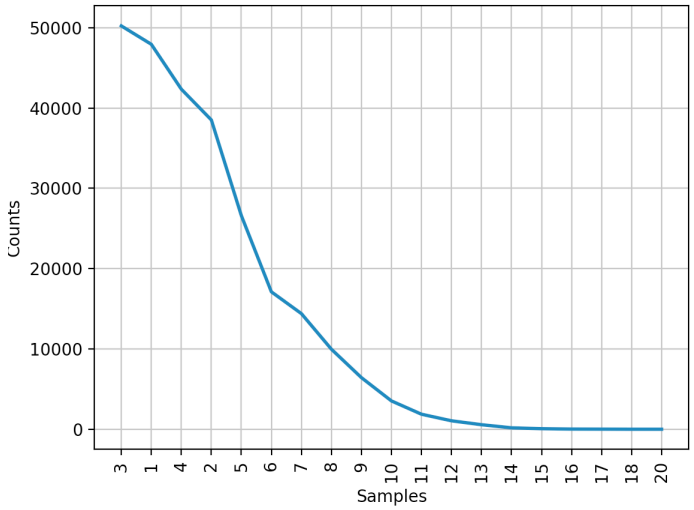
- Still more than 4000 words in common

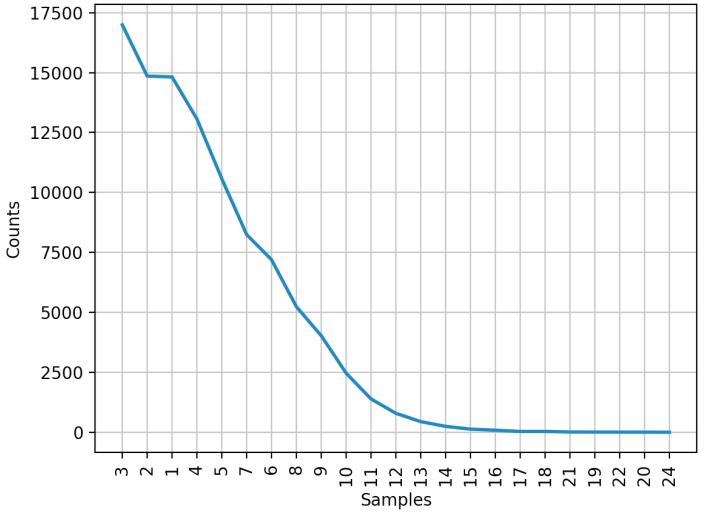
```
>>> len(set(text1).intersection(set(text7)))
4642
>>> len(set(text1))
19317
>>> len(set(text7))
12408
```

NLTK Book

- Frequency distribution comparison:

```
>>> fd1 = FreqDist(len(w) for w in text1)
>>> fd7 = FreqDist(len(w) for w in text7)
>>> fd1.plot()
>>> fd7.plot()
```





NLTK Book

```
>>> t1_3 = [w for w in text1 if len(w) == 3]
>>> t7_3 = [w for w in text7 if len(w) == 3]
>>> len(set(t1_3))
658
>>> len(set(t7_3))
595
>>> len(set(t7_3).intersection(set(t1_3)))
232
>>> set(t7_3).intersection(set(t1_3))
{'his', 'sum', 'NEW', 'sad', '890', 'far', 'die', 'log', 'him', 'its', 'how', 'Lee', 'Air', '144', 'six',
'end', 'air', 'jet', 'but', 'yon', 'Don', 'fit', 'old', 'How', 'arm', '125', 'fly', 'Red', 'eat', 'saw',
'day', 'raw', 'Too', 'not', 'aim', 'who', 'had', 'own', 'car', 'toy', '102', 'fee', 'Who', 'our', 'Leo',
'net', 'Del', 'sky', 'And', '800', 'Ark', 'Nor', 'get', 'son', 'sit', 'Law', 'led', '100', 'For', 'III',
'lay', 'joy', 'man', 'now', 'lot', 'New', 'Dan', '500', '132', 'may', 'End', 'few', 'war', 'den', 'Not',
'hay', 'set', 'Can', 'art', 'act', 'jam', 'Van', 'dam', 'fed', 'cow', 'hot', '128', 'add', 'ton', 'put',
'Old', 'Ray', '108', 'two', 'try', 'bag', 'run', 'gas', 'one', 'Joe', 'cry', '150', '103', 'Pan', 'beg',
'hit', 'THE', 'top', 'Any', 'pie', 'Two', 'TWO', 'van', 'see', '114', 'via', 'tow', 'The', 'ire', 'yet',
'107', 'out', 'odd', 'pit', 'say', 'You', 'any', 'Its', 'law', 'why', 'bar', 'His', 'rim', 'won', 'you',
'400', 'fat', 'low', 'has', 'rap', 'tip', 'boy', 'too', '115', '118', 'But', 'new', 'she', 'per', 'hid',
'Mrs', 'due', 'nor', 'key', '...', 'ask', 'eye', 'Yet', 'met', 'Put', '111', 'box', 'War', 'and', 'yes',
'for', 'map', 'Why', 'aid', 'bid', 'Now', 'Sir', 'fan', 'big', 'all', '106', 'row', 'Far', '120', 'men',
'All', 'sex', 'Few', 'Man', '1st', 'God', 'Her', '119', 'Ten', '105', 'age', 'May', 'Day', 'bit', '110',
'did', 'was', 'ill', '130', 'her', 'ran', 'lap', 'AND', 'job', 'the', 'dry', 'bad', 'red', 'tea', 'cap',
'got', 'let', 'Tom', 'are', '180', '135', 'can', 'oil', 'cut', '133', 'ago', 'She', 'One', '101', 'buy',
'use', 'vow', 'Sea', 'off', 'bat', 'way', 'pay'}
```

About 1/3 of the 3 letter words in common

NLTK Book

- 5 letter words in common:

```
>>> t1_5 = [w for w in text1 if len(w) == 5]
>>> t7_5 = [w for w in text7 if len(w) == 5]
>>> len(set(t1_5))
2397
>>> len(set(t7_5))
1531
>>> len(set(t1_5).intersection(set(t7_5)))
705
>>> t1_7 = [w for w in text1 if len(w) == 7]
>>> t7_7 = [w for w in text7 if len(w) == 7]
>>> len(set(t1_7))
3005
>>> len(set(t7_7))
1937
>>> len(set(t1_7).intersection(set(t7_7)))
746
```

About ½-1/3 of the 5 letter words in common

About ¼-1/3 of the 7 letter words in common