



Hands on Introduction to Text Mining

Hadaiq Rolis Sanabila
hadaiq@cs.ui.ac.id

Natural Language Processing and Text Mining

Pusilkom UI
22 – 26 Maret 2016



Preparation

- Download nltk.corpus -> names
 - Go to python interpreter on CMD
 - `>>>import nltk`
 - `>>>nltk.download()`

Load some text of several books

- Go to the command prompt

```
>>>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
```



Searching Text

- Concordance

>>>text1.concordance (cat)

Displaying 2 of 2 matches:

*78 Cistern and Buckets . Nimble as a cat , Tashtego
mounts aloft ; and without
ok the slight cedar as a mildly cruel cat her mouse . With
unastonished eyes Fe*

Searching Text

- Similar

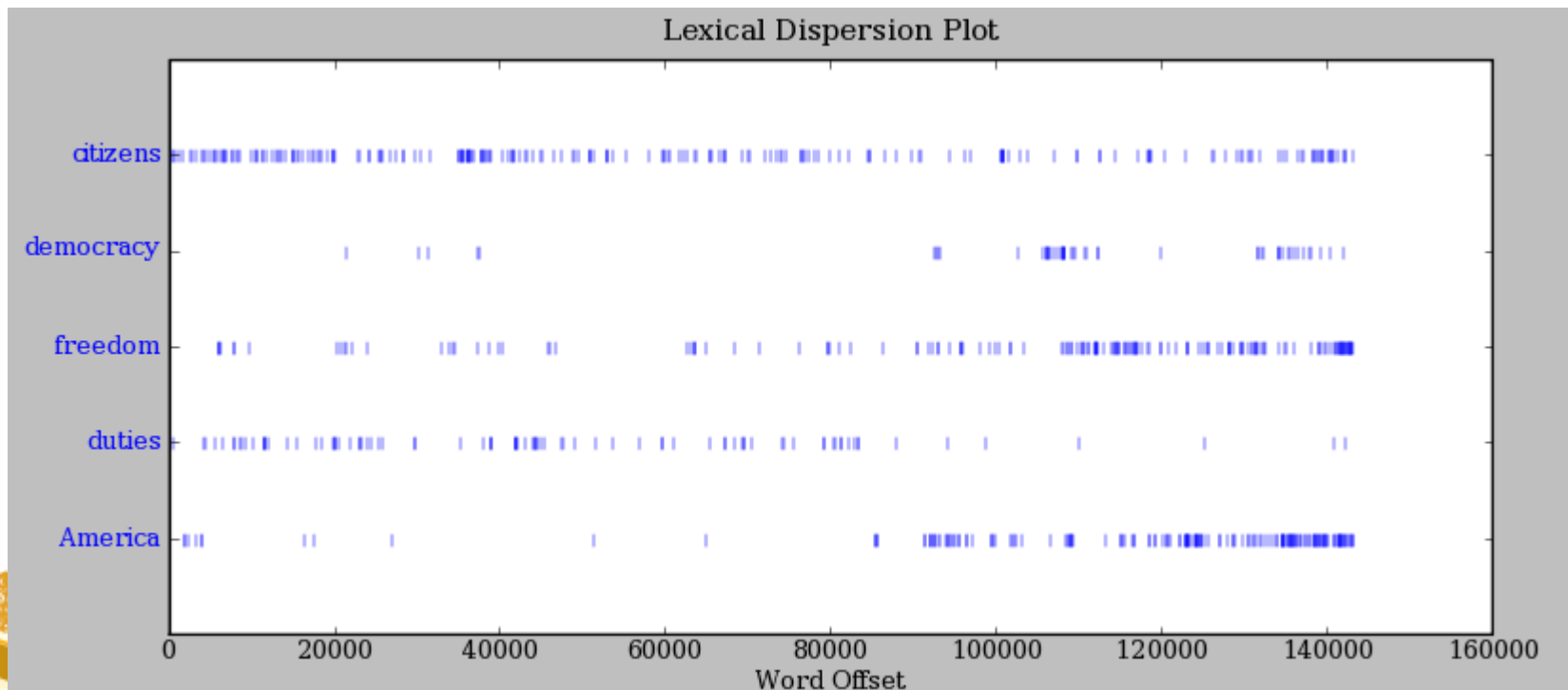
```
>>> text2.similar("love")
```

*affection town heart see mother sister time me word elinor
life it
marianne dear family him bed do regard her*

Searching Text

- Dispersion Plot

```
>>> text4.dispersion_plot(["citizens", "democracy", "freedom",  
"duties", "America"])
```



Counting Vocabulary

- Length of the text

```
>>> len(text2)  
141576
```

Counting Vocabulary

- Length of the text (character)

```
>>> len(text3)
```

```
44764
```

```
>>> sorted(set(text3))
```

```
['!', '"', '(', ')', ',', '.', ':', ';', '?', '?'),  
'A', 'Abel', 'Abelmizraim', 'Abidah', 'Abide', 'Abimael',  
'Abimelech',  
'Abr', 'Abrah', 'Abraham', 'Abram', 'Accad', 'Achbor',  
'Adah', ...]
```

```
>>> len (set(text3))
```

```
2789
```


Counting Vocabulary

- Lexical richness

```
>>> len(set(text3))/len(text3)
```

```
0.0623045304262353 ≈ 1/16
```

```
>>> text3. count("love")
```

```
4
```

Lists

```
>>> sent2
```

```
['The', 'family', 'of', 'Dashwood', 'had', 'long', 'been',  
'settled', 'in', 'Sussex', '.']
```

```
>>> len (sent2)
```

```
11
```

Lists

Concatenating two list

```
>>> ['Monty', 'Python'] + ['and', 'the', 'Holy', 'Grail']
```

```
['Monty', 'Python', 'and', 'the', 'Holy', 'Grail']
```

```
>>> sent1+sent6
```

```
['Call', 'me', 'Ishmael', '.', 'SCENE', 'I', ':', '[', 'wind', ']', '[',  
'clap', 'clap', 'clap', ']', 'KING', 'ARTHUR', ':', 'Whoa', 'there',  
'!']
```

```
>>> sent1.append("okay")
```

```
>>> sent1
```

```
['Call', 'me', 'Ishmael', '.', 'some', 'okay']
```

Lists

Indexing list

```
>>> text4[173]  
'awaken'  
>>> text4.index('awaken')  
173  
>>> text5[16715:16735]  
['U86', 'thats', 'why', 'something', 'like', 'gamefly', 'is', 'so',  
'good', 'because', 'you', 'can', 'actually', 'play', 'a', 'full',  
'game', 'without',  
'buying', 'it']  
>>> text6[1600:1625]  
['We', '', 're', 'an', 'anarcho', '-', 'syndicalist', 'commune', '.',  
'We', 'take', 'it', 'in', 'turns', 'to', 'act', 'as', 'a', 'sort', 'of',  
'executive', 'officer', 'for', 'the', 'week']
```



Lists

List slicing

```
>>> sent = ['word1', 'word2', 'word3', 'word4', 'word5',  
'word6', 'word7', 'word8', 'word9', 'word10']
```

```
>>> sent[0]
```

```
'word1'
```

```
>>> sent[9]
```

```
'word10'
```

```
>>> sent[5:8]
```

```
['word6', 'word7', 'word8']
```



Lists

List slicing

```
>>> sent[:3]  
['word1', 'word2', 'word3']
```

```
>>> text2[141525:]
```

```
['among', 'the', 'merits', 'and', 'the', 'happiness', 'of',  
'Elinor', 'and', 'Marianne', ',', 'let', 'it', 'not', 'be', 'ranked',  
'as', 'the', 'least', 'considerable', ',', 'that', 'though', 'sisters',  
,', 'and', 'living', 'almost', 'within', 'sight', 'of', 'each',  
'other', ',', 'they', 'could', 'live', 'without', 'disagreement',  
'between', 'themselves', ',', 'or', 'producing', 'coolness',  
'between', 'their', 'husbands', '.',  
'THE', 'END']
```



Lists

Variables

```
>>> my_sent = ['Bravely', 'bold', 'Sir', 'Robin', ',', 'rode',  
              'forth', 'from', 'Camelot', '.']
```

```
>>> noun_phrase = my_sent[1:4]
```

```
>>> noun_phrase  
['bold', 'Sir', 'Robin']
```

```
>>> words = sorted(noun_phrase)
```

```
>>> words  
['Robin', 'Sir', 'bold']
```

Lists

Strings

```
>>> name = 'Monty'
```

```
>>> name[0]  
'M'
```

```
>>> name[:4]  
'Mont'
```

```
>>> name * 2  
'MontyMonty'
```


Lists

Strings

```
>>> name + '!'  
'Monty!'
```

```
>>> ' '.join(['Monty', 'Python'])  
'Monty Python'
```

```
>>> 'Monty Python'.split()  
['Monty', 'Python']
```

Simple Statistics

Frequency Distribution

```
>>> fdist1 = FreqDist(text1)
```

```
>>> print(fdist1)
```

```
<FreqDist with 19317 samples and 260819 outcomes>
```



Simple Statistics

Frequency Distribution

```
>>> fdist1.most_common(10)
```

```
[(',', 18713), ('the', 13721), ('.', 6862), ('of', 6536), ('and',  
6024),  
('a', 4569), ('to', 4542), (';', 4072), ('in', 3916), ('that',  
2982),
```

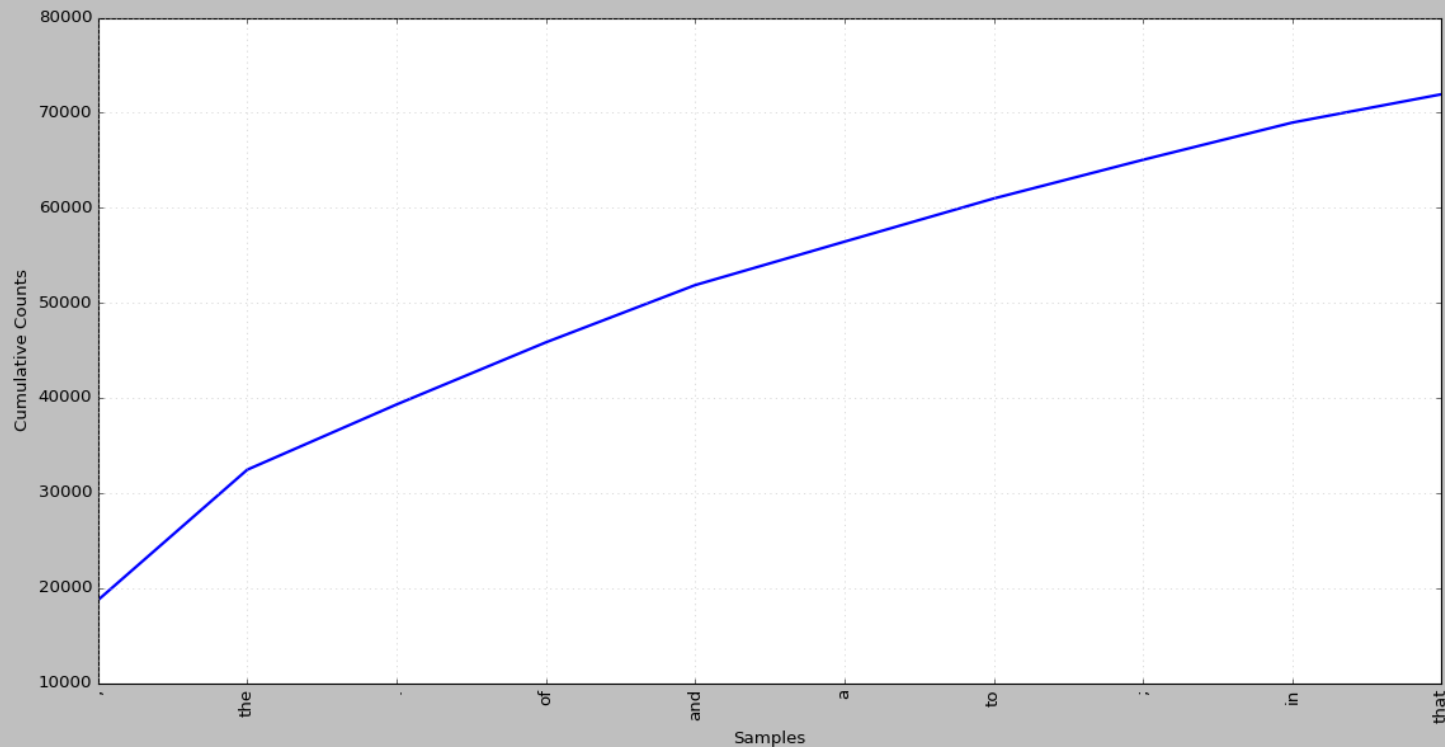
```
>>> fdist1['whale']
```

```
906
```

Simple Statistics

Frequency Distribution plot

```
>>> fdist1.plot(10, cumulative=True)
```



Simple Statistics

Fine-grained selection

```
>>> V = set(text1)
```

```
>>> long_words = [w for w in V if len(w) > 15]
```

```
>>> sorted(long_words)
```

```
['CIRCUMNAVIGATION', 'Physiognomically',  
'apprehensiveness', 'cannibalistically', 'characteristically',  
'circumnavigating', 'circumnavigation', 'circumnavigations',  
'comprehensiveness', 'hermaphroditical', 'indiscriminately',  
'indispensableness', 'irresistibleness', 'physiognomically',  
'preternaturalness', 'responsibilities', 'simultaneousness',  
'subterraneousness', 'supernaturalness', 'superstitiousness',  
'uncomfortableness', 'uncompromisedness', 'undiscriminating',  
'uninterpenetratingly']
```

Simple Statistics

Collocations and bigram

```
>>> list(bigrams(['more', 'is', 'said', 'than', 'done']))
```

```
[('more', 'is'), ('is', 'said'), ('said', 'than'), ('than', 'done')]
```

```
>>> text4.collocations()
```

*United States; fellow citizens; four years; years ago;
Federal Government; General Government; American
people; Vice President; Old World; Almighty God; Fellow
citizens; Chief Magistrate; Chief Justice; God bless; every
citizen; Indian tribes; public debt; one another; foreign
nations; political parties*

Thank you

