lab1_practice

January 18, 2022

1 Lab 1 - Practice

```
[ ]: import nltk
```

```
[ ]: # nltk.download()
```

```
[ ]: # import the gutenberg corpus
     from nltk.corpus import gutenberg
```

```
[ ]: gutenberg.fileids()
```

```
[ ]: ['austen-emma.txt',
      'austen-persuasion.txt',
      'austen-sense.txt',
      'bible-kjv.txt',
      'blake-poems.txt',
      'bryant-stories.txt',
      'burgess-busterbrown.txt',
      'carroll-alice.txt',
      'chesterton-ball.txt',
      'chesterton-brown.txt',
      'chesterton-thursday.txt',
      'edgeworth-parents.txt',
      'melville-moby_dick.txt',
      'milton-paradise.txt',
      'shakespeare-caesar.txt',
      'shakespeare-hamlet.txt',
      'shakespeare-macbeth.txt',
      'whitman-leaves.txt']
```

```
[ ]: hamlet = gutenberg.sents("shakespeare-hamlet.txt")
     hamlet
```

```
[ ]: [['[', 'The', 'Tragedie', 'of', 'Hamlet', 'by', 'William', 'Shakespeare',
     '1599', ']'], ['Actus', 'Primus', '.'], …]
```

```
[ ]: sum([len(s) for s in hamlet]) / len(hamlet)
[ ]: 12.028332260141662
```

1.1 NLTK functions

NLTK provides quite a lot of stuff...

Let's access functions available on the Text object

```
[ ]: from nltk.book import *
```

```
[ ]: [fn for fn in dir(text1) if "__" not in fn]
```

```
[ ]: ['_CONTEXT_RE',
      '_COPY_TOKENS',
      '_context',
      '_train_default_ngram_lm',
      'collocation_list',
      'collocations',
      'common_contexts',
      'concordance',
      'concordance_list',
      'count',
      'dispersion_plot',
      'findall',
      'generate',
      'index',
      'name',
      'plot',
      'readability',
      'similar',
      'tokens',
      'vocab']
```

```
[ ]: text1.concordance("fish")
```

```
Displaying 25 of 169 matches:
to teach them by what name a whale - fish is to be called in our tongue leavin
 " Now the Lord had prepared a great fish to swallow up Jonah ." -- JONAH . "
 and robbers , is the right to royal fish , which are whale and sturgeon . And
 the vast Atlantic is ; Not a fatter fish than he , Flounders round the Polar
 bright red windows of the " Sword - Fish Inn ," there came such fervent rays
rossed Harpoons ," and " The Sword - Fish ?"-- this , then must needs be the s
ar a faint resemblance to a gigantic fish ? even the great leviathan himself ?
here was a parcel of outlandish bone fish hooks on the shelf over the fire - p
nah --' And God had prepared a great fish to swallow up Jonah .'" " Shipmates
 noble thing is that canticle in the fish ' s belly ! How billow - like and bo
onah prayed unto the Lord out of the fish ' s belly . But observe his prayer ,
n he cried . Then God spake unto the fish ; and from the shuddering cold and b
 disdain , " ah ! him bevy small - e fish - e ; Queequeg no kill - e so small
 ; Queequeg no kill - e so small - e fish - e ; Queequeg kill - e big whale !"
 supper , till you began to look for fish - bones coming through your clothes
aw Hosea ' s brindled cow feeding on fish remnants , and marching along the sa
whale - boat ? did you ever strike a fish ?" Without saying a word , Queequeg
aw , the whale is declared " a royal fish ."* Oh , that ' s only nominal ! The
n ; nor for persisting in fighting a fish that too much persisted in fighting
matter of whales ; he followed these fish for the fun of it ; and a three year
ns a moot point whether a whale be a fish . In his System of Nature , A . D .
 hereby separate the whales from the fish ." But of my own knowledge , I know
fashioned ground that the whale is a fish , and call upon holy Jonah to back m
ect does the whale differ from other fish . Above , Linnaeus has given you tho
and warm blood ; whereas , all other fish are lungless and cold blooded . Next
```

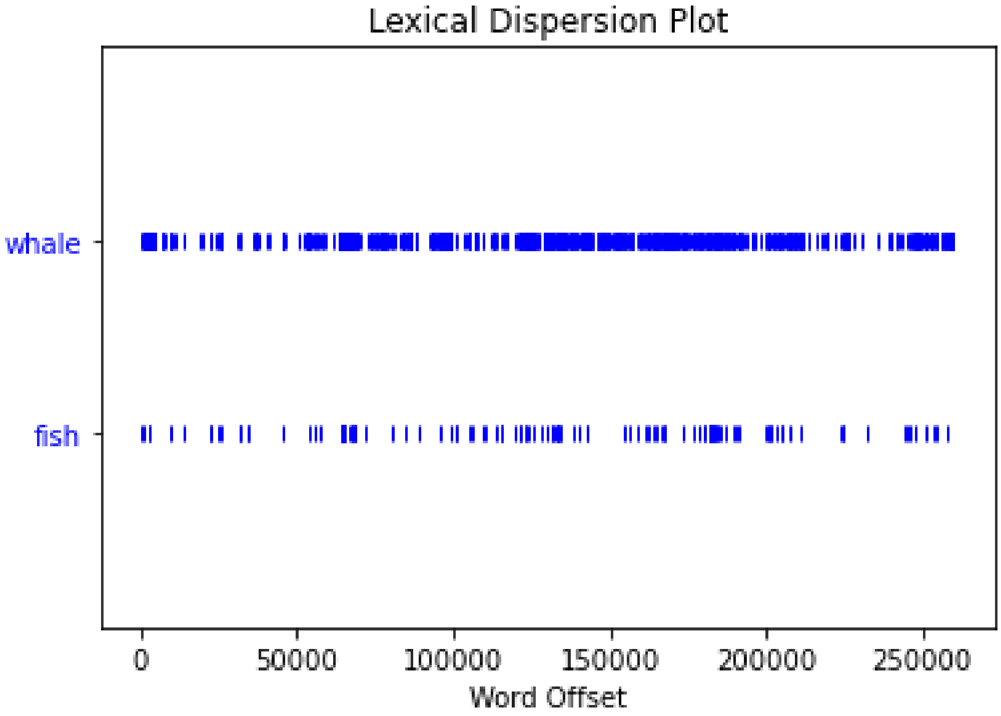
[ ]:
```
text1.similar("fish")
```

```
whale boat ship wind sea way captain line body world man mate time
carpenter leviathan thing crew chase harpoon pequod
```

[ ]:
```
text1.common_contexts(["fish"])
```

```
fast_what loose_what the_s the_and fast_and loose_and great_to e_e
a_that these_for other_are spouting_with sword_and the_all loose_is
a_is whale_is royal_which fatter_than sword_inn
```

[ ]:
```
text1.dispersion_plot(["whale", "fish"])
```



1.2 Basic operations

Length, uniqueness (diversity), sentence operations, …

```
[ ]: print("Total tokens: {}".format(len(text1)))
```

```
Total tokens: 260819
```

```
[ ]: print("Unique tokens: {}".format(len(set(text1))))
```

```
Unique tokens: 19317
```

```
[ ]: text1[0:10]
```

```
[ ]: ['[',
      'Moby',
      'Dick',
      'by',
      'Herman',
      'Melville',
      '1851',
      ']',
      'ETYMOLOGY',
      '.']
```

```
[ ]: text1.index("Ishmael")
```

```
[ ]: 4714
```

```
[ ]: text1[4710:4720]
```

```
[ ]: ['Loomings', '.', 'Call', 'me', 'Ishmael', '.', 'Some', 'years', 'ago', '--']
```

1.3 Conditional Frequency Distribution

Using the state-of-the-union corpus

```
[ ]: from nltk.corpus import state_union

     state_union.fileids()
```

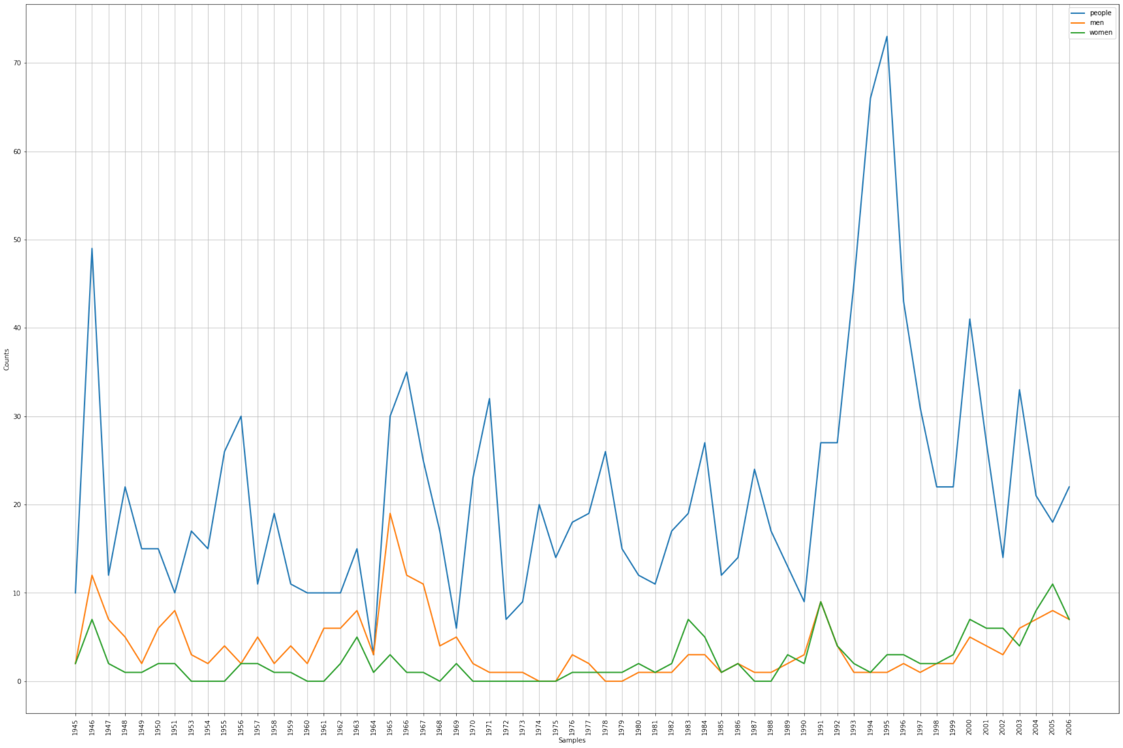
```
[ ]: ['1945-Truman.txt',
      '1946-Truman.txt',
      '1947-Truman.txt',
      '1948-Truman.txt',
      '1949-Truman.txt',
      '1950-Truman.txt',
      '1951-Truman.txt',
      '1953-Eisenhower.txt',
      '1954-Eisenhower.txt',
      '1955-Eisenhower.txt',
      '1956-Eisenhower.txt',
      '1957-Eisenhower.txt',
      '1958-Eisenhower.txt',
      '1959-Eisenhower.txt',
      '1960-Eisenhower.txt',
      '1961-Kennedy.txt',
      '1962-Kennedy.txt',
      '1963-Johnson.txt',
      '1963-Kennedy.txt',
      '1964-Johnson.txt',
      '1965-Johnson-1.txt',
      '1965-Johnson-2.txt',
      '1966-Johnson.txt',
      '1967-Johnson.txt',
      '1968-Johnson.txt',
      '1969-Johnson.txt',
      '1970-Nixon.txt',
      '1971-Nixon.txt',
      '1972-Nixon.txt',
      '1973-Nixon.txt',
      '1974-Nixon.txt',
      '1975-Ford.txt',
      '1976-Ford.txt',
      '1977-Ford.txt',
      '1978-Carter.txt',
      '1979-Carter.txt',
      '1980-Carter.txt',
      '1981-Reagan.txt',
      '1982-Reagan.txt',
      '1983-Reagan.txt',
      '1984-Reagan.txt',
      '1985-Reagan.txt',
      '1986-Reagan.txt',
      '1987-Reagan.txt',
      '1988-Reagan.txt',
      '1989-Bush.txt',
      '1990-Bush.txt',
      '1991-Bush-1.txt',
      '1991-Bush-2.txt',
      '1992-Bush.txt',
      '1993-Clinton.txt',
      '1994-Clinton.txt',
      '1995-Clinton.txt',
      '1996-Clinton.txt',
      '1997-Clinton.txt',
      '1998-Clinton.txt',
      '1999-Clinton.txt',
      '2000-Clinton.txt',
      '2001-GWBush-1.txt',
      '2001-GWBush-2.txt',
      '2002-GWBush.txt',
      '2003-GWBush.txt',
      '2004-GWBush.txt',
      '2005-GWBush.txt',
      '2006-GWBush.txt']
```

[ ]:

```python
from nltk.corpus import state_union

# increase default plot size
import matplotlib.pyplot as plt
plt.figure(figsize=(30, 20))

cfd = nltk.ConditionalFreqDist(
    (target, fileid[:4])
    # For each file
    for fileid in state_union.fileids()
    # Find all the words
    for w in state_union.words(fileid)
    for target in ['men', 'women', 'people']
    # filter out so we only return words in target
    if w.lower() == target)
cfd.plot()
```



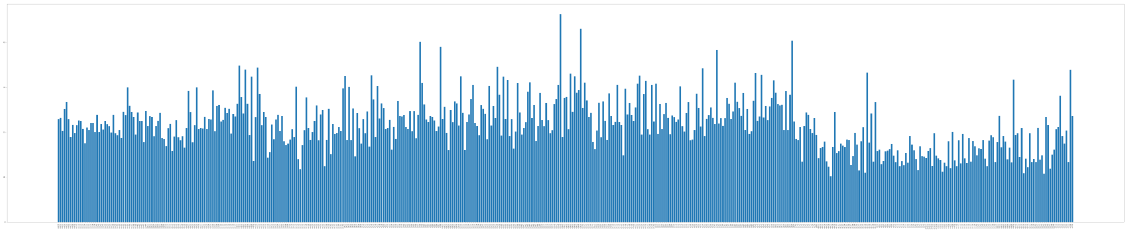
```
[ ]: <AxesSubplot:xlabel='Samples', ylabel='Counts'>
```

```
[ ]: # FileIds are found by nltk.corpus.[corpus_name].fileids(), this is the names␣
      ↪of the files within the corpus
     fileids = nltk.corpus.brown.fileids()
     output = {}
     for ID in fileids:
         # Getting a given document can be done as following : nltk.corpus.
      ↪[corpus_name].[split_type](fileids=[ID])
         sentences = nltk.corpus.brown.sents(fileids=ID)
         average_length = sum([len(sent) for sent in sentences])/len(sentences)
         output[ID] = average_length

     # Equal length lists to represent the values in x and y directions
     x_axis = list(output.keys())
     y_axis = list(output.values())

     plt.figure(figsize=(100, 20))
     # using a bar graph, you can use .plot to get points or lines if applicable
     plt.bar(x_axis, y_axis)
     plt.xticks(rotation=90)

     # Show the graph
     plt.show()
```



```
[ ]:
```