CS2021
Week 5
Counters
Collections Module

A set of simple to use, high-performance container classes .
Collections.Counter class:
An easy to use container fori multisets, and based /sublass of a dict. They are useful for determining frequencies in collections.

Technically, Counter is a dict subclass for counting any collection of hashable objects.

```
>>> from collections import Counter
>>> c = Counter()
>>> c
Counter()
>>> d = Counter("abracadabra")
>>> d
Counter({'a': 5, 'r': 2, 'b': 2, 'c':
```

```
1, 'd': 1})
```

Listing, Adding and Subtracting

```
>>> list(d.elements())
['r', 'r', 'c', 'd', 'b', 'b', 'a',
'a', 'a', 'a', 'a']
>> e=Counter("Ajji Majji la Tarajji")
>>> e
Counter({'j': 6, 'a': 4, ' ': 3, 'i':
3, 'r': 1, 'A': 1, 'M': 1, 'T': 1, 'l':
1})
>>> e+d
Counter({'a': 9, 'j': 6, 'r': 3, ' ':
3, 'i': 3, 'b': 2, 'c': 1, 'T': 1, 'd':
1, 'l': 1, 'A': 1, 'M': 1})
>>> f=e-d; f
Counter({'j': 6, ' ': 3, 'i': 3, 'A':
1, 'M': 1, 'l': 1, 'T': 1})
>>> g=d-e;  g
Counter({'b': 2, 'r': 1, 'a': 1, 'c':
1, 'd': 1})
```

Updating Counters

```
import collections
```

```
c = collections.Counter()
print ('Initial :', c)

c.update('abcdaab')
print ('Sequence:', c)

c.update({'a':1, 'd':5})
print ('Dict    :', c)
```

Accessing Counts

Once a Counter is populated, its values can be retrieved using the dictionary API.

```
import collections
c = collections.Counter('abracadabra’)
for letter in 'abcde’:
    print ('%s : %d' % (letter,
c[letter]), end=‘ | ‘)
>>>
a : 5 | b : 2 | c : 1 | d : 1 | e : 0 |
```

Note that Counter does not raise KeyError

for unknown items. If a value has not been seen in the input (as with e in this example), its count is 0.

most_common()

Use most_common(n) to produce a sequence of the n most frequently encountered input values and their respective counts.

```
import collections
c =
collections.Counter()
with open('/usr/share/
dict/words', ’r') as f:
    for line in f:
```

```
c.update(line.rstrip().lower())

print 'Most common:'
for letter, count in c.most_common(5):
    print('%s: %7d' % (letter, count))
```

Finding the most common word in file

```
file = open('/Users/fred/debate.txt', 'r')
text = file.read().lower()
file.close()

# replaces anything that is not a lowercase letter, a space, or an apostrophe with a space:
import re
```

```
text = re.sub('[^a-z\ \']+', ' ', text)
wordlist = list(text.split())

import collections
c= collections.Counter(wordlist)
print(c.most_common(10))
```

Removing stop words

```
sfile = open('/Users/fred/
stopwords.txt', 'r')
stext = sfile.read().lower()
sfile.close()
stext = re.sub('[^a-z\ \']+', ' ',
stext)
stopwordlist = list(stext.split())

import collections
c= collections.Counter()
for word in stopwordlist:
          del c[word]
print(c.most_common(10))
>>> [('tapper', 243), ('people', 205),
('trump', 181), ("i'm", 149), ('think',
```

```
146), ('want', 142), ('know', 139),
('governor', 134), ('one', 130),
('going', 121)]
```

Extracting Text from Web

```
from urllib.request import Request, urlopen
from urllib.error import URLError, HTTPError
req = Request('http://www.gutenberg.org/files/
55/55-h/55-h.htm’)
try:
    response = urlopen(req)
except HTTPError as e:
    print('The server couldn\'t fulfill the
request.')
    print('Error code: ', e.code)
except URLError as e:
    print('We failed to reach a server.')
    print('Reason: ', e.reason)
else:
    print ("Everthing is fine”)
html=response.read()
html=html.decode()
print (html[:4000])
```

Homework #5: Extracting Text and

Analyzing

Download a text file of a significant literary workk, e.g.,Wizard of Oz or Shakespeare's Sonnets https://archive.org/stream/shakespearessonn01041gut/wssnt10.txt

And create a program that opens the file and using the Counter() data type determines the 10 most frequent words that are not stopwords.