## HU Extension Assignment 12 E63 Big Data Analytics

### Handed out: 04/23/2016 Due by 11:30PM EST on Friday, 04/29/2016

Please, describe every step of your work and present all intermediate and final results in a Word document. Please, copy past text version of all essential command and snippets of results into the Word document with explanations of the purpose of those commands. We cannot retype text that is in JPG images. Please, always submit a separate copy of the original, working scripts and/or class files you used. Sometimes we need to run your code and retyping is too costly. Please include in your MS Word document only relevant portions of the console output or output files. Sometime either console output or the result file is too long and including it into the MS Word document makes that document too hard to read. PLEASE DO NOT EMBED files into your MS Word document. For issues and comments visit the class Discussion Board. If you use some other language other than Python in your daily work with NLP, please be free to use that language and a framework of your choice to do this assignment.

**Problem 1.** Create atable displaying **relative** frequencies with which “modals” (can, could, may, might, will, would and should) are used in 18 texts provided by NLTK in their extract from Gutenberg Corpus. For two modals with the largest span of relative frequencies (most used minus least used), select a text which uses it the most and the text that uses it the least. Compare usage in both texts by examining the concordances of those modals in two texts. Perhaps try to understand how are those words used in different texts.

First I installed nltk package using pip package manager.

*Note: I had already done this step, so the below screenshot shows that the package is installed*

rpulekar-m1:nltk\_workspace rpulekar$ sudo pip3 install -U nltk

Requirement already up-to-date: nltk in /Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-packages

rpulekar-m1:nltk\_workspace rpulekar$

Then I started the python shell:

rpulekar-m1:nltk\_workspace rpulekar$ python3

Python 3.5.1 (v3.5.1:37a07cee5969, Dec 5 2015, 21:12:44)

[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin

Type "help", "copyright", "credits" or "license" for more information.

>>>

Then I imported nltk package:

>>> import nltk

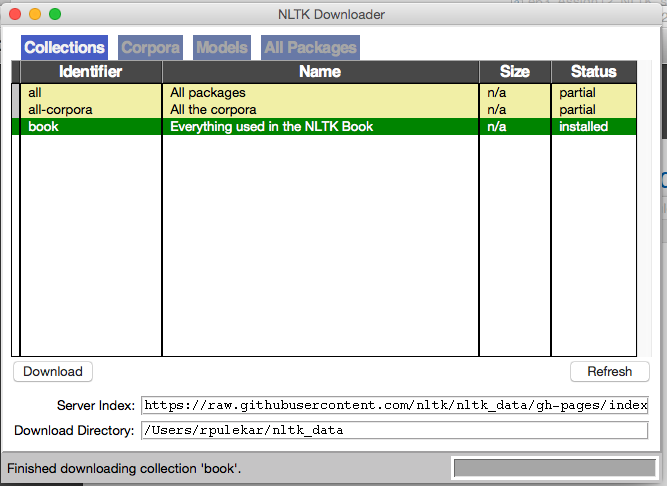
>>>

Then I downloaded nltk sample data by:

>>> nltk.download()

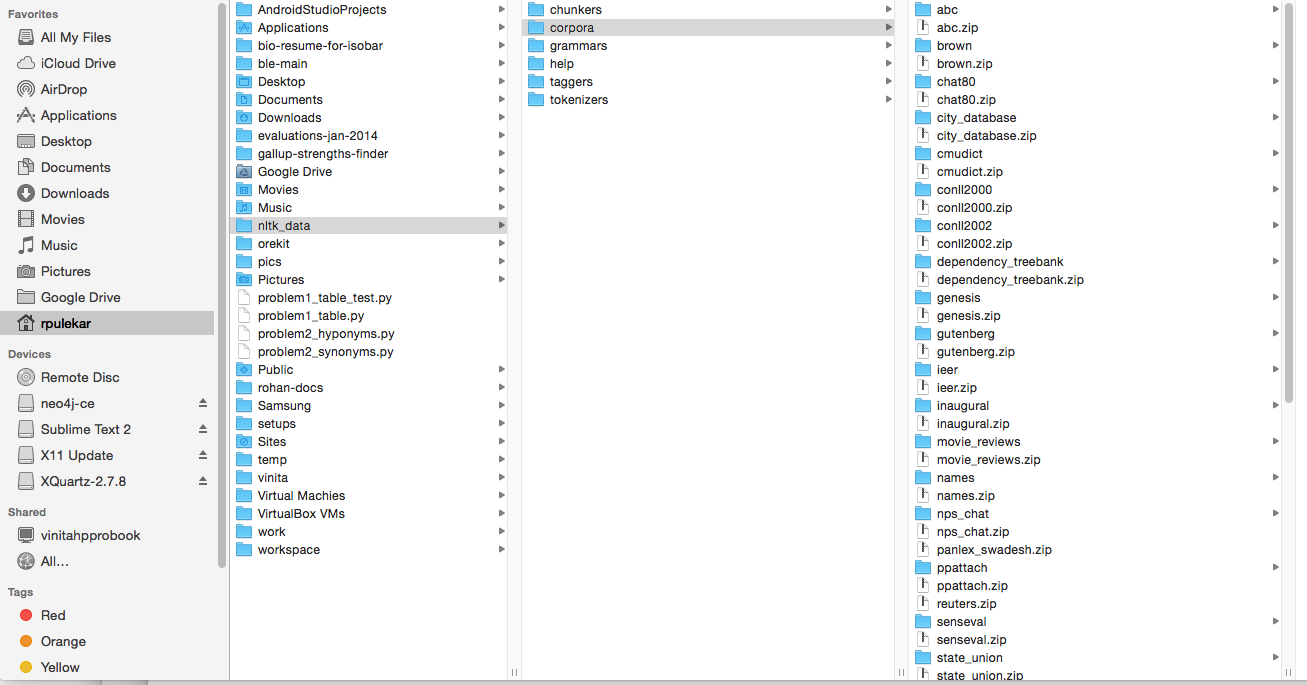
showing info https://raw.githubusercontent.com/nltk/nltk\_data/gh-pages/index.xml

True….



I then downloaded the book corpora.

This is where the files are downloaded:



Then I imported the book collection:

>>> 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

>>>

To find relative frequencies of modals in Gutenberg corpus, I used 2 approaches:

Approach 1(using nltk.FreqDist )

# the below import is to make sure decimal numbers are shown after division

from \_\_future\_\_ import division

# import gutenberg corpus

from nltk.corpus import gutenberg

print("\nRelative frequencies AS FRACTIONS (not percentage) of modals in gutenberg text")

print("\ncan\tcould\tmay\tmight\twill\twould\tshould")

# loop for each text of gutenberg

for fileid in gutenberg.fileids():

# get frequency distribution in that file

fdist = nltk.FreqDist(gutenberg.words(fileid))

# get relative frequency of each modal

relative\_frequency\_of\_can = round((fdist["can"])/(len(gutenberg.words(fileid))), 4)

relative\_frequency\_of\_could = round((fdist["could"])/(len(gutenberg.words(fileid))), 4)

relative\_frequency\_of\_may = round((fdist["may"])/(len(gutenberg.words(fileid))), 4)

relative\_frequency\_of\_might = round((fdist["might"])/(len(gutenberg.words(fileid))), 4)

relative\_frequency\_of\_will = round((fdist["will"])/(len(gutenberg.words(fileid))), 4)

relative\_frequency\_of\_would = round((fdist["would"])/(len(gutenberg.words(fileid))), 4)

relative\_frequency\_of\_should = round((fdist["should"])/(len(gutenberg.words(fileid))), 4)

# print the relative frequencies

print("{}\t{}\t{}\t{}\t{}\t{}\t{}\t{}".format(relative\_frequency\_of\_can,relative\_frequency\_of\_could,relative\_frequency\_of\_may,relative\_frequency\_of\_might,relative\_frequency\_of\_will,relative\_frequency\_of\_would,relative\_frequency\_of\_should,fileid))

Approach 2 (using nltk.ConditionalFreqDist):

# the below import is to make sure decimal numbers are shown after division

from \_\_future\_\_ import division

# import gutenberg corpus

from nltk.corpus import gutenberg

print("\nRelative frequencies AS FRACTIONS (not percentage) of modals in gutenberg text\n")

conditional\_fdist = nltk.ConditionalFreqDist(

(fileid, word)

for fileid in gutenberg.fileids()

for word in gutenberg.words(fileid))

modals = ["can", "could", "may", "might", "will", "would", "should"]

#conditional\_fdist.tabulate(samples=modals)

print("%-30s" % "Category", end="")

for modal in modals: # column headings

print("%10s" % modal, end="")

print()

for category in gutenberg.fileids():

print("%-30s" % category, end="") # row heading

for modal in modals: # for each word

relative\_freq\_of\_modal = round(conditional\_fdist[category][modal]/conditional\_fdist[category].N(), 4)

print("%10s" % format(float(relative\_freq\_of\_modal), '.4f'), end="") # print table cell

print()

Then I executed the script in this way:

>>> exec(open("/Users/rpulekar/problem1\_with\_conditional\_fDistribution.py").read())

Relative frequencies AS FRACTIONS (not percentage) of modals in gutenberg text

Category can could may might will would should

austen-emma.txt 0.0014 0.0043 0.0011 0.0017 0.0029 0.0042 0.0019

austen-persuasion.txt 0.0010 0.0045 0.0009 0.0017 0.0017 0.0036 0.0019

austen-sense.txt 0.0015 0.0040 0.0012 0.0015 0.0025 0.0036 0.0016

bible-kjv.txt 0.0002 0.0002 0.0010 0.0005 0.0038 0.0004 0.0008

blake-poems.txt 0.0024 0.0004 0.0006 0.0002 0.0004 0.0004 0.0007

bryant-stories.txt 0.0013 0.0028 0.0003 0.0004 0.0026 0.0020 0.0007

burgess-busterbrown.txt 0.0012 0.0030 0.0002 0.0009 0.0010 0.0024 0.0007

carroll-alice.txt 0.0017 0.0021 0.0003 0.0008 0.0007 0.0021 0.0008

chesterton-ball.txt 0.0014 0.0012 0.0009 0.0007 0.0020 0.0014 0.0008

chesterton-brown.txt 0.0015 0.0020 0.0005 0.0008 0.0013 0.0015 0.0007

chesterton-thursday.txt 0.0017 0.0021 0.0008 0.0010 0.0016 0.0017 0.0008

edgeworth-parents.txt 0.0016 0.0020 0.0008 0.0006 0.0025 0.0024 0.0013

melville-moby\_dick.txt 0.0008 0.0008 0.0009 0.0007 0.0015 0.0016 0.0007

milton-paradise.txt 0.0011 0.0006 0.0012 0.0010 0.0017 0.0005 0.0006

shakespeare-caesar.txt 0.0006 0.0007 0.0014 0.0005 0.0050 0.0015 0.0015

shakespeare-hamlet.txt 0.0009 0.0007 0.0015 0.0007 0.0035 0.0016 0.0014

shakespeare-macbeth.txt 0.0009 0.0006 0.0013 0.0002 0.0027 0.0018 0.0018

whitman-leaves.txt 0.0006 0.0003 0.0005 0.0002 0.0017 0.0005 0.0003

Then I did the calculation for largest span of frequencies and found that could and would have the largest relative frequency span.

could modal:

highest relative frequency: 0.0045 (austen-persuasion.txt)

lowest relative frequency: 0.0003 (whitman-leaves.txt)

relative frequency span: 0.0045 - 0.0003 = 0.0042

would modal:

highest relative frequency: 0.0042 (austen-emma.txt)

lowest relative frequency: 0.0004 (blake-poems.txt)

relative frequency span: 0.0042 - 0.0004 = 0.0038

To find concordance of could in 'austen-persuasion.txt':

>>> nltk.Text(gutenberg.words('austen-persuasion.txt')).concordance('could')

Displaying 25 of 451 matches:

every other leaf were powerless , he could read his own history with an interes

as still a very fine man . Few women could think more of their personal appeara

ersonal appearance than he did , nor could the valet of any new made lord be mo

l ; but it was only in Anne that she could fancy the mother to revive again . A

mild dark eyes from his own ), there could be nothing in them , now that she wa

ood looks of everybody else ; for he could plainly see how old all the rest of

self - possession and decision which could never have given the idea of her bei

heir , and whose strong family pride could see only in him a proper match for S

aronet from A to Z whom her feelings could have so willingly acknowledged as an

ing black ribbons for his wife , she could not admit him to be worth thinking o

were hereafter to be his own . This could not be pardoned . Such were Elizabet

alarm , set seriously to think what could be done , and had finally proposed t

l part of his estate that Sir Walter could dispose of ; but had every acre been

em , as anybody of sense and honesty could well be . She was a benevolent , cha

the most comprehensive retrenchments could secure , and saw no dignity in anyth

Russell ' s had no success at all : could not be put up with , were not to be

id not appear to him that Sir Walter could materially alter his style of living

ondon ; but Mr Shepherd felt that he could not be trusted in London , and had b

beyond their own circle . Sir Walter could not have borne the degradation of be

it a friendship quite out of place , could hint of caution and reserve . Lady R

. They will be all wanting a home . Could not be a better time , Sir Walter ,

d observed , Sir Walter ' s concerns could not be kept a secret ,)-- accidental

man who knew it only by description could feel ; and given Mr Shepherd , in hi

an extraordinary taste , certainly , could they have been supposed in the secre

than Admiral Croft bid fair to be , could hardly offer . So far went his under

>>>

To find concordance of could in 'whitman-leaves.txt':

>>> nltk.Text(gutenberg.words('whitman-leaves.txt')).concordance('could')

Displaying 25 of 52 matches:

ching it to me with full hands ; How could I answer the child ? I do not know w

ofs of mouths for nothing . I wish I could translate the hints about the dead y

the sun - rise would kill me , If I could not now and always send sun - rise o

fissure of the cliff . 32 I think I could turn and live with animals , they ar

he verge of a usual mistake . That I could forget the mockers and insults ! Tha

get the mockers and insults ! That I could forget the trickling tears and the b

f the bludgeons and hammers ! That I could look with a separate look on my own

bryo has never been torpid , nothing could overlay it . For it the nebula coher

you have come from yourself , if you could trace back through the centuries ?)

to look on you to touch you , For I could not die till I once look ' d on you

of myself , But I wonder ' d how it could utter joyous leaves standing alone t

thout its friend near , for I knew I could not , And I broke off a twig with a

nd a lover near , I know very well I could not . } To a Stranger Passing strang

r dialects , And it seems to me if I could know those men I should become attac

e , Fancying how happy you were if I could be with you and become your comrade

and all free poems also , I think I could stop here myself and do miracles , I

ve you not accepted ? What the study could not teach -- what the preaching coul

ould not teach -- what the preaching could not accomplish is accomplish ' d , i

rtist , And the laborers perceive he could labor with them and love them , No m

cient temples , sculptures classic , could none of them retain her ? Nor shades

, effulgently flowing forever . O I could sing such grandeurs and glories abou

guns . Pale , silent , stern , what could I say to that long - accrued retribu

to that long - accrued retribution ? Could I wish humanity different ? Could I

? Could I wish humanity different ? Could I wish the people made of wood and s

can never be destroy ' d , Here too could rise at last murdering and ecstatic

>>>

Concordances help in identifying the context in which a word is used.

For the modal word ‘could’, the concordance shows it is used more with word I (as in I could). This means the word ‘could’ is used more to describe the writer’s own probable past actions.

To find concordance of would in 'austen-emma.txt':

>>> nltk.Text(gutenberg.words('austen-emma.txt')).concordance('would')

Displaying 25 of 820 matches:

rk for her . The want of Miss Taylor would be felt every hour of every day . Sh

thing for herself as for them , and would have been a great deal happier if sh

hly deserves a good wife ;-- and you would not have had Miss Taylor live with u

k of her . It was very lucky , for I would not have had poor James think himsel

n your head , I know -- and what you would certainly say if my father were not

eeable to Emma herself , she knew it would be so much less so to her father , t

uch less so to her father , that she would not have him really suspect such a c

when so many people said Mr . Weston would never marry again , may comfort me f

eplied , " Ah ! my dear , I wish you would not make matches and foretell things

!-- Every body said that Mr . Weston would never marry again . Oh dear , no ! M

e it . Oh no ! Mr . Weston certainly would never marry again . Some people even

yourself one idle day , ` I think it would be a very good thing for Miss Taylor

p his house so comfortably , that it would be a shame to have him single any lo

ay , he looked so very much as if he would like to have the same kind office do

and importance , which the connexion would offend . Miss Churchill , however ,

ual to her situation than most girls would have been , and had sense , and ener

gy , and spirits that might be hoped would bear her well and happily through it

isposition and circumstances , which would make the approaching season no hindr

ying , " Ah , poor Miss Taylor ! She would be very glad to stay ." There was no

in their hands : but Mr . Woodhouse would never believe it . CHAPTER III Mr .

fit for any acquaintance but such as would visit him on his own terms . Fortuna

it taken place only once a year , it would have been a grievance . Mrs . Bates

elegance to be quite perfect . \_She\_ would notice her ; she would improve her ;

rfect . \_She\_ would notice her ; she would improve her ; she would detach her f

ce her ; she would improve her ; she would detach her from her bad acquaintance

>>>

To find concordance of would in 'blake-poems.txt':

>>> nltk.Text(gutenberg.words('blake-poems.txt')).concordance('would')

Displaying 5 of 5 matches:

swell . But , if at the Church they would give us some ale , And a pleasant fi

me Lurch , who is always at church , Would not have bandy children , nor fastin

ildren as pleasant and happy as he , Would have no more quarrel with the Devil

e - hearse . THE HUMAN ABSTRACT Pity would be no more If we did not make somebo

and therefore did I weep : That God would love a Worm I knew , and punish the

>>>

Concordances help in identifying the context in which a word is used.

The word ‘would’ is used quite a few times with word ‘not’ (as in ‘would not’). So this means that would is used more to express non-occurrences of events in the past.

Deliverables:

* problem1\_with\_fDistribution.py (python script to find relative frequencies using nltk.FreqDist class)
* problem1\_with\_conditional\_fDistribution.py (python script to find relative frequencies using nltk. ConditionalFreqDist class)

**Problem 2**. In the Inaugural corpus identify 10 most frequently used words longer than 7 characters. Which one of those has the largest number of synonyms? List all synonyms for those 10 words. Which one of those 10 words has the largest number of hyponyms? List all hyponyms of those 10 most frequently used “long” words.

This is how I identified 10 most frequently used words longer than 7 characters in inaugural corpus:

>>> from nltk.corpus import inaugural

>>> fdist = FreqDist([word.lower() for word in inaugural.words() if len(word) > 7])

>>> [str(i[0]) for i in fdist.most\_common(10)]

['government', 'citizens', 'constitution', 'national', 'american', 'congress', 'interests', 'political', 'executive', 'principles']

>>>

So the 10 most frequently used and longer than 7 chars words are:

'government', 'citizens', 'constitution', 'national', 'american', 'congress', 'interests', 'political', 'executive', 'principles'

To list the synonyms for all those words I used this script:

# import the inaugural corpus since we will do analysis on that corpus

from nltk.corpus import inaugural

# import the wordnet corpus since we will use that to find synonyms and hyponyms

from nltk.corpus import wordnet

# create Text object out of inaugural corpus words

all\_inaugural\_words\_text = nltk.Text(inaugural.words())

# create a frequency distribution of inaugural corpus words that are longer than 7 chars

fdist = nltk.FreqDist([word.lower() for word in all\_inaugural\_words\_text if len(word) > 7])

# loop through the 10 most common words

for fdist\_of\_ten\_common\_words in fdist.most\_common(10):

# get the word

word = fdist\_of\_ten\_common\_words[0]

# create a synonyms list

synonyms\_list = list()

# loop through the synonyms sets of this word

for synset in wordnet.synsets(word):

# add lemma names of that word to the synonyms list

synonyms\_list = synonyms\_list + synset.lemma\_names()

# convert the strings in synonyms list from unicode to utf-8 and to lower case

synonyms\_list = [str(synonym).lower() for synonym in synonyms\_list]

# get the unique synonyms and sort them

synonyms\_set = sorted(set(synonyms\_list))

# check if the word itself is present in the synonyms\_set

# this can happen as lemma\_names() output might include the word itself

if (word in synonyms\_set):

# if word is present in its synonyms set, remove it

synonyms\_set.remove(word)

# print synonyms for the word

print("\nSynonyms of '{}': {}".format(word, synonyms\_set))

print("Number of synonyms for '{}': {}".format(word, len(synonyms\_set)))

Then I executed the script in this way:

>>> exec(open("/Users/rpulekar/problem2\_synonyms.py").read())

Synonyms of 'government': ['administration', 'authorities', 'governance', 'governing', 'government\_activity', 'political\_science', 'politics', 'regime']

Number of synonyms for 'government': 8

Synonyms of 'citizens': ['citizen']

Number of synonyms for 'citizens': 1

Synonyms of 'constitution': ['composition', 'constitution\_of\_the\_united\_states', 'establishment', 'formation', 'fundamental\_law', 'make-up', 'makeup', 'old\_ironsides', 'organic\_law', 'organisation', 'organization', 'physical\_composition', 'u.s.\_constitution', 'united\_states\_constitution', 'us\_constitution']

Number of synonyms for 'constitution': 15

Synonyms of 'national': ['home', 'interior', 'internal', 'subject']

Number of synonyms for 'national': 4

Synonyms of 'american': ['american\_english', 'american\_language']

Number of synonyms for 'american': 2

Synonyms of 'congress': ['carnal\_knowledge', 'coition', 'coitus', 'copulation', 'intercourse', 'relation', 'sex\_act', 'sexual\_congress', 'sexual\_intercourse', 'sexual\_relation', 'u.s.\_congress', 'united\_states\_congress', 'us\_congress']

Number of synonyms for 'congress': 13

Synonyms of 'interests': ['concern', 'interest', 'interest\_group', 'interestingness', 'involvement', 'matter\_to', 'occupy', 'pastime', 'pursuit', 'sake', 'stake', 'worry']

Number of synonyms for 'interests': 12

Synonyms of 'political': []

Number of synonyms for 'political': 0

Synonyms of 'executive': ['administrator', 'executive\_director']

Number of synonyms for 'executive': 2

Synonyms of 'progress': ['advance', 'advancement', 'build', 'build\_up', 'come\_along', 'come\_on', 'forward\_motion', 'get\_along', 'get\_on', 'go\_on', 'march\_on', 'move\_on', 'onward\_motion', 'pass\_on', 'procession', 'progression', 'shape\_up', 'work\_up']

Number of synonyms for 'progress': 18

>>>

So above is the list of synonyms for the 10 most frequent (longer than 7 characters) words.

The word progress has the most number of synonyms: 18

To find hyponyms I used this script:

# import the inaugural corpus since we will do analysis on that corpus

from nltk.corpus import inaugural

# import the wordnet corpus since we will use that to find synonyms and hyponyms

from nltk.corpus import wordnet

# create Text object out of inaugural corpus words

all\_inaugural\_words\_text = nltk.Text(inaugural.words())

# create a frequency distribution of inaugural corpus words that are longer than 7 chars

fdist = nltk.FreqDist([word.lower() for word in all\_inaugural\_words\_text if len(word) > 7])

# loop through the 10 most common words

for fdist\_of\_ten\_common\_words in fdist.most\_common(10):

# get the word

word = fdist\_of\_ten\_common\_words[0]

# create a hynonyms list

hyponyms\_list = list()

# loop through the synonyms sets of this word

for synset in wordnet.synsets(word):

# loop through the hyponyms of the synonym set

for hyponym\_synset in synset.hyponyms():

# add lemma names of this hyponym set to the list of hyponyms

hyponyms\_list = hyponyms\_list + hyponym\_synset.lemma\_names()

# convert the strings in hyponyms list from unicode to utf-8 and to lower case

hyponyms\_list = [str(hyponym).lower() for hyponym in hyponyms\_list]

# get the unique synonyms and sort them

hyponyms\_set = sorted(set(hyponyms\_list))

# print hyponyms for the word

print("\nHyponyms of '{}': {}".format(word, hyponyms\_set))

print("Number of hyponyms for '{}': {}".format(word, len(hyponyms\_set)))

Then I executed this script in this way:

>>> exec(open("/Users/rpulekar/problem2\_hyponyms.py").read())

Hyponyms of 'government': ['ancien\_regime', 'authoritarian\_regime', 'authoritarian\_state', 'bureaucracy', 'court', 'downing\_street', 'empire', 'federal\_government', 'geopolitics', 'government-in-exile', 'lawmaking', 'legislating', 'legislation', 'local\_government', 'military\_government', 'misgovernment', 'misrule', 'palace', 'papacy', 'pontificate', 'practical\_politics', 'pupet\_regime', 'puppet\_government', 'puppet\_state', 'realpolitik', 'royal\_court', 'state', 'state\_government', 'stratocracy', 'totalitarian\_state', 'totalitation\_regime', 'trust\_busting']

Number of hyponyms for 'government': 32

Hyponyms of 'citizens': ['active\_citizen', 'civilian', 'elector', 'freeman', 'freewoman', 'private\_citizen', 'repatriate', 'thane', 'voter']

Number of hyponyms for 'citizens': 9

Hyponyms of 'constitution': ['collectivisation', 'collectivization', 'colonisation', 'colonization', 'communisation', 'communization', 'federation', 'genetic\_constitution', 'genotype', 'grain', 'karyotype', 'phenotype', 'settlement', 'structure', 'texture', 'unionisation', 'unionization']

Number of hyponyms for 'constitution': 17

Hyponyms of 'national': ['citizen', 'compatriot', 'nationalist', 'patriot']

Number of hyponyms for 'national': 4

Hyponyms of 'american': ['aave', 'african-american', 'african\_american', 'african\_american\_english', 'african\_american\_vernacular\_english', 'afro-american', 'alabaman', 'alabamian', 'alaskan', 'anglo-american', 'appalachian', 'arizonan', 'arizonian', 'arkansan', 'arkansawyer', 'asian\_american', 'badger', 'bay\_stater', 'beaver', 'black\_american', 'black\_english', 'black\_english\_vernacular', 'black\_vernacular', 'black\_vernacular\_english', 'bluegrass\_stater', 'bostonian', 'buckeye', 'californian', 'carolinian', 'coloradan', 'connecticuter', 'cornhusker', 'creole', 'delawarean', 'delawarian', 'down\_easter', 'ebonics', 'floridian', 'franco-american', 'garden\_stater', 'georgian', 'german\_american', 'gopher', 'granite\_stater', 'hawaiian', 'hispanic', 'hispanic\_american', 'hoosier', 'idahoan', 'illinoisan', 'indianan', 'iowan', 'kansan', 'kentuckian', 'keystone\_stater', 'latin\_american', 'latino', 'louisianan', 'louisianian', 'mainer', 'marylander', 'mesoamerican', 'michigander', 'minnesotan', 'mississippian', 'missourian', 'montanan', 'nebraskan', 'nevadan', 'new\_englander', 'new\_hampshirite', 'new\_jerseyan', 'new\_jerseyite', 'new\_mexican', 'new\_yorker', 'nisei', 'north\_american', 'north\_carolinian', 'north\_dakotan', 'northerner', 'ohioan', 'oklahoman', 'oregonian', 'pennsylvanian', 'puerto\_rican', 'rhode\_islander', 'sooner', 'south\_american', 'south\_carolinian', 'south\_dakotan', 'southerner', 'spanish\_american', 'tarheel', 'tennessean', 'texan', 'tory', 'utahan', 'vermonter', 'virginian', 'volunteer', 'washingtonian', 'west\_indian', 'west\_virginian', 'wisconsinite', 'wolverine', 'wyomingite', 'yank', 'yankee', 'yankee-doodle']

Number of hyponyms for 'american': 109

Hyponyms of 'congress': ['ass', 'continental\_congress', 'criminal\_congress', 'defloration', 'fuck', 'fucking', 'hank\_panky', 'nookie', 'nooky', 'penetration', 'piece\_of\_ass', 'piece\_of\_tail', 'roll\_in\_the\_hay', 'screw', 'screwing', 'shag', 'shtup', 'unlawful\_carnal\_knowledge']

Number of hyponyms for 'congress': 18

Hyponyms of 'interests': ['absorb', 'avocation', 'behalf', 'by-line', 'charisma', 'color', 'colour', 'compound\_interest', 'concern', 'controlling\_interest', 'engage', 'engross', 'enthusiasm', 'equity', 'fascinate', 'fee', 'grip', 'grubstake', 'hobby', 'insurable\_interest', 'intrigue', 'news', 'newsworthiness', 'occupy', 'personal\_appeal', 'personal\_magnetism', 'pursuit', 'reversion', 'right', 'security\_interest', 'shrillness', 'sideline', 'simple\_interest', 'spare-time\_activity', 'special\_interest', 'spellbind', 'terminable\_interest', 'topicality', 'transfix', 'undivided\_interest', 'undivided\_right', 'vested\_interest', 'vividness']

Number of hyponyms for 'interests': 43

Hyponyms of 'political': []

Number of hyponyms for 'political': 0

Hyponyms of 'executive': ['bush\_administration', 'business\_executive', 'carter\_administration', 'clinton\_administration', 'commissioner', 'corporate\_executive', 'dci', 'director\_of\_central\_intelligence', 'government\_minister', 'minister', 'prefect', 'rainmaker', 'reagan\_administration', 'secretary\_general', 'surgeon\_general', 'triumvir', 'v.p.', 'vice\_president']

Number of hyponyms for 'executive': 18

Hyponyms of 'progress': ['career', 'clear\_sailing', 'climb', 'close\_in', 'creep\_up', 'draw\_in', 'easy\_going', 'edge', 'elapse', 'encroach', 'forge', 'forwarding', 'furtherance', 'glide\_by', 'go\_along', 'go\_by', 'head', 'headway', 'impinge', 'inch', 'infringe', 'lapse', 'leapfrog', 'life\_history', 'march', 'overhaul', 'overtake', 'pass', 'penetrate', 'plain\_sailing', 'plough\_on', 'press\_on', 'promotion', 'push', 'push\_on', 'rachet\_up', 'ratchet', 'ratchet\_down', 'slide\_by', 'slip\_away', 'slip\_by', 'sneak\_up', 'stride', 'string', 'string\_along', 'work\_flow', 'workflow']

Number of hyponyms for 'progress': 47

>>>

The word ‘american’ has the most number of hyponyms: 109

Deliverables:

* problem2\_synonyms.py (python script to find synonyms of 10 most frequently larger than 7 chars words)
* problem2\_hyponyms.py (python script to find hyponyms of 10 most frequently larger than 7 chars words)

**~~Problem 3.~~** ~~Create for us one graph displaying cumulative word length distribution for six different genres in Brown corpus. Create a tabular display of basic word statistics for all genres in Brown corpus. Include: average word length, average sentence length, number of concurrences in each genre, percentage of the text consumed by conditional words: would, could and should.~~

You literature for this assignment are chapters 1 and 2 of Natural Language Processing with Python book by Steven Bird et al.