

Assignment 1

Prof. Lu Xiao

Name: Raghav Raheja

Course: IST664

1. Analysis of State of the Union Addresses dataset: Description

The corpus texts “state_union_part1”, “state_union_part2”, “state_union_policy” belong to one publication format, namely Gutenberg eBook. It covers State of Union addresses between different time periods. The three documents in whole acts as a complete eBook of all the addresses between 1790 to 2016 and their policies related to licensing of the eBook. However, the state union addresses between 1860 to 1946 is missing from the corpus. “state_union_part1” covers addresses from 1790 to 1860 which were released in 2004 and last updated in June, 2007. The format of the document starting with a preface, title with the edition information followed by an index and then finally with the actual speeches. Every State of Union Address is separated by three asterisks. All the addresses in the corpus covers a common format starting with what kind of speech that is, followed by the name of the person giving the speech and finally the date on which the speech was made. The document has 510561 words, out of which 11586 are unique words, making it almost 2.269%. The second document “state_union_part2” follows a similar approach in terms of the structure of the document as well as the documentation of the addresses. The document contains 423466 words, whereas 13323 are unique words, which make it 3.146 % unique words. This tells us that document 2 has a higher percentage of unique words. The third document “state_union_policy” covers the copyright licenses and instruction on using the Gutenberg eBook as well as its distribution among the groups.

One of the reason for the huge corpus is to make semantic study possible. The document contain everything from the preface to the license. The text in the corpus is the complete text (Complete Speeches) and not a part of small cut out text from the chunk The document “state_union_part1” covers speeches which were made during the time US was going through a transformation in terms of the war, revolution, collaboration with new countries; whereas the document “state_union_part2” has speeches of the time when terrorism was on the high.

2. Analysis of State of the Union Addresses dataset: Part1

In the first step I converted the whole corpus into tokens using **nltk.word_tokenize** to work on the individual characters/words instead of the complete text. The next step I performed was to covert tokens into lower case as I did not want the capital words to be considered as a completely different word (in comparison to its lower-case word). Furthermore, I removed the non-characters using **is.alpha** to focus just on non- character words; I did not use the customized function we used in the lab as I was fine with hyphen words or apostrophe getting separated, I was interested in knowing just the word. The final step before the calculation, I removed the stop words as for the assignment, we are more interested in words other than the stop words.

3. Analysis of State of the Union Addresses dataset: Part2

In the first step I converted the whole corpus into tokens using **nltk.word_tokenize** to work on the individual characters/words instead of the complete text. The next step I performed was to

Assignment 1

Prof. Lu Xiao

Name: Raghav Raheja

Course: IST664

covert tokens into lower case as I did not want the capital words to be considered as a completely different word (in comparison to its lower-case word). Furthermore, I removed the non-characters using **is.alpha** to focus just on non-character words; I did not use the customized function we used in the lab as I was fine with hyphen words or apostrophe getting separated, I was interested in knowing just the word. The final step before the calculation, I removed the stop words as for the assignment, we are more interested in words other than the stop words.

4. Comparison

- a. How are state_union_part1 and state_union_part2 similar or different in the use of the language, based on your results? Why?

The similarity in the two is the use of the country name and discussion on the monetary front. However, the differences in the two stand in terms of the assertion. “State_union_part1” uses a couple of words like ‘may, would’, which shows a bit of uncertainty. On the other hand “State_union_part2” uses words like ‘must, new’, which are more positive and certain. The other difference that can be comprehended from the two is that in part1, it talks a lot about different nations and the war and the issue in terms of money. While in part2, it talks more on the security aspects and the presence of the terrorists’ organisations.

- b. Are there any problems with the word or bigram lists that you found? Could you get a better list of bigrams?

I think that the bigrams that I found are very informative. However, I feel some of the bigrams like individual names and country names like “Unites States, Great Britain, Porto Rico” if considered as a single word would give us a better idea of what words are most commonly used with the specific country name or person name.

- c. How are the top 50 bigrams by frequency different from the top 50 bigrams scored by Mutual Information?

The bigram by frequency is the calculation of one word after the other. For example, if there is a word like “I like”, this will calculate the number of times “like” came after “I” and then divide it by the number of times “I” was there in the corpus. It is kind of a fraction of times a word came after a specific word. The top 50 will show the highest fraction of the same. In the other concept of Mutual Information, for the probability to be high or for the result to be high, both the words should come together maximum number of times. For example, if “I” separately come 100 times in a corpus and “like” comes 100 times in a corpus and together they come 90 times in the corpus, it will have very high result. On the other and if “I” comes 100 times and “like” comes 100 times and they come together 50 times, the probability will be low. If we consider our corpus, we will see that 50 bigrams by frequency consists of words that have come together

Assignment 1

Prof. Lu Xiao

Name: Raghav Raheja

Course: IST664

most often, whereas the 50 bigrams by mutual Information consist of the names, countries or organisation's name and not necessarily the English language words.

Appendix

50 Words by frequency (Normalized)

('states', 0.011439774311094691)	('must', 0.007389319075155003)
('government', 0.009319742741515676)	('people', 0.006835574033896459)
('united', 0.00782522543702037)	('world', 0.006762951733403535)
('may', 0.006557404577588958)	('new', 0.006540545938143956)
('congress', 0.0062971234739970785)	('america', 0.0057689339954066395)
('upon', 0.0061082097697771655)	('year', 0.005741700632721793)
('would', 0.00579755167839331)	('congress', 0.005582839350393522)
('public', 0.005772363184497322)	('us', 0.005519294837462213)
('country', 0.004882369733505735)	('government', 0.005042710990477401)
('great', 0.00450454232506591)	('years', 0.005042710990477401)
('made', 0.004454165337273933)	('american', 0.004311949091767355)
('state', 0.004386996020217964)	('nation', 0.003907987545275465)
('last', 0.0038244529898742254)	('one', 0.0036492705997694243)
('war', 0.0035012006515423755)	('every', 0.0035403371490300385)
('present', 0.0034088428405904185)	('make', 0.003531259361468423)
('time', 0.0033920505113264263)	('work', 0.0034223259107290373)
('people', 0.003299692700374469)	('federal', 0.0033769369729209596)
('year', 0.003295494618058471)	('time', 0.0033633202915785364)
('power', 0.0031233732431025507)	('states', 0.0032271534781543043)
('citizens', 0.003035213514466592)	('americans', 0.0031227589211957263)
('subject', 0.002984836526674615)	('help', 0.003113681133634111)
('shall', 0.002913469127302648)	('security', 0.003109142239853303)
('without', 0.0027833285755067085)	('war', 0.0030592144082644177)
('union', 0.0026993669291867474)	('economic', 0.0030455977269219945)
('act', 0.0026321976121307785)	('peace', 0.0030319810455795714)
('treaty', 0.0026196033651827845)	('united', 0.0029548198513058396)
('one', 0.0026028110359187923)	('nations', 0.0029275864886209933)
('part', 0.0025944148712867964)	('also', 0.002900353125936147)
('mexico', 0.0025398398011788215)	('program', 0.0028958142321553393)
('general', 0.0025230474719148293)	('country', 0.002859503081908877)
('every', 0.002476868566438851)	('national', 0.0027641863125119144)

Assignment 1

Prof. Lu Xiao

Name: Raghav Raheja

Course: IST664

('treasury', 0.002476868566438851) ('necessary', 0.00241389733169888) ('constitution', 0.002338331850010915) ('new', 0.0023005491091669327) ('duty', 0.0022207855451629697) ('foreign', 0.002178804722002989) ('two', 0.0021410219811590064) ('commerce', 0.002124229651895014) ('nations', 0.0021074373226310224) ('peace', 0.002103239240315024) ('system', 0.002073852664103038) ('laws', 0.0020654564994710416) ('duties', 0.0020486641702070494) ('within', 0.0020108814293630668) ('law', 0.002002485264731071) ('us', 0.001943712112307098) ('interests', 0.0018933351245151214) ('interest', 0.0018639485483031351) ('amount', 0.001859750465987137)	('economy', 0.002668869543114952) ('great', 0.002646175074210913) ('last', 0.0025962472426220282) ('many', 0.0025553971985947583) ('free', 0.0025327027296907197) ('need', 0.002514547154567489) ('first', 0.002510008260786681) ('let', 0.00249185268566345) ('would', 0.0024873137918826425) ('state', 0.002360224766020026) ('tax', 0.002332991403335179) ('know', 0.002301219146869525) ('million', 0.002301219146869525) ('freedom', 0.002283063571746294) ('budget', 0.0022739857841846786) ('health', 0.0022195190588149855) ('future', 0.0021559745458836774) ('system', 0.0021015078205139843) ('programs', 0.0020969689267331766)
--	---

50 Bigrams by Frequency

(('united', 'states'), 0.007648905979748451) (('great', 'britain'), 0.0011502745545834663) (('last', 'session'), 0.0010159359204715286) (('public', 'debt'), 0.0007514567345636513) (('state', 'union'), 0.0007262682406676631) (('house', 'representatives'), 0.0006213161827677117) (('fiscal', 'year'), 0.0006045238535037195) (('union', 'address'), 0.0006045238535037195) (('report', 'secretary'), 0.0005835334419237293) (('public', 'lands'), 0.0005457507010797468) (('two', 'countries'), 0.0005121660425517623) (('present', 'year'), 0.0004449967254957935) (('within', 'limits'), 0.00041980823159980523) (('secretary', 'treasury'), 0.00041561014928380717) (('fellow', 'citizens'), 0.00040721398465181106) (('session', 'congress'), 0.00040721398465181106) (('act', 'congress'), 0.0003946197377038169)	(('united', 'states'), 0.0020969689267331766) (('state', 'union'), 0.001211884639475667) (('american', 'people'), 0.0010847956136130503) (('last', 'year'), 0.001021251100681742) (('fiscal', 'year'), 0.0008442342432302399) (('federal', 'government'), 0.0008351564556686245) (('social', 'security'), 0.000826078668107009) (('health', 'care'), 0.000807923092983778) (('let', 'us'), 0.0007988453054221626) (('years', 'ago'), 0.0007353007924908541) (('union', 'address'), 0.0006263673417514683) (('united', 'nations'), 0.0006127506604090451) (('billion', 'dollars'), 0.0005900561915050064) (('million', 'dollars'), 0.0005764395101625832) (('soviet', 'union'), 0.0005673617226009677) (('men', 'women'), 0.0005128949972312748) (('free', 'world'), 0.0004947394221080438)
---	--

Assignment 1

Prof. Lu Xiao

Name: Raghav Raheja

Course: IST664

(('general', 'government'), 0.00039042165538781884) (('year', 'ending'), 0.00039042165538781884) (('british', 'government'), 0.0003862235730718208) (('two', 'governments'), 0.0003736293261238266) (('citizens', 'united'), 0.0003610350791758325) (('federal', 'government'), 0.00035683699685983445) (('secretary', 'war'), 0.0003526389145438364) (('annual', 'message'), 0.0003400446675958422) (('public', 'service'), 0.00033584658527984417) (('senate', 'house'), 0.00033584658527984417) (('consideration', 'congress'), 0.00032325233833185) (('ending', 'june'), 0.0003148561736998539) (('last', 'annual'), 0.0003148561736998539) (('attention', 'congress'), 0.00031065809138385584) (('government', 'united'), 0.0003064600090678578) (('public', 'money'), 0.0002896676798038656) (('indian', 'tribes'), 0.00027707343285587145) (('mexican', 'government'), 0.0002728753505398734) (('part', 'united'), 0.0002728753505398734) (('treasury', 'notes'), 0.0002728753505398734) (('upon', 'subject'), 0.00026867726822387534) (('commercial', 'intercourse'), 0.0002644791859078773) (('several', 'states'), 0.0002644791859078773) (('secretary', 'state'), 0.0002602811035918792) (('provision', 'made'), 0.00024768685664388506) (('article', 'treaty'), 0.00023929069201188898) (('claims', 'citizens'), 0.00023929069201188898) (('address', 'december'), 0.00023509260969589092) (('new', 'mexico'), 0.00023509260969589092) (('favorable', 'consideration'), 0.00023089452737989286) (('naval', 'force'), 0.00023089452737989286) (('bank', 'united'), 0.0002266964450638948) (('people', 'united'), 0.0002266964450638948)	(('every', 'american'), 0.0004493504842999664) (('members', 'congress'), 0.00043119490917673545) (('economic', 'growth'), 0.0004266560153959277) (('middle', 'east'), 0.0004130393340535045) (('make', 'sure'), 0.0003994226527110813) (('free', 'nations'), 0.000385805971368658) (('first', 'time'), 0.00036765039624542706) (('four', 'years'), 0.00036765039624542706) (('state', 'local'), 0.00036311150246461934) (('ask', 'congress'), 0.00035857260868381156) (('armed', 'forces'), 0.0003404170335605806) (('world', 'war'), 0.0003404170335605806) (('must', 'continue'), 0.0003358781397797729) (('next', 'years'), 0.0003358781397797729) (('work', 'together'), 0.0003358781397797729) (('foreign', 'policy'), 0.0003177225646565419) (('new', 'jobs'), 0.0003177225646565419) (('two', 'years'), 0.00030410588331411867) (('vice', 'president'), 0.00030410588331411867) (('around', 'world'), 0.00029048920197169545) (('national', 'security'), 0.0002859503081908877) (('must', 'also'), 0.00028141141441007995) (('address', 'january'), 0.0002723336268484645) (('human', 'rights'), 0.0002677947330676567) (('health', 'insurance'), 0.000263255839286849) (('fellow', 'americans'), 0.0002541780517252335) (('fellow', 'citizens'), 0.0002541780517252335) (('past', 'year'), 0.0002541780517252335) (('past', 'years'), 0.0002541780517252335) (('states', 'america'), 0.0002541780517252335) (('civil', 'rights'), 0.00024510026416361806) (('urge', 'congress'), 0.00024510026416361806) (('young', 'people'), 0.00024510026416361806)
--	--

50 Bigrams by Mutual Information Scores (min frequency 5)

(('bona', 'fide'), 15.539910019165042) (('posse', 'comitatus'), 15.539910019165042) (('punta', 'arenas'), 15.539910019165042)	(('el', 'salvador'), 15.164265341881517) (('ladies', 'gentlemen'), 15.164265341881517) (('bin', 'laden'), 14.94187292054507)
---	--

Assignment 1

Prof. Lu Xiao

Name: Raghav Raheja

Course: IST664

(('ballot', 'box'), 15.276875613331246)	(('saudi', 'arabia'), 14.94187292054507)
(('del', 'norte'), 15.276875613331246)	(('sam', 'rayburn'), 14.749227842602672)
(('millard', 'fillmore'), 15.276875613331246)	(('jimmy', 'carter'), 14.42729974771531)
(('clayton', 'bulwer'), 14.861838114052404)	(('endowed', 'creator'), 14.316268435326567)
(('guadalupe', 'hidalgo'), 14.691913112610091)	(('northern', 'ireland'), 14.164265341881517)
(('porto', 'rico'), 14.691913112610091)	(('gerald', 'ford'), 14.097151146022979)
(('writ', 'mandamus'), 14.598803708218608)	(('floor', 'appears'), 14.012262248436468)
(('franklin', 'pierce'), 14.539910019165042)	(('iron', 'curtain'), 13.94187292054507)
(('la', 'plata'), 14.402406495415105)	(('grass', 'roots'), 13.901230936047725)
(('vera', 'cruz'), 14.276875613331246)	(('thomas', 'jefferson'), 13.785753718627788)
(('entangling', 'alliances'), 14.206486285439848)	(('sons', 'daughters'), 13.749227842602675)
(('seminaries', 'learning'), 14.013841207497453)	(('red', 'tape'), 13.749227842602673)
(('gun', 'boats'), 13.884558190552486)	(('jill', 'biden'), 13.678838514711277)
(('nucleus', 'around'), 13.861838114052404)	(('lyndon', 'johnson'), 13.661765001352334)
(('ruler', 'universe'), 13.861838114052404)	(('barack', 'obama'), 13.66176500135233)
(('costa', 'rica'), 13.8618381140524)	(('teen', 'pregnancy'), 13.57930284116036)
(('santa', 'anna'), 13.774375272802065)	(('abraham', 'lincoln'), 13.49183999991002)
(('santa', 'fe'), 13.774375272802065)	(('mom', 'dad'), 13.456446093374828)
(('van', 'buren'), 13.774375272802065)	(('empowerment', 'zones'), 13.356910419823912)
(('project', 'gutenberg'), 13.774375272802063)	(('william', 'clinton'), 13.327764074164396)
(('sublime', 'porte'), 13.732555097107436)	(('ronald', 'reagan'), 13.289796223965373)
(('tea', 'coffee'), 13.613910600608818)	(('synthetic', 'fuels'), 13.275296654270264)
(('martin', 'van'), 13.604450271359752)	(('greece', 'turkey'), 13.204907326378866)
(('ad', 'valorem'), 13.53991001916504)	(('elementary', 'secondary'), 13.122788705905355)
(('beacons', 'buoys'), 13.402406495415105)	(('intercontinental', 'ballistic'), 13.003273465209212)
(('water', 'witch'), 13.402406495415105)	(('feeding', 'hungry'), 12.967868129078013)
(('quincy', 'adams'), 13.402406495415104)	(('river', 'basins'), 12.8912468474751)
(('statute', 'book'), 13.338276157995391)	(('status', 'quo'), 12.891246847475099)
(('buenos', 'ayres'), 13.276875613331244)	(('commander', 'chief'), 12.856143046519184)
(('indiana', 'illinois'), 13.139372089581311)	(('prime', 'minister'), 12.842337246994152)
(('de', 'facto'), 13.128483773438575)	(('nationwide', 'radio'), 12.801695262496807)
(('franking', 'privilege'), 13.106950611888934)	(('reported', 'floor'), 12.764334734992882)
(('rocky', 'mountains'), 13.054483191994798)	(('radio', 'television'), 12.749227842602675)
(('andrew', 'jackson'), 12.972021031802825)	(('introduced', 'thomas'), 12.670276501207852)
(('retired', 'list'), 12.916979668244863)	(('project', 'gutenberg'), 12.661765001352334)
(('sooner', 'later'), 12.876945006442611)	(('dwight', 'eisenhower'), 12.643433178580077)
(('circulating', 'medium'), 12.81744399469395)	(('al', 'qaeda'), 12.619944825657706)
(('intent', 'meaning'), 12.798828316526604)	(('al', 'qaida'), 12.619944825657704)
(('th', 'jefferson'), 12.774375272802065)	(('richard', 'nixon'), 12.602871312298765)
(('john', 'quincy'), 12.774375272802063)	(('saddam', 'hussein'), 12.57930284116036)
(('precious', 'metals'), 12.715481583748494)	(('harry', 'truman'), 12.539774476973722)
(('thomas', 'jefferson'), 12.686912431551724)	(('supreme', 'court'), 12.525226168404565)
(('lake', 'erie'), 12.633019423556524)	(('carbon', 'pollution'), 12.469119923409936)
(('almighty', 'god'), 12.604450271359752)	(('baby', 'boom'), 12.463825623740426)

Assignment 1

Prof. Lu Xiao

Name: Raghav Raheja

Course: IST664

((('john', 'tyler'), 12.604450271359752))	((('persian', 'gulf'), 12.437169991739372))
((('san', 'jacinto'), 12.576435895190155))	((('capitol', 'introduced'), 12.396711427881888))
((('san', 'juan'), 12.576435895190155))	((('sides', 'aisle'), 12.356910419823912))

```
Python 3.6
File Edit Search Source Run Debug Windows Projects Task View Help
Editor: C:\Users\rader\Desktop\Raghav\Syracuse University\Semester 2\IST 664 Natural Language Processing\Assignment 1
File explorer
Python console

# coding: utf-8
import os
import nltk
from nltk.corpus import PlaintextCorpusReader
from nltk import word_tokenize
from nltk.collocations import *

os.getcwd()
stateunion1 = PlaintextCorpusReader('.', 'state_un_')
stateunion1.load = stateunion1.raw('state_un_')
stateunion1.tokenize = nltk.word_tokenize
stateunion1.tokenize('')
stateunion1.tokenize('')
stateunion1_lower = [w.lower() for w in stateunion1.tokenize('')]
stateunion1_lower[:10]
stateunion1_chars = [w for w in stateunion1_lower if w.isalpha()]
stateunion1_chars[:10]
stopwords = nltk.corpus.stopwords.words('eng')
stateunion1_stopwords = [w for w in stateunion1_lower if w not in stopwords]
stateunion1_stopwords[:10]
stateunion1_freqdist = FreqDist(stateunion1_stopwords)
for val in stateunion1_freqdist.keys():
    stateunion1_freqdist[val] = stateunion1_freqdist[val]**0.5
stateunion1_keys = list(stateunion1_freqdist.keys())
stateunion1_keys[:10]
topkeys_stateunion1 = stateunion1_freqdist.most_common(10)
for pair in topkeys_stateunion1:
    print(pair)

bigram_stateunion1 = nltk.collocations.BigramCollocationFinder.from_words(stateunion1_lower)
bigram_stateunion1.scores()

[('states', 0.011430774311094691),
 ('government', 0.009319742741513676),
 ('united', 0.0078252543702837),
 ('may', 0.006557404657748054),
 ('congress', 0.0062071234738970785),
 ('upon', 0.00610021007097771555),
 ('would', 0.0057955107839331),
 ('public', 0.005772363184497322),
 ('country', 0.004882389733585735),
 ('great', 0.0046045423506591),
 ('made', 0.004454165337273931),
 ('state', 0.004359960210217964),
 ('last', 0.0043244519096742254),
 ('war', 0.0035012800515423755),
 ('present', 0.0034888421840594189),
 ('time', 0.003392450511326423),
 ('people', 0.003280692780374469),
 ('year', 0.003295404618056471),
 ('power', 0.003123573243182387),
 ('citizens', 0.003093212514460592),
 ('subject', 0.002964836520674615),
 ('shall', 0.002913469127302648),
 ('without', 0.002783328575047085),
 ('union', 0.0026993669251067474),
 ('act', 0.0026321976121307785),
 ('several', 0.0026196833651827045),
 ('one', 0.0026081120393167923)]
```

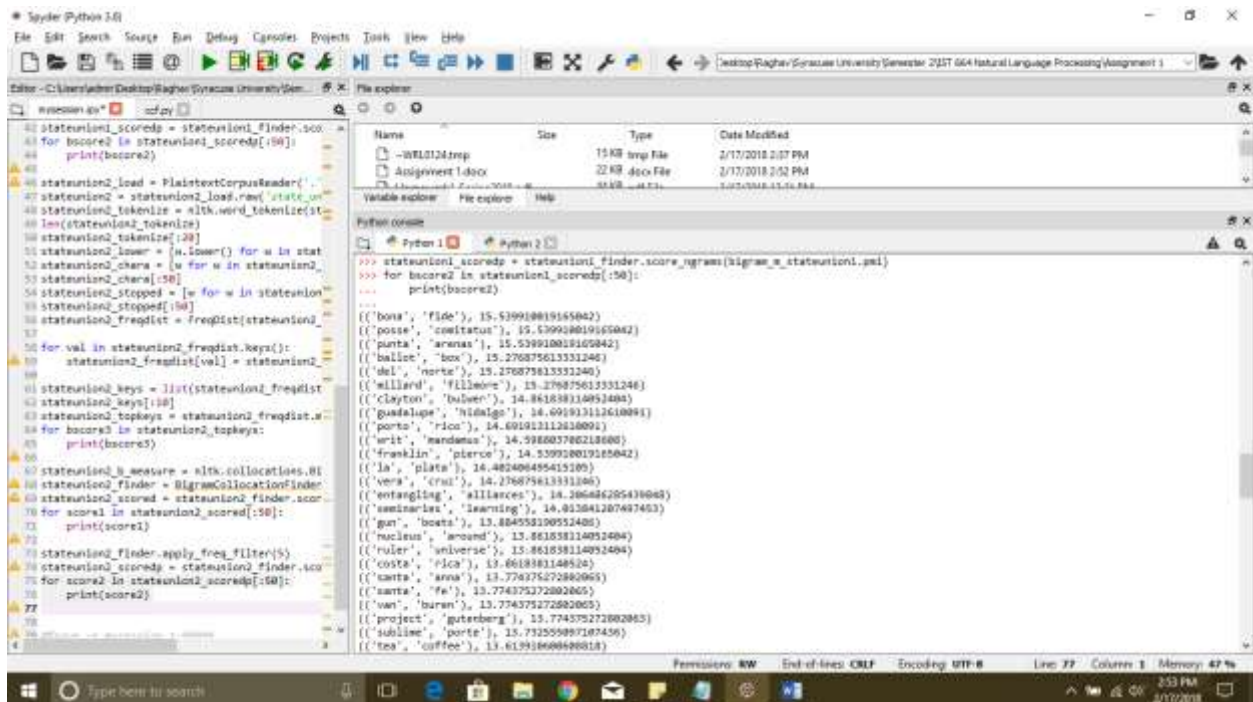
```
Python 3.6
In [18]: bigram_stateunion1.scores()[10:]
[('united', 'states'), 0.007648905979748451],
[('great', 'britain'), 0.0011502741543834663],
[('last', 'session'), 0.0010159359284715786],
[('public', 'debt'), 0.0007514567345636513],
[('state', 'union'), 0.0007262624469766311],
[('house', 'representatives'), 0.0006521101827671117],
[('fiscal', 'year'), 0.000445230735071905],
[('union', 'address'), 0.000404523853507195],
[('report', 'secretary'), 0.000389334419237203],
[('public', 'lands'), 0.000457587019797460],
[('two', 'countries'), 0.0005121608415517623],
[('present', 'year'), 0.0004440967254957835],
[('within', 'limits'), 0.00041080623159080523],
[('secretary', 'treasury'), 0.00041561014928380717],
[('fellow', 'citizens'), 0.00040721398465181106],
[('session', 'congress'), 0.00040721398465181106],
[('act', 'congress'), 0.00039463197377038169],
[('general', 'government'), 0.00039042165538781884],
[('year', 'ending'), 0.00039042165538781884],
[('british', 'government'), 0.0003862235730718288],
[('two', 'governments'), 0.0003736293201238266],
[('citizens', 'united'), 0.0003618358791758325],
[('federal', 'government'), 0.0003588189685981445],
[('secretary', 'war'), 0.0003525389145438364],
[('annual', 'message'), 0.0003400446675958422],
[('public', 'service'), 0.0003358458527984417],
[('senate', 'house'), 0.0003358458527984417],
[('consideration', 'congress'), 0.00032325233833185],
[('ending', 'june'), 0.0003148561736998539],
[('last', 'annual'), 0.0003148561736998539],
[('attention', 'congress'), 0.0003106588913838584],
[('government', 'united'), 0.0003064640090678578],
[('public', 'money'), 0.0002896676790838856],
[('indian', 'tribes'), 0.00027707343385187145],
[('mexican', 'government'), 0.0002728753505398734],
[('poet', 'united'), 0.0002728753505398734],
[('treasury', 'notes'), 0.0002728753505398734],
[('upon', 'subject'), 0.00026867726822387534],
[('commercial', 'intercourse'), 0.000264479185987773],
[('several', 'states'), 0.000264479185987773]]
```


Assignment 1

Prof. Lu Xiao

Name: Raghav Raheja

Course: IST664



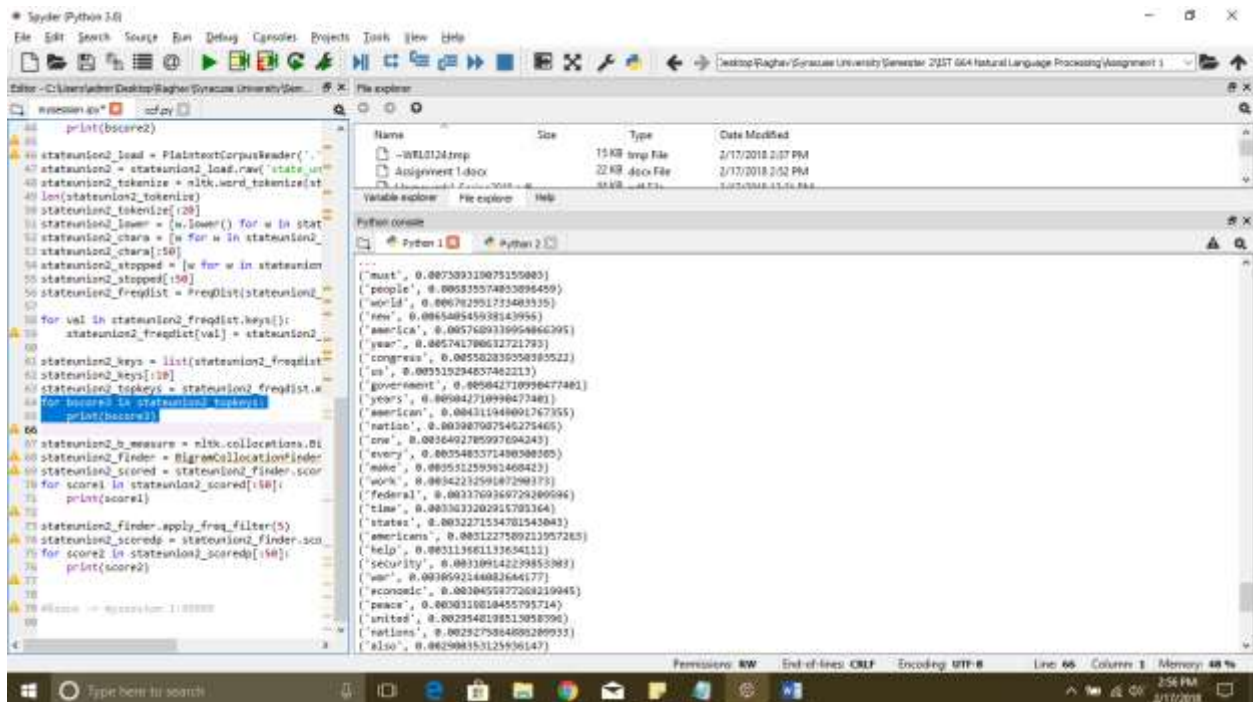
```
stateunion1_scored = stateunion1_finder.score
for bscore2 in stateunion1_scored[150]:
    print(bscore2)

stateunion2_load = PlaintextCorpusReader('...')
stateunion2 = stateunion2_load.raw('state_un...')
stateunion2_tokenize = nltk.word_tokenize(st...
len(stateunion2_tokenize)
stateunion2_tokenize[10]
stateunion2_lower = [w.lower() for w in stat...
stateunion2_charr = [w for w in stateunion2...
stateunion2_charr[150]
stateunion2_stopped = [w for w in stateunion...
stateunion2_freqdist = FreqDist(stateunion2...
for val in stateunion2_freqdist.keys():
    stateunion2_freqdist[val] = stateunion2...
stateunion2_keys = list(stateunion2_freqdist...
stateunion2_topkeys = stateunion2_freqdist.m...
for bscore3 in stateunion2_topkeys:
    print(bscore3)

stateunion2_h_measure = nltk.collocations.BI...
stateunion2_finder = BigramCollocationFinder...
stateunion2_scored = stateunion2_finder.sco...
for score1 in stateunion2_scored[150]:
    print(score1)

stateunion2_finder.apply_freq_filter(5)
stateunion2_scored = stateunion2_finder.sco...
for score2 in stateunion2_scored[150]:
    print(score2)
```

```
>>> stateunion1_scored = stateunion1_finder.score_ngrams(bigram_n_stateunion1.ps1)
>>> for bscore2 in stateunion1_scored[150]:
...     print(bscore2)
...
(('bona', 'fide'), 15.5399188191545842)
(('posse', 'comitatus'), 15.5399188191545842)
(('punta', 'armas'), 15.5399188191545842)
(('baliet', 'box'), 15.276875613331246)
(('del', 'norte'), 15.276875613331246)
(('willard', 'fillmore'), 15.276875613331246)
(('clayton', 'dubee'), 14.86188114851485)
(('guadalupe', 'hidalg'), 14.691913112610891)
(('ports', 'rice'), 14.691913112610891)
(('writ', 'mandamus'), 14.59688788218888)
(('franklin', 'pierce'), 14.5399188191545842)
(('la', 'plate'), 14.48240649545128)
(('vera', 'cru'), 14.276875613331246)
(('entangling', 'alliances'), 14.26648265438043)
(('seminaries', 'learning'), 14.913841287487453)
(('gun', 'boats'), 13.84553199514851)
(('nucleus', 'around'), 13.86188114851485)
(('ruler', 'universe'), 13.86188114851485)
(('costa', 'rica'), 13.86188114851485)
(('santa', 'anna'), 13.774375272880865)
(('santa', 'fe'), 13.774375272880865)
(('van', 'buren'), 13.774375272880865)
(('project', 'gutenberg'), 13.774375272880865)
(('sublime', 'porte'), 13.71255897187458)
(('tea', 'coffee'), 13.613918688888888)
```



```
print(bscore2)

stateunion2_load = PlaintextCorpusReader('...')
stateunion2 = stateunion2_load.raw('state_un...')
stateunion2_tokenize = nltk.word_tokenize(st...
len(stateunion2_tokenize)
stateunion2_tokenize[10]
stateunion2_lower = [w.lower() for w in stat...
stateunion2_charr = [w for w in stateunion2...
stateunion2_charr[150]
stateunion2_stopped = [w for w in stateunion...
stateunion2_freqdist = FreqDist(stateunion2...
for val in stateunion2_freqdist.keys():
    stateunion2_freqdist[val] = stateunion2...
stateunion2_keys = list(stateunion2_freqdist...
stateunion2_topkeys = stateunion2_freqdist.m...
for bscore3 in stateunion2_topkeys:
    print(bscore3)

stateunion2_h_measure = nltk.collocations.BI...
stateunion2_finder = BigramCollocationFinder...
stateunion2_scored = stateunion2_finder.sco...
for score1 in stateunion2_scored[150]:
    print(score1)

stateunion2_finder.apply_freq_filter(5)
stateunion2_scored = stateunion2_finder.sco...
for score2 in stateunion2_scored[150]:
    print(score2)
```

```
>>> stateunion1_scored = stateunion1_finder.score_ngrams(bigram_n_stateunion1.ps1)
>>> for bscore2 in stateunion1_scored[150]:
...     print(bscore2)
...
('must', 0.007309319075150885)
('people', 0.005335574853094559)
('world', 0.00678295173483335)
('new', 0.00548545938143955)
('america', 0.0057689319956866395)
('year', 0.00576178651271792)
('congress', 0.0055235505855522)
('us', 0.00515234457462213)
('government', 0.005642718998477481)
('years', 0.005847718998477481)
('american', 0.004311488891767355)
('nation', 0.003987807546229445)
('ow', 0.0036492789977694243)
('every', 0.00348537148058655)
('make', 0.00351259392468423)
('work', 0.0034221259187298173)
('federal', 0.003376936472848094)
('time', 0.0033632202925783164)
('states', 0.003227155478154843)
('americans', 0.003127589211957285)
('help', 0.00311368113858411)
('security', 0.00310914223985383)
('war', 0.003059214488264477)
('economic', 0.0030455877264219945)
('peace', 0.0030312818455795714)
('united', 0.0029548119851305896)
('nations', 0.002927584489289933)
('also', 0.002988153125596147)
```


Prof. Lu Xiao

Name: Raghav Raheja

Course: IST664

```
Python Cadres (Documents)
In [10]: for score1 in stateunion_scored['score1']:
...:     print(score1)

('united', 'states'), 0.00206096089267311766)
('state', 'union'), 0.001111884639475667)
('american', 'people'), 0.0010547956136130583)
('last', 'year'), 0.001011211100011742)
('fiscal', 'year'), 0.0009442342432302399)
('federal', 'government'), 0.000831564556686245)
('social', 'security'), 0.000826870608187000)
('health', 'care'), 0.000807471052983779)
('let', 'us'), 0.0007938453054121626)
('years', 'ago'), 0.0007153087924008541)
('union', 'address'), 0.000621673817614683)
('united', 'nations'), 0.0006127506606040451)
('billion', 'dollars'), 0.0005900561915850064)
('million', 'dollars'), 0.0005704395101621032)
('soviet', 'union'), 0.0005673617226009677)
('men', 'women'), 0.0005120349973112748)
('free', 'world'), 0.0004947304221800438)
('every', 'american'), 0.0004649350048299066)
('members', 'congress'), 0.0004319498917673545)
('economic', 'growth'), 0.0004205560153950277)
('middle', 'east'), 0.0004180923140935045)
('make', 'sure'), 0.0003594226527110813)
('new', 'nations'), 0.0003580571138068)
('first', 'time'), 0.000367010042454280)
('four', 'years'), 0.0003676503962454280)
('state', 'local'), 0.0003631115046501934)
('ask', 'congress'), 0.0003587260068381156)
('armed', 'forces'), 0.0003406170315005806)
('world', 'war'), 0.000340417033505806)
('must', 'continue'), 0.0003358781307797729)
('next', 'years'), 0.000335878130777729)
('work', 'together'), 0.000335878130777729)
('foreign', 'policy'), 0.0003177225646565419)
('new', 'jobs'), 0.0003177225646565419)
('two', 'years'), 0.000290106021141100)
('vice', 'president'), 0.00028410880331411867)
('around', 'world'), 0.00029048910197160545)
('national', 'security'), 0.000283956300190877)
('must', 'also'), 0.00028141143441007945)
('address', 'january'), 0.0002723336268404445)
('human', 'rights'), 0.0002677947330676517)
```

The screenshot shows a Jupyter Notebook environment with a Python script for text analysis. The script is as follows:

```

print(bscore2)

stateunion1_load = PlaintextCorpusReader('.', ".txt")
stateunion1 = stateunion1_load.raw('state_union_part2.txt')
stateunion1_tokenizer = nltk.word_tokenizer(stateunion1)
len(stateunion1_tokenizer)
stateunion1_tokenizer[20]
stateunion1_lower = [w.lower() for w in stateunion1_tokenizer]
stateunion1_chars = [w for w in stateunion1_lower if w.isalpha()]
stateunion1_chars[50]
stateunion1_stopped = [w for w in stateunion1_chars if w not in stopwords]
stateunion1_stopped[50]
stateunion1_freqdist = FreqDist(stateunion1_stopped)
len(stateunion1_freqdist.keys())
stateunion1_freqdist[val] = stateunion1_freqdist[val]/len(stateunion1_stopped)
stateunion1_keys = list(stateunion1_freqdist.keys())
stateunion1_keys[10]
stateunion1_topkeys = stateunion1_freqdist.most_common(50)
for bscore1 in stateunion1_topkeys:
    print(bscore1)

stateunion1_h_measure = nltk.collocations.BigramAssocMeasures()
stateunion1_finder = BigramCollocationFinder.from_words(stateunion1_stopped)
stateunion1_scored = stateunion1_finder.score_ngrams(stateunion1_h_measure.raw_freq)
for score1 in stateunion1_scored[10]:
    print(score1)

stateunion1_finder.apply_freq_filter(5)
stateunion1_scored = stateunion1_finder.score_ngrams(stateunion1_h_measure.pmi)
for score2 in stateunion1_scored[10]:
    print(score2)

# Save as Assignment 1.ipynb

```

The file explorer window on the right shows the following files:

Name	Size	Type	Date Modified
-WORLD124.txt	15 KB	Text File	2/17/2019 2:37 PM
Assignment 1.docx	22 KB	Word File	2/17/2019 2:52 PM

The Python console on the right shows the output of the script, displaying a list of words and their associated scores:

```

...
(('el', 'salvador'), 15.164205541681517)
(('ladies', 'gentlemen'), 15.164205541681517)
(('bin', 'laden'), 14.9418728054907)
(('saudi', 'arabia'), 14.9418728054907)
(('sam', 'rayburn'), 14.749227642682672)
(('jimmy', 'carter'), 14.42728974771531)
(('endowed', 'creator'), 14.316268415520567)
(('northern', 'ireland'), 14.184205541681517)
(('gerald', 'ford'), 14.097315114002297)
(('finor', 'appens'), 14.01226208416688)
(('iron', 'curtain'), 13.9418728054907)
(('grass', 'roots'), 13.901228936047725)
(('thomas', 'jefferson'), 13.785753710627788)
(('son', 'daughters'), 13.749227642682672)
(('red', 'tape'), 13.749227642682672)
(('jill', 'biden'), 13.678838514711237)
(('lyndon', 'johnson'), 13.661765081162234)
(('barack', 'obama'), 13.461765081162234)
(('teen', 'pregnancy'), 13.57938204116036)
(('abraham', 'lincoln'), 13.4018399991082)
(('mo', 'dad'), 13.45644605374828)
(('empowerment', 'zevec'), 13.356918419823812)
(('ronald', 'clinton'), 13.327766034164396)
(('willard', 'waggon'), 13.289796223945373)
(('synthetic', 'humid'), 13.275290634276264)
(('grey', 'barley'), 13.204987526178064)
(('elementary', 'secondary'), 13.12278878505555)
(('laterconferential', 'ballistic'), 13.093273465299122)

```