# PresidentSpeeches

August 31, 2016

In [2]:
```
# import required libraries to scrape presidential transcripts
from bs4 import BeautifulSoup
import pandas as pd
import pickle
import urllib2
import re
```

In [35]:
```
import urllib2,sys,os
from bs4 import BeautifulSoup,NavigableString
from string import punctuation as p
from multiprocessing import Pool
import re, nltk
import requests
reload(sys)


sys.setdefaultencoding('utf8')

#==============================================================================
# Scraping and cleaning one speech from Obama to show the method works
#==============================================================================

obama_4427_url = 'http://www.millercenter.org/president/obama/speeches/speech-4427'
obama_4427 = urllib2.urlopen(obama_4427_url).read()
obama_4427 = BeautifulSoup(obama_4427)

# find the speech itself within the HTML
obama_4427 = obama_4427.find('div',{'id': 'transcript'},{'class': 'displaytext'})

# obama_4427_div.text removes extraneous characters (e.g. '<br/>')
obama_4427 = obama_4427.text.lower()

# for further text analysis, remove punctuation
punctuation = re.compile('[{}]+'.format(re.escape(p)))

# obama_4427_nopunct = [line.decode('utf-8').strip() for line in obama_4427_html.readlines()]

obama_4427 = punctuation.sub('', obama_4427)
obama_4427 = obama_4427.replace('|',' ')
obama_4427 = obama_4427.replace('transcript','')

# divide obama_4427_str_processed into individual words
words = obama_4427.split(' ')
```

```
#===============================================================================
# Cleaning links begins below, so that  we can process all 911 speeches through processURL()
#===============================================================================

url = 'http://www.millercenter.org/president/speeches'
url2 = 'http://www.millercenter.org'

conn = urllib2.urlopen(url)
html = conn.read()

miller_center_soup = BeautifulSoup(html)
links = miller_center_soup.find_all('a')

linklist = [tag.get('href') for tag in links if tag.get('href') is not None]

# remove all items in list that don't contain 'speeches'
linkslist = [_ for _ in linklist if re.search('speeches',_)]
del linkslist[0:2]

# concatenate 'http://www.millercenter.org' with end of speech links
every_link_dups = [url2 + end_link for end_link in linkslist]

# remove duplicates
seen = set()
every_link = [] # no duplicates array
for l in every_link_dups:
    if l not in seen:
        every_link.append(l)
        seen.add(l)

# list of presidents (print(len(set(presidents))) = 43 total)
presidents_dups = [l[l.find('president/')+len('president/'):] for l in every_link if 'presiden
presidents_dups = [l[0:l.find('/')] for l in presidents_dups]
set2 = set()
presidents = []
for l in presidents_dups:
    if l not in set2:
        presidents.append(l)
        seen.add(l)

presidents = sorted(presidents)

# the following two lines - now commented out - were used to identify duplicates in the origin
# import collections
# print [l for l, count in collections.Counter(every_link).items() if count > 1]

# define a function to clean & store speeches from 'every_link' repository

def processURL(l):
        open_url = urllib2.urlopen(l).read()
        x=urllib2.urlopen(obama_4427_url).read()
        item_soup = BeautifulSoup(x)
        item_div = item_soup.find('div',{'id':'transcript'},{'class':'displaytext'})
        item_str = item_div.text.lower()
```

```
                item_str_processed = punctuation.sub('',item_str)
                item_str_processed_final = item_str_processed.replace('|',' ')

                splitlink = l.split("/")
                president = splitlink[4]
                speech_num = splitlink[-1].split("-")[1]
                filename = "{0}_{1}".format(president, speech_num)

                return filename, item_str_processed_final # returning a tuple

        # right now, this loop only works for 423 speeches - where are the remaining ones?
        for l in every_link[1:423]:
            filename, content = processURL(l) # tuple unpacking
            with open(filename, 'w') as f:
                f.write(content)
```

In [3]:
```
import os
from os import path
root = "F:/topicmodel_speeches"
files = os.listdir(root)
```

In [4]:
```
#load speeches into a list
docs = list()
for file in files:
    with open(path.join(root, file), 'r') as fd:
       txt = fd.read()
       docs.append(txt)
```

In [5]:
```
import re
def clean(doc):
    doc = re.sub(r'[^\w\s]*', '', doc)
    doc = re.sub(r'[\s]+', ' ', doc)
    doc = doc.lower().strip()
    return doc
```

In [6]:
```
clean_docs = list()
for doc in docs:
    doc = clean(doc)
    clean_docs.append(doc)
```

In [5]:
```
#tokenize speeches
token_docs = list()
for doc in clean_docs:
    token_docs.append(doc.split())
```

In [6]:
```
#remove stopwords
stopwords = list()
with open('F:/2060724/topicmodel_stopwords.txt', 'r') as fd:
    for line in fd.readlines():
        stopwords.append(line.strip())
```

In [7]:
```
sw_token_docs = list()
for doc in token_docs:
    sw_doc = list()
    for token in doc:
        if not token in stopwords:
```

```
                sw_doc.append(token)
        sw_token_docs.append(sw_doc)
```

In [8]:
```
# perform topic modelling
from gensim import corpora, models, similarities
dictionary = corpora.Dictionary(sw_token_docs)
corpus = [dictionary.doc2bow(doc) for doc in sw_token_docs]
```

In [7]:
```
from __future__ import division

import graphlab as gl
import pandas as pd
import pyLDAvis
import pyLDAvis.graphlab

pyLDAvis.enable_notebook()
```

```
C:\Users\pankaj\Anaconda2\lib\site-packages\IPython\core\formatters.py:98: DeprecationWarning: DisplayFo
  def _formatters_default(self):
C:\Users\pankaj\Anaconda2\lib\site-packages\IPython\core\formatters.py:677: DeprecationWarning: PlainTex
  def _deferred_printers_default(self):
C:\Users\pankaj\Anaconda2\lib\site-packages\IPython\core\formatters.py:669: DeprecationWarning: PlainTex
  def _singleton_printers_default(self):
C:\Users\pankaj\Anaconda2\lib\site-packages\IPython\core\formatters.py:672: DeprecationWarning: PlainTex
  def _type_printers_default(self):
```

In [8]:
```
doc=gl.SFrame(clean_docs)
sf_paragraphs = doc.rename({'X1': 'paragraph'})
doc.head()
```

```
[INFO] graphlab.cython.cy_server: GraphLab Create v2.1 started. Logging: C:\Users\pankaj\AppData\Local\T

This non-commercial license of GraphLab Create for academic use is assigned to pankajvshrma@gmail.com ar

C:\Users\pankaj\Anaconda2\lib\site-packages\IPython\core\formatters.py:92: DeprecationWarning: DisplayFo
  def _ipython_display_formatter_default(self):
C:\Users\pankaj\Anaconda2\lib\site-packages\IPython\core\formatters.py:669: DeprecationWarning: PlainTex
  def _singleton_printers_default(self):
C:\Users\pankaj\Anaconda2\lib\site-packages\IPython\core\formatters.py:672: DeprecationWarning: PlainTex
  def _type_printers_default(self):
C:\Users\pankaj\Anaconda2\lib\site-packages\IPython\core\formatters.py:677: DeprecationWarning: PlainTex
  def _deferred_printers_default(self):
```

Out[8]:
```
Columns:
        paragraph       str

Rows: 10

Data:
+--------------------------------+
|           paragraph            |
+--------------------------------+
| fellow citizens of the sen... |
| whereas it is the duty of ... |
| fellow citizens of the sen... |
| fellow citizens of the sen... |
```

```
| i the president of the uni... |
| i meet you upon the presen... |
| gentlemen of the house of ... |
| fellowcitizens of the sena... |
| whereas i have received au... |
| fellowcitizens i am again ... |
+--------------------------------+
[10 rows x 1 columns]
```

```
In [9]: re_words_split = re.compile("(\w+)")
        sf_paragraphs['paragraph_words_number'] = sf_paragraphs['paragraph'].apply(lambda p: len(re_word
        sf_paragraphs = sf_paragraphs[sf_paragraphs['paragraph_words_number'] >=25]
```

```
In [10]: docs =  gl.text_analytics.count_ngrams(sf_paragraphs['paragraph'], n=1)
```

```
In [11]: stopwords = gl.text_analytics.stopwords()
         # adding some additional stopwords to make the topic model more clear
         stopwords |= set(['man', 'mr', 'sir', 'make', 'made', 'll', 'door', 'long', 'day', 'small'])
         docs = docs.dict_trim_by_keys(stopwords, exclude=True)
         docs = docs.dropna()
```

```
In [12]: topic_model = gl.topic_model.create(docs, num_topics=10)
```

```
Learning a topic model


       Number of documents       622


           Vocabulary size     36392


   Running collapsed Gibbs sampling


+------------+---------------+----------------+-----------------+


| Iteration | Elapsed Time  | Tokens/Second  | Est. Perplexity |


+------------+---------------+----------------+-----------------+


| 10         | 952.857ms     | 5.83093e+006   | 0               |


+------------+---------------+----------------+-----------------+
```

```
In [13]: topic_model.get_topics().print_rows(100)
```

```
+--------+------------+-------------------+
| topic |     word    |       score       |
+--------+------------+-------------------+
|   0    |   people   | 0.0518808217563  |
|   0    |   peace    | 0.0280909494448  |
```

```
|   0    |  nations   | 0.0240855117597  |
|   0    |  america   | 0.0215886155404  |
|   0    |    time    | 0.0190136913143  |
|   1    |    law     | 0.0212298873226  |
|   1    |   state    | 0.0205198576463  |
|   1    |    part    | 0.0158104771399  |
|   1    |  question  | 0.0157235347306  |
|   1    |   power    | 0.0139557057405  |
|   2    | government | 0.0582714339429  |
|   2    |  congress  | 0.0272801740345  |
|   2    |  country   | 0.0252658860182  |
|   2    |   great    |  0.022455474643  |
|   2    |  national  | 0.0166140393956  |
|   3    |   states   | 0.0565825654513  |
|   3    |   united   | 0.0503400597837  |
|   3    |   world    | 0.0381020882464  |
|   3    |    war     | 0.0305768759347  |
|   3    |   great    | 0.0113647808662  |
|   4    |    time    |  0.020797494843  |
|   4    |    meet    | 0.0151005711818  |
|   4    |  citizens  | 0.0131829792304  |
|   4    |    plan    | 0.0129223356642  |
|   4    |   party    | 0.0120659353753  |
|   5    |    year    | 0.0253373614164  |
|   5    | president  | 0.0210878095595  |
|   5    |  vietnam   | 0.00948435615977 |
|   5    |  federal   | 0.00917202829502 |
|   5    |  security  | 0.00914363485277 |
|   6    |   years    | 0.0408231758269  |
|   6    |    work    | 0.0224122673637  |
|   6    |   future   | 0.0163026950611  |
|   6    | president  |  0.016165709135  |
|   6    |  progress  |  0.014494480837  |
|   7    |  american  | 0.0311206720485  |
|   7    |    good    | 0.0252488471337  |
|   7    |   rights   | 0.0114762600909  |
|   7    |   health   | 0.00935326618581 |
|   7    |    past    | 0.00909796945039 |
|   8    |   today    | 0.0241819637855  |
|   8    | americans  | 0.0204738860966  |
|   8    |    men     | 0.0186510075688  |
|   8    |    hope    |  0.015644037006  |
|   8    |    life    | 0.0118113180502  |
|   9    |   order    | 0.00801987346995 |
|   9    |   effect   | 0.00778655912923 |
|   9    | attention  | 0.00773744032065 |
|   9    |  treasury  | 0.00722169283062 |
|   9    |    men     | 0.00689014087275 |
+--------+------------+------------------+
[50 rows x 3 columns]
```

In [17]:
```python
import pyLDAvis
import pyLDAvis.graphlab
pyLDAvis.enable_notebook()
```

```
pyLDAvis.graphlab.prepare(topic_model, docs)
```

Out[17]:

```
PreparedData(topic_coordinates=            Freq  cluster  topics         x         y
topic
5      13.003340        1       1  0.132520 -0.236946
0      12.873568        1       2  0.136203 -0.214329
3      12.633409        1       3 -0.249032 -0.088049
2      11.868215        1       4 -0.238315 -0.055087
6      10.834902        1       5 -0.110124  0.071484
7       8.598419        1       6  0.016187  0.065130
4       8.090620        1       7  0.057047  0.153710
9       7.965658        1       8  0.109521  0.158670
1       7.544927        1       9  0.027125  0.120734
8       6.586941        1      10  0.118869  0.024682, topic_info=      Category         Freq
term
20294  Default  5914.000000  government  5914.000000  30.0000  30.0000
25665  Default  5613.000000      states  5613.000000  29.0000  29.0000
9471   Default  6089.000000      people  6089.000000  28.0000  28.0000
17743  Default  5031.000000      united  5031.000000  27.0000  27.0000
10151  Default  3461.000000    congress  3461.000000  26.0000  26.0000
22802  Default  3239.000000   president  3239.000000  25.0000  25.0000
996    Default  3912.000000       world  3912.000000  24.0000  24.0000
27835  Default  3275.000000        time  3275.000000  23.0000  23.0000
3583   Default  3461.000000    american  3461.000000  22.0000  22.0000
8394   Default  3159.000000       peace  3159.000000  21.0000  21.0000
27527  Default  2647.000000     nations  2647.000000  20.0000  20.0000
20907  Default  3040.000000         war  3040.000000  19.0000  19.0000
3065   Default  2399.000000     america  2399.000000  18.0000  18.0000
18513  Default  3535.000000     country  3535.000000  17.0000  17.0000
710    Default  2895.000000       years  2895.000000  16.0000  16.0000
15690  Default  1885.000000    national  1885.000000  15.0000  15.0000
4564   Default  1851.000000         men  1851.000000  14.0000  14.0000
18886  Default  2440.000000      nation  2440.000000  13.0000  13.0000
34361  Default  2132.000000      public  2132.000000  12.0000  12.0000
29025  Default  3378.000000       great  3378.000000  11.0000  11.0000
29621  Default  1888.000000        work  1888.000000  10.0000  10.0000
31546  Default  1815.000000       state  1815.000000   9.0000   9.0000
28328  Default  1845.000000       power  1845.000000   8.0000   8.0000
20350  Default  2547.000000        year  2547.000000   7.0000   7.0000
34242  Default  1566.000000        free  1566.000000   6.0000   6.0000
22598  Default  1563.000000     freedom  1563.000000   5.0000   5.0000
9496   Default  1546.000000    citizens  1546.000000   4.0000   4.0000
32565  Default  1525.000000   americans  1525.000000   3.0000   3.0000
13727  Default  1252.000000      rights  1252.000000   2.0000   2.0000
6249   Default  1576.000000       today  1576.000000   1.0000   1.0000
...        ...          ...         ...          ...      ...      ...
11889  Topic10   247.166930  revolution   262.430748   2.6602  -5.3745
6465   Topic10   133.060869    concerns   139.710721   2.6713  -5.9938
33157  Topic10   241.508778    benefits   257.720889   2.6551  -5.3977
4046   Topic10   715.850503        meet   790.762126   2.6206  -4.3111
15690  Topic10  1655.103698    national  1885.743368   2.5896  -3.4729
4564   Topic10  1612.667560         men  1851.105520   2.5822  -3.4989
6237   Topic10   325.438030      entire   355.139931   2.6327  -5.0994
4394   Topic10   158.522552    payments   169.135260   2.6553  -5.8187
```

| | | | | | | |
|---|---|---|---|---|---|---|
| 24129 | Topic10 | 613.060745 | common | 752.771266 | 2.5148 | -4.4661 |
| 16706 | Topic10 | 414.082408 | millions | 504.123079 | 2.5233 | -4.8585 |
| 33457 | Topic10 | 239.622728 | hopes | 277.841500 | 2.5721 | -5.4055 |
| 21636 | Topic10 | 683.787643 | history | 925.041694 | 2.4179 | -4.3569 |
| 17876 | Topic10 | 369.760219 | individual | 470.064683 | 2.4801 | -4.9717 |
| 12168 | Topic10 | 104.770110 | proportion | 112.284762 | 2.6508 | -6.2328 |
| 217 | Topic10 | 619.661922 | service | 878.346339 | 2.3712 | -4.4554 |
| 10866 | Topic10 | 452.746445 | high | 681.120506 | 2.3117 | -4.7692 |
| 35389 | Topic10 | 267.913487 | return | 364.687393 | 2.4117 | -5.2939 |
| 1177 | Topic10 | 457.461571 | full | 751.871988 | 2.2232 | -4.7589 |
| 4663 | Topic10 | 217.933146 | school | 286.267257 | 2.4473 | -5.5004 |
| 29659 | Topic10 | 687.559744 | law | 1533.877279 | 1.9177 | -4.3514 |
| 6053 | Topic10 | 237.736677 | task | 333.871545 | 2.3805 | -5.4134 |
| 3843 | Topic10 | 255.654158 | success | 395.679561 | 2.2833 | -5.3407 |
| 2652 | Topic10 | 256.597183 | hand | 415.540601 | 2.2380 | -5.3371 |
| 1188 | Topic10 | 263.198360 | experience | 449.411921 | 2.1850 | -5.3116 |
| 22656 | Topic10 | 262.255335 | friends | 461.523022 | 2.1549 | -5.3152 |
| 10279 | Topic10 | 261.312310 | resources | 518.732364 | 2.0344 | -5.3188 |
| 11052 | Topic10 | 345.241561 | hope | 1393.939569 | 1.3244 | -5.0403 |
| 10529 | Topic10 | 231.135500 | community | 495.942121 | 1.9566 | -5.4416 |
| 8394 | Topic10 | 255.654158 | peace | 3159.748604 | 0.2057 | -5.3407 |
| 9385 | Topic10 | 238.679703 | life | 1226.344820 | 1.0834 | -5.4094 |

[634 rows x 6 columns], token_table=

| term | Topic | Freq | Term |
|---|---|---|---|
| 23942 | 2 | 0.997020 | 20 |
| 27531 | 6 | 0.987603 | acquisition |
| 10099 | 1 | 0.324132 | act |
| 10099 | 2 | 0.209732 | act |
| 10099 | 4 | 0.458331 | act |
| 10099 | 10 | 0.007333 | act |
| 25384 | 3 | 0.017692 | action |
| 25384 | 4 | 0.423684 | action |
| 25384 | 5 | 0.557773 | action |
| 31839 | 2 | 0.995190 | active |
| 5765 | 1 | 0.050815 | acts |
| 5765 | 6 | 0.780159 | acts |
| 5765 | 8 | 0.158423 | acts |
| 5765 | 10 | 0.008967 | acts |
| 24883 | 2 | 0.014217 | actual |
| 24883 | 6 | 0.980940 | actual |
| 33900 | 1 | 0.833128 | administration |
| 33900 | 6 | 0.163566 | administration |
| 33900 | 10 | 0.002353 | administration |
| 255 | 2 | 0.096015 | affairs |
| 255 | 3 | 0.184829 | affairs |
| 255 | 6 | 0.715311 | affairs |
| 32386 | 9 | 0.991450 | afghanistan |
| 13186 | 9 | 0.993444 | agenda |
| 32456 | 6 | 0.983734 | agent |
| 28943 | 1 | 0.766633 | ago |
| 28943 | 2 | 0.232985 | ago |
| 27894 | 1 | 0.985668 | agreement |
| 27894 | 3 | 0.014081 | agreement |

```
20778      2  0.223032            aid
...      ...       ...            ...
34984      6  0.019768           ways
34984      7  0.006589           ways
27537      6  0.924991         wealth
27537     10  0.069251         wealth
28130      1  0.997665        weapons
911        2  0.993561           west
911        3  0.002799           west
17739      9  0.998958           weve
22278      6  0.988308         whilst
34877      9  0.998695          white
32884     10  0.971625         whites
33092      2  0.984671         wishes
12027      9  0.995889           wont
29621      3  0.066710           work
29621      5  0.932879           work
26605      7  0.998387        workers
649        3  0.001764        working
649        5  0.014112        working
649        7  0.001764        working
649        9  0.980809        working
996        3  0.129824          world
996        5  0.212114          world
996        7  0.657808          world
20350      1  0.651352           year
20350      2  0.348252           year
710        4  0.146088          years
710        5  0.248660          years
710        9  0.605072          years
17353      9  0.991178          youre
23715      9  0.995523          youve

[1171 rows x 3 columns], R=30, lambda_step=0.01, plot_opts={'xlab': 'PC1', 'ylab': 'PC2'}, topi
```

In [49]:
```python
import graphlab as gl
import urllib2
import gensim
import nltk
import re

txt = urllib2.urlopen("https://drive.google.com/open?id=0B5g0WZIzJLuEU3hqczNEb21MaGM").read()

re_words_split = re.compile("(\w+)")
tokenizer = nltk.data.load('tokenizers/punkt/english.pickle')
def txt2words(s):
    s = re.sub("[^a-zA-Z]", " ", s).lower()
    return re_words_split.findall(s)

class MySentences(object):
        def __init__(self, txt):
            self._txt = txt.decode("utf8")

        def __iter__(self):
```

```
            """
            Split the English text into sentences and then to words using NLTK
            :param txt: input text.
            :param remove_none_english_chars: if True then remove none English chars from text
            :return: list of words in which each list consists of single sentence's words from
            :rtype: str
            """
            # split text into sentences using NLTK package
            for s in tokenizer.tokenize(self._txt):
                yield txt2words(s)

sentences = MySentences(txt)
model = gensim.models.Word2Vec(sentences, size=100, window=5, min_count=3, workers=4)
```

```
In [55]: print model.most_similar("obama")
```

```
[(u'com', 0.7005459666252136), (u'google', 0.685105562210083), (u'a', 0.6578261256217957), (u'true', 0.
```

```
In [60]: import numpy as np
         def txt2avg_vector(txt, w2v_model):
             words = [w for w in txt2words(txt.lower()) if w in w2v_model]
             v = np.mean([w2v_model[w] for w in words],axis=0)
             return v

         sf_paragraphs['mean_vector'] = sf_paragraphs['paragraph'].apply(lambda p: txt2avg_vector(p, mo
```

```
In [61]: #construncting nearest neighbors model
         nn_model = gl.nearest_neighbors.create(sf_paragraphs, features=['mean_vector'])

         #calaculating the two nearest neighbors of each paragraph from all the paragraphs
         r = nn_model.query(sf_paragraphs, k=2)
         r.head(10)
```

```
Starting ball tree nearest neighbors model training.


+------------+--------------+


| Tree level | Elapsed Time |


+------------+--------------+


| 0          | 17.634ms     |


+------------+--------------+


+--------------+------------+--------------+


| Query points | % Complete. | Elapsed Time |
```

```
+--------------+-------------+--------------+
| 1            | 0           | 5.003ms      |
| Done         |             | 638.43ms     |
+--------------+-------------+--------------+
```

Out[61]:
```
Columns:
        query_label        int
        reference_label        int
        distance        float
        rank        int

Rows: 10

Data:
+-------------+-----------------+-------------------+------+
| query_label | reference_label |      distance     | rank |
+-------------+-----------------+-------------------+------+
|      0      |        0        |        0.0        |  1   |
|      0      |       247       | 0.000894301737532 |  2   |
|      1      |        1        |        0.0        |  1   |
|      1      |        75       |  0.00177082049266 |  2   |
|      2      |        2        |        0.0        |  1   |
|      2      |        16       | 0.000928484636116 |  2   |
|      3      |        3        |        0.0        |  1   |
|      3      |       110       |  0.00074604515377 |  2   |
|      4      |        4        |        0.0        |  1   |
|      4      |       139       |  0.00162629810938 |  2   |
+-------------+-----------------+-------------------+------+
[10 rows x 4 columns]
```

In [62]:
```
#filter out paragraphs that are exactly exactly the same
r = r[r['distance'] != 0]

#filter out paragraphs that are with distance >= 0.1
r = r[r['distance'] < 0.08]
r
```

Out[62]:
```
Columns:
        query_label        int
        reference_label        int
        distance        float
        rank        int

Rows: Unknown

Data:
+-------------+-----------------+-------------------+------+
| query_label | reference_label |      distance     | rank |
```

```
+-------------+----------------+------------------+------+
|      0      |      247       | 0.000894301737532 |  2   |
|      1      |       75       |  0.00177082049266 |  2   |
|      2      |       16       | 0.000928484636116 |  2   |
|      3      |      110       |  0.00074604515377 |  2   |
|      4      |      139       |  0.00162629810938 |  2   |
|      5      |       11       | 0.000744305314392 |  2   |
|      6      |       93       |  0.00553016265452 |  2   |
|      7      |      103       | 0.000770922708859 |  2   |
|      8      |      214       |  0.00120041979798 |  2   |
|      9      |      371       |  0.00371562996025 |  2   |
+-------------+----------------+------------------+------+
[? rows x 4 columns]
Note: Only the head of the SFrame is printed. This SFrame is lazily evaluated.
You can use sf.materialize() to force materialization.
```

In [63]:
```
sf_paragraphs = sf_paragraphs.add_row_number('query_label')
sf_paragraphs = sf_paragraphs.add_row_number('reference_label')
sf_similar = r.join(sf_paragraphs, on="query_label").join(sf_paragraphs, on="reference_label")
```

In [65]:
```
sf_similar[['paragraph', 'paragraph.1', 'distance']]
```

Out[65]:
```
Columns:
        paragraph        str
        paragraph.1        str
        distance        float

Rows: 622

Data:
+--------------------------------+--------------------------------+
|           paragraph            |          paragraph.1           |
+--------------------------------+--------------------------------+
| fellow citizens of the sen... | senator wagner governor le... |
| whereas it is the duty of ... | to the house of representa... |
| fellow citizens of the sen... | i trust i do not deceive m... |
| fellow citizens of the sen... | fellowcitizens of the sena... |
| i the president of the uni... | by the president of the un... |
| i meet you upon the presen... | fellow citizens of the sen... |
| gentlemen of the house of ... | whereas by an act of the c... |
| fellowcitizens of the sena... | fellow citizens of the sen... |
| whereas i have received au... | gentlemen of the congress ... |
| fellowcitizens i am again ... | dr newhouse chancellor tol... |
+--------------------------------+--------------------------------+
+-------------------+
|     distance      |
+-------------------+
| 0.000894301737532 |
|  0.00177082049266 |
| 0.000928484636116 |
|  0.00074604515377 |
|  0.00162629810938 |
| 0.000744305314392 |
|  0.00553016265452 |
| 0.000770922708859 |
```

```
|  0.00120041979798 |
|  0.00371562996025 |
+--------------------+
[622 rows x 3 columns]
Note: Only the head of the SFrame is printed.
You can use print_rows(num_rows=m, num_columns=n) to print more rows and columns.
```

In [66]:
```
print sf_similar[1]['paragraph']
print "-"*100
print sf_similar[1]['paragraph.1']
```

```
whereas it is the duty of all nations to acknowledge the providence of almighty god to obey his will to
----------------------------------------------------------------------------------------------------
to the house of representatives of the united states having considered the bill this day presented to me
```

In [ ]:

In [12]:
```
import logging, gensim
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)
lda = gensim.models.ldamodel.LdaModel(corpus, id2word=dictionary, num_topics=25, update_every=
```

In [13]:
```
#print 25 topics out to 20 words
t=0
for i in lda.show_topics(num_topics=25, num_words=20, log=False, formatted=True):
    print "Topic # ", t , i
    t = t + 1
```

```
Topic #  0 (0, u'0.007*people + 0.006*kosovo + 0.006*war + 0.006*bosnia + 0.005*army + 0.005*peace + 0.0
Topic #  1 (1, u'0.007*increase + 0.007*people + 0.007*states + 0.006*government + 0.006*year + 0.006*wo
Topic #  2 (2, u'0.018*business + 0.011*law + 0.010*labor + 0.009*government + 0.009*great + 0.008*publi
Topic #  3 (3, u'0.010*america + 0.009*people + 0.007*iraq + 0.006*american + 0.006*world + 0.005*nation
Topic #  4 (4, u'0.012*people + 0.008*years + 0.008*america + 0.007*american + 0.006*government + 0.006
Topic #  5 (5, u'0.016*government + 0.011*states + 0.009*united + 0.007*congress + 0.006*country + 0.005
Topic #  6 (6, u'0.009*black + 0.003*blacks + 0.003*soldiers + 0.002*negro + 0.002*cities + 0.002*santia
Topic #  7 (7, u'0.013*tariff + 0.010*schedule + 0.010*country + 0.009*articles + 0.007*rates + 0.007*co
Topic #  8 (8, u'0.009*watergate + 0.008*statute + 0.005*house + 0.005*made + 0.004*case + 0.004*court +
Topic #  9 (9, u'0.016*states + 0.012*united + 0.008*government + 0.006*great + 0.006*congress + 0.005*s
Topic #  10 (10, u'0.009*world + 0.008*people + 0.007*war + 0.007*president + 0.006*nations + 0.006*peac
Topic #  11 (11, u'0.011*war + 0.009*people + 0.008*government + 0.008*peace + 0.006*world + 0.005*germa
Topic #  12 (12, u'0.011*nuclear + 0.009*treaty + 0.008*cooperation + 0.007*gorbachev + 0.007*united + 0
Topic #  13 (13, u'0.012*vietnam + 0.008*energy + 0.007*congress + 0.007*president + 0.006*south + 0.005
Topic #  14 (14, u'0.021*president + 0.012*people + 0.009*senator + 0.008*united + 0.007*states + 0.006
Topic #  15 (15, u'0.024*president + 0.010*lebanon + 0.007*israel + 0.007*government + 0.006*middle + 0
Topic #  16 (16, u'0.014*berlin + 0.010*rights + 0.010*victims + 0.006*city + 0.005*law + 0.005*wall + 0
Topic #  17 (17, u'0.007*law + 0.006*judge + 0.006*crime + 0.005*justice + 0.005*negro + 0.005*race + 0
Topic #  18 (18, u'0.012*people + 0.009*nation + 0.009*men + 0.009*great + 0.008*freedom + 0.008*world +
Topic #  19 (19, u'0.023*united + 0.017*states + 0.014*nations + 0.011*world + 0.008*peace + 0.007*ameri
Topic #  20 (20, u'0.013*government + 0.012*people + 0.006*constitution + 0.006*power + 0.006*states + 0
Topic #  21 (21, u'0.010*people + 0.007*action + 0.007*affirmative + 0.006*women + 0.005*opportunity + 0
Topic #  22 (22, u'0.007*ireland + 0.007*northern + 0.004*people + 0.003*belfast + 0.003*peace + 0.003*b
Topic #  23 (23, u'0.014*united + 0.014*peace + 0.011*world + 0.011*soviet + 0.010*states + 0.009*vietna
Topic #  24 (24, u'0.014*care + 0.014*health + 0.009*insurance + 0.008*people + 0.008*system + 0.006*gov
```

In [14]:
```
#print topic weight in each speech
count=1
for doc in sw_token_docs:
```

```python
        vec = dictionary.doc2bow(doc)
        print "Speech # ", count, lda[vec]
        #print first 100 words of speech to verify correct speech
        print doc[0:100]
        count = count + 1
```

```
Speech #  1 [(9, 0.50640713197693921), (18, 0.1351239923731434), (20, 0.35489200038374513)]
['fellow', 'citizens', 'senate', 'house', 'representatives', 'vicissitudes', 'incident', 'life', 'event
Speech #  2 [(9, 0.48031336104297279), (18, 0.038903812756833481), (20, 0.47563662736909162)]
['duty', 'nations', 'acknowledge', 'providence', 'almighty', 'god', 'obey', 'grateful', 'benefits', 'hum
Speech #  3 [(9, 0.85788618980384024), (18, 0.016940799056699755), (20, 0.12264427550532239)]
['fellow', 'citizens', 'senate', 'house', 'representatives', 'embrace', 'great', 'satisfaction', 'oppor
Speech #  4 [(9, 0.96433459673170208), (20, 0.033974226797508569)]
['fellow', 'citizens', 'senate', 'house', 'representatives', 'meeting', 'feel', 'satisfaction', 'repeat
Speech #  5 [(2, 0.019201746428718 23), (9, 0.88722024552655465), (18, 0.091920757573697401)]
['president', 'united', 'states', 'mouth', 'written', 'speech', 'signed', 'hand', 'sealed', 'seal', 'spe
Speech #  6 [(9, 0.78537066381864251), (20, 0.21360483729484889)]
['meet', 'present', 'occasion', 'feelings', 'naturally', 'inspired', 'strong', 'impression', 'prosperous
Speech #  7 [(6, 0.27893439469922354), (9, 0.70598363808641584)]
['gentlemen', 'house', 'representatives', 'maturely', 'considered', 'act', 'passed', 'houses', 'intitle
Speech #  8 [(5, 0.1033456691486738), (9, 0.73556906389931564), (20, 0.16015108436167524)]
['fellowcitizens', 'senate', 'house', 'representatives', 'abatement', 'satisfaction', 'meet', 'present'
Speech #  9 [(9, 0.98829268292546646)]
['received', 'authentic', 'information', 'lawless', 'wicked', 'persons', 'western', 'frontier', 'state'
Speech #  10 [(9, 0.25859419927253813), (20, 0.60129730499773759), (21, 0.12381219943145229)]
['fellowcitizens', 'called', 'voice', 'country', 'execute', 'functions', 'chief', 'magistrate', 'occasio
Speech #  11 [(9, 0.99030303030214084)]
['appears', 'state', 'war', 'exists', 'austria', 'prussia', 'sardinia', 'great', 'britain', 'united', 'n
Speech #  12 [(9, 0.88265064894725132), (20, 0.11615764120806112)]
['fellow', 'citizens', 'senate', 'house', 'representatives', 'commencement', 'term', 'called', 'office'
Speech #  13 [(9, 0.72357720365982547), (11, 0.011938779270439888), (19, 0.010541064522359263), (20, 0.2
['combinations', 'defeat', 'execution', 'laws', 'laying', 'duties', 'spirits', 'distilled', 'united', 's
Speech #  14 [(7, 0.018133486774113052), (9, 0.44330317229990956), (20, 0.53530408166642396)]
['hope', 'combinations', 'constitution', 'laws', 'united', 'states', 'western', 'counties', 'pennsylvani
Speech #  15 [(9, 0.59240198311730441), (20, 0.40681370315712201)]
['fellow', 'citizens', 'senate', 'house', 'representatives', 'call', 'mind', 'gracious', 'indulgence',
Speech #  16 [(9, 0.6702256451312204), (19, 0.32429816439210213)]
['commissioners', 'appointed', 'president', 'united', 'states', 'confer', 'citizens', 'western', 'countr
Speech #  17 [(5, 0.041331081800033434), (9, 0.93454316645196434), (20, 0.023064231844384046)]
['trust', 'deceive', 'indulge', 'persuasion', 'met', 'period', 'present', 'situation', 'public', 'affai
Speech #  18 [(5, 0.12960732717010376), (9, 0.84264425898816653), (15, 0.025602072378076039)]
['gentlemen', 'house', 'representatives', 'utmost', 'attention', 'considered', 'resolution', '24th', 'in
Speech #  19 [(9, 0.436783365099843), (18, 0.11973421854646764), (23, 0.44194931530817533)]
['beloved', 'cherokees', 'years', 'passed', 'white', 'people', 'america', 'long', 'space', 'time', 'goo
Speech #  20 [(5, 0.0446205804212585), (9, 0.46322136792961477), (20, 0.49176147661528574)]
['period', 'election', 'citizen', 'administer', 'executive', 'government', 'united', 'states', 'distant
Speech #  21 [(9, 0.99919597989940534)]
['fellow', 'citizens', 'senate', 'house', 'representatives', 'recurring', 'internal', 'situation', 'cou
Speech #  22 [(9, 0.35106488356826915), (18, 0.12482873671106175), (20, 0.52316419556639571)]
['perceived', 'early', 'times', 'middle', 'america', 'remained', 'unlimited', 'submission', 'foreign',
Speech #  23 [(9, 0.79778067186820123), (20, 0.20148332813171665)]
['personal', 'inconveniences', 'members', 'senate', 'house', 'representatives', 'leaving', 'families',
Speech #  24 [(9, 0.99443260414873924)]
['time', 'apprehensive', 'account', 'contagious', 'sickness', 'afflicted', 'city', 'philadelphia', 'con
```

```
Speech #  25 [(9, 0.40104233439592768), (19, 0.11996710957901698), (20, 0.47589196447541182)]
['safety', 'prosperity', 'nations', 'ultimately', 'essentially', 'depend', 'protection', 'blessing', 'a
Speech #  26 [(9, 0.99894505494495411)]
['gentlemen', 'senate', 'gentlemen', 'house', 'representatives', 'reverence', 'resignation', 'contempla
Speech #  27 [(5, 0.027370856210638651), (9, 0.96811538629562466)]
['peculiar', 'satisfaction', 'meet', '6th', 'congress', 'united', 'states', 'america', 'coming', 'parts
Speech #  28 [(9, 0.23453777754934166), (18, 0.11579102538112021), (20, 0.1733092603627969), (23, 0.474
['gentlemen', 'senate', 'gentlemen', 'house', 'representatives', 'letter', 'herewith', 'transmitted', '
Speech #  29 [(19, 0.99244094488118917)]
['late', 'wicked', 'treasonable', 'insurrection', 'authority', 'united', 'states', 'sundry', 'persons',
Speech #  30 [(5, 0.12387825239535569), (9, 0.83718340782554423), (20, 0.037389044004312436)]
['gentlemen', 'senate', 'gentlemen', 'house', 'representatives', 'immediately', 'adjournment', 'congres
Speech #  31 [(5, 0.059212222414988573), (9, 0.16329735374043625), (18, 0.14270853820030646), (20, 0.63
['friends', 'fellowcitizens', 'called', 'undertake', 'duties', 'executive', 'office', 'country', 'avail
Speech #  32 [(3, 0.23087777237722912), (5, 0.13633792959815047), (9, 0.30263448899650519), (20, 0.3282
['gentleman', 'received', 'remonstrance', 'pleased', 'address', 'appointment', 'samuel', 'bishop', 'off
Speech #  33 [(9, 0.80179540059271404), (20, 0.19653934487244712)]
['fellow', 'citizens', 'senate', 'house', 'representatives', 'circumstance', 'sincere', 'gratification'
Speech #  34 [(1, 0.012259525161309346), (9, 0.34292872090760096), (18, 0.17105285946269452), (20, 0.46
['gentleman', 'affectionate', 'sentiments', 'esteem', 'approbation', 'good', 'express', 'behalf', 'danb
Speech #  35
```

```
---------------------------------------------------------------------------

ValueError                                Traceback (most recent call last)

<ipython-input-14-d5357dc7b030> in <module>()
      3 for doc in sw_token_docs:
      4     vec = dictionary.doc2bow(doc)
----> 5     print "Speech # ", count, lda[vec]
      6     #print first 100 words of speech to verify correct speech
      7     print doc[0:100]


C:\Users\pankaj\Anaconda2\lib\site-packages\ipykernel\iostream.pyc in write(self, string)
    315
    316             is_child = (not self._is_master_process())
--> 317             self._buffer.write(string)
    318             if is_child:
    319                 # newlines imply flush in subprocesses


ValueError: I/O operation on closed file
```

```
In [15]: lda.print_topics(20)
```

```
Out[15]: [(3,
           u'0.010*america + 0.009*people + 0.007*iraq + 0.006*american + 0.006*world + 0.005*nation + 
          (24,
           u'0.014*care + 0.014*health + 0.009*insurance + 0.008*people + 0.008*system + 0.006*governme
          (13,
           u'0.012*vietnam + 0.008*energy + 0.007*congress + 0.007*president + 0.006*south + 0.005*amer
          (11,
```

```
  u’0.011*war + 0.009*people + 0.008*government + 0.008*peace + 0.006*world + 0.005*germany +
 (1,
  u’0.007*increase + 0.007*people + 0.007*states + 0.006*government + 0.006*year + 0.006*work
 (19,
  u’0.023*united + 0.017*states + 0.014*nations + 0.011*world + 0.008*peace + 0.007*american +
 (15,
  u’0.024*president + 0.010*lebanon + 0.007*israel + 0.007*government + 0.006*middle + 0.005*p
 (21,
  u’0.010*people + 0.007*action + 0.007*affirmative + 0.006*women + 0.005*opportunity + 0.005*
 (8,
  u’0.009*watergate + 0.008*statute + 0.005*house + 0.005*made + 0.004*case + 0.004*court + 0.
 (4,
  u’0.012*people + 0.008*years + 0.008*america + 0.007*american + 0.006*government + 0.006*con
 (17,
  u’0.007*law + 0.006*judge + 0.006*crime + 0.005*justice + 0.005*negro + 0.005*race + 0.004*m
 (22,
  u’0.007*ireland + 0.007*northern + 0.004*people + 0.003*belfast + 0.003*peace + 0.003*lands
 (23,
  u’0.014*united + 0.014*peace + 0.011*world + 0.011*soviet + 0.010*states + 0.009*vietnam + 0
 (0,
  u’0.007*people + 0.006*kosovo + 0.006*war + 0.006*bosnia + 0.005*army + 0.005*peace + 0.005*
 (14,
  u’0.021*president + 0.012*people + 0.009*senator + 0.008*united + 0.007*states + 0.006*kenne
 (7,
  u’0.013*tariff + 0.010*schedule + 0.010*country + 0.009*articles + 0.007*rates + 0.007*consu
 (18,
  u’0.012*people + 0.009*nation + 0.009*men + 0.009*great + 0.008*freedom + 0.008*world + 0.00
 (5,
  u’0.016*government + 0.011*states + 0.009*united + 0.007*congress + 0.006*country + 0.005*ye
 (12,
  u’0.011*nuclear + 0.009*treaty + 0.008*cooperation + 0.007*gorbachev + 0.007*united + 0.007*
 (20,
  u’0.013*government + 0.012*people + 0.006*constitution + 0.006*power + 0.006*states + 0.005*
```

```
In [ ]: !
```