

mcpp_taller9_Daniela_Gaita?n_C

October 23, 2016

1 Taller 9

Métodos Computacionales para Políticas Públicas - UROSARIO

Entrega: viernes 28-oct-2016 11:59 PM

[Daniela Gaitán Cotrino] [daniela.gaitanc@urosario.edu.co]

1.1 Instrucciones:

- Guarde una copia de este *Jupyter Notebook* en su computador, idealmente en una carpeta destinada al material del curso.
- Modifique el nombre del archivo del *notebook*, agregando al final un guión inferior y su nombre y apellido, separados estos últimos por otro guión inferior. Por ejemplo, mi *notebook* se llamaría: mcpp_taller9_santiago_mataallana
- Marque el *notebook* con su nombre y e-mail en el bloque verde arriba. Reemplace el texto “[Su nombre acá]” con su nombre y apellido. Similar para su e-mail.
- Desarrolle la totalidad del taller sobre este *notebook*, insertando las celdas que sea necesario debajo de cada pregunta. Haga buen uso de las celdas para código y de las celdas tipo *markdown* según el caso.
- Recuerde salvar periódicamente sus avances.
- Cuando termine el taller:
 1. Descárguelo en PDF. Si tiene algún problema con la conversión, descárguelo en HTML.
 2. Suba todos los archivos a su repositorio en GitHub, en una carpeta destinada exclusivamente para este taller, antes de la fecha y hora límites.

NLTK Book (<http://www.nltk.org/book/>), Exercises: - Chapter 1: 22, 26, 28 - Chapter 2: 2, 4, 11

```
In [5]: import nltk
        from nltk.book import *
        import matplotlib.pyplot as plt
        %matplotlib inline
```

```
*** 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
```

1.2 Language Processing and Python

2. Find all the four-letter words in the Chat Corpus (text5). With the help of a frequency distribution (FreqDist), show these words in decreasing order of frequency.

```
In [6]: punto2 = sorted([w.lower() for w in text5 if len(w) is 4 if w.isalpha()])

In [7]: fdist2 = FreqDist(punto2)
        fdist2.most_common()

Out[7]: [('part', 1022),
         ('join', 1021),
         ('that', 284),
         ('what', 201),
         ('here', 185),
         ('have', 171),
         ('like', 160),
         ('with', 154),
         ('chat', 146),
         ('your', 142),
         ('good', 132),
         ('lmao', 128),
         ('just', 128),
         ('know', 104),
         ('room', 103),
         ('this', 98),
         ('from', 96),
         ('well', 91),
         ('hiya', 85),
         ('yeah', 85),
         ('they', 84),
         ('back', 79),
         ('dont', 77),
         ('want', 71),
         ('love', 63),
         ('guys', 59),
         ('some', 59),
```

('been', 58),
('talk', 58),
('when', 54),
('nice', 54),
('time', 52),
('haha', 46),
('girl', 45),
('make', 44),
('need', 44),
('mode', 42),
('song', 42),
('will', 42),
('then', 41),
('much', 40),
('over', 40),
('were', 39),
('does', 38),
('work', 38),
('take', 38),
('even', 37),
('seen', 36),
('come', 36),
('damn', 36),
('only', 36),
('more', 35),
('nick', 33),
('long', 30),
('tell', 29),
('hell', 29),
('last', 29),
('name', 28),
('sure', 28),
('away', 28),
('them', 28),
('look', 27),
('down', 26),
('call', 26),
('baby', 26),
('cool', 26),
('sexy', 25),
('play', 25),
('many', 23),
('same', 23),
('stop', 23),
('live', 23),
('hate', 23),
('said', 23),
('life', 22),

('ever', 22),
('hear', 21),
('very', 20),
('give', 19),
('hugs', 19),
('feel', 19),
('must', 19),
('mean', 19),
('find', 18),
('cant', 18),
('shit', 17),
('hair', 17),
('lets', 17),
('nite', 17),
('left', 17),
('busy', 17),
('fine', 17),
('lost', 17),
('eyes', 16),
('heya', 16),
('game', 16),
('real', 16),
('fuck', 15),
('sits', 15),
('kill', 15),
('read', 14),
('goes', 14),
('nope', 14),
('true', 14),
('wait', 14),
('shut', 14),
('keep', 14),
('else', 14),
('awww', 13),
('male', 13),
('free', 13),
('near', 13),
('pick', 13),
('bout', 12),
('hehe', 12),
('told', 12),
('gets', 12),
('home', 12),
('hope', 12),
('kids', 12),
('yall', 12),
('cold', 12),
('than', 12),

('head', 12),
('stay', 12),
('face', 12),
('used', 12),
('show', 11),
('perv', 11),
('babe', 11),
('help', 11),
('doin', 11),
('year', 11),
('into', 11),
('days', 11),
('hard', 11),
('dang', 11),
('wont', 11),
('rock', 10),
('care', 10),
('mind', 10),
('week', 10),
('kiss', 10),
('once', 10),
('liam', 10),
('type', 10),
('mine', 10),
('hey', 10),
('full', 9),
('such', 9),
('book', 9),
('neck', 9),
('dead', 9),
('runs', 9),
('aint', 9),
('sick', 9),
('hour', 9),
('ahhh', 9),
('best', 9),
('pics', 9),
('hmmm', 9),
('crap', 9),
('soon', 9),
('okay', 9),
('says', 8),
('case', 8),
('rule', 8),
('word', 8),
('oops', 8),
('suck', 8),
('wana', 8),

('poor', 8),
('lady', 8),
('wife', 8),
('sang', 8),
('dude', 8),
('made', 8),
('hows', 8),
('kick', 8),
('hand', 8),
('went', 8),
('blue', 8),
('kool', 7),
('alot', 7),
('none', 7),
('fast', 7),
('wear', 7),
('took', 7),
('food', 7),
('dear', 7),
('ummm', 6),
('whos', 6),
('pink', 6),
('came', 6),
('woot', 6),
('cali', 6),
('yoko', 6),
('caps', 6),
('ohhh', 6),
('gone', 6),
('done', 6),
('seem', 6),
('high', 6),
('send', 6),
('next', 6),
('goin', 6),
('list', 6),
('comp', 6),
('late', 6),
('ball', 6),
('thru', 6),
('knew', 6),
('most', 6),
('sock', 6),
('ride', 6),
('sing', 6),
('main', 6),
('blah', 6),
('holy', 5),

('fire', 5),
('wish', 5),
('kent', 5),
('meds', 5),
('also', 5),
('till', 5),
('nose', 5),
('miss', 5),
('lose', 5),
('lime', 5),
('lord', 5),
('felt', 5),
('boys', 5),
('lick', 5),
('huge', 5),
('legs', 5),
('boss', 5),
('idea', 5),
('land', 5),
('quit', 5),
('turn', 5),
('xbox', 5),
('evil', 5),
('beer', 5),
('heck', 5),
('meet', 5),
('soul', 5),
('cute', 5),
('hang', 5),
('warm', 5),
('wall', 5),
('luck', 5),
('fall', 5),
('fool', 5),
('roll', 5),
('joke', 5),
('easy', 5),
('feet', 5),
('both', 5),
('puff', 4),
('ohio', 4),
('elle', 4),
('mary', 4),
('wind', 4),
('drew', 4),
('ways', 4),
('road', 4),
('lies', 4),

('hott', 4),
('open', 4),
('hail', 4),
('self', 4),
('shot', 4),
('john', 4),
('rest', 4),
('sigh', 4),
('hawt', 4),
('mmmm', 4),
('ouch', 4),
('fart', 4),
('beat', 4),
('grrr', 4),
('each', 4),
('pain', 4),
('glad', 4),
('date', 4),
('team', 4),
('shes', 4),
('lame', 4),
('door', 4),
('tisk', 4),
('line', 4),
('rofl', 4),
('hook', 4),
('ugly', 4),
('jerk', 4),
('ones', 4),
('pfft', 4),
('swim', 4),
('pass', 4),
('town', 3),
('guyz', 3),
('kewl', 3),
('rain', 3),
('died', 3),
('born', 3),
('isnt', 3),
('bare', 3),
('hump', 3),
('hurt', 3),
('half', 3),
('city', 3),
('hiii', 3),
('orgy', 3),
('hola', 3),
('drop', 3),

('slow', 3),
('toes', 3),
('hick', 3),
('ring', 3),
('phil', 3),
('tune', 3),
('yawn', 3),
('gosh', 3),
('clap', 3),
('note', 3),
('ahem', 3),
('whoa', 3),
('akdt', 3),
('lead', 3),
('deop', 3),
('slap', 3),
('soft', 3),
('gawd', 3),
('wash', 3),
('wine', 3),
('wack', 3),
('band', 3),
('hold', 3),
('hank', 3),
('wazz', 3),
('move', 3),
('deal', 3),
('skin', 3),
('vote', 3),
('ding', 3),
('yada', 3),
('ello', 3),
('roof', 3),
('army', 3),
('jump', 3),
('rubs', 3),
('butt', 3),
('deep', 3),
('bend', 3),
('itch', 3),
('gold', 3),
('nana', 3),
('toss', 3),
('walk', 3),
('piff', 3),
('snow', 3),
('imma', 3),
('elev', 3),

('amen', 3),
('moon', 2),
('typo', 2),
('mike', 2),
('argh', 2),
('ewww', 2),
('cmon', 2),
('kold', 2),
('opps', 2),
('chip', 2),
('park', 2),
('rich', 2),
('tock', 2),
('golf', 2),
('deaf', 2),
('cars', 2),
('dumb', 2),
('ages', 2),
('luvs', 2),
('see', 2),
('cash', 2),
('eric', 2),
('sort', 2),
('heal', 2),
('gays', 2),
('whud', 2),
('hill', 2),
('rush', 2),
('rang', 2),
('hits', 2),
('doll', 2),
('fits', 2),
('sand', 2),
('hint', 2),
('cell', 2),
('limp', 2),
('aunt', 2),
('bite', 2),
('grrl', 2),
('root', 2),
('club', 2),
('spot', 2),
('eats', 2),
('area', 2),
('ltns', 2),
('sell', 2),
('ciao', 2),
('pour', 2),

('mass', 2),
('trip', 2),
('twin', 2),
('foot', 2),
('mono', 2),
('wats', 2),
('loud', 2),
('babi', 2),
('spin', 2),
('clue', 2),
('whip', 2),
('pmsl', 2),
('mama', 2),
('halo', 2),
('wooo', 2),
('haze', 2),
('side', 2),
('west', 2),
('flow', 2),
('tiff', 2),
('cast', 2),
('gimp', 2),
('tyvm', 2),
('meat', 2),
('porn', 2),
('corn', 2),
('hummm', 2),
('blew', 2),
('plan', 2),
('bone', 2),
('past', 2),
('newp', 2),
('shop', 2),
('adds', 2),
('hall', 2),
('king', 2),
('temp', 2),
('howz', 2),
('sooo', 2),
('uses', 2),
('hmp', 2),
('bear', 2),
('sore', 2),
('size', 2),
('ears', 2),
('yeas', 2),
('mins', 2),
('flaw', 2),

('pool', 2),
('five', 2),
('yard', 2),
('teck', 2),
('zone', 2),
('pies', 2),
('rent', 2),
('tick', 2),
('cost', 2),
('burp', 2),
('lawl', 2),
('slip', 2),
('kind', 2),
('jail', 1),
('vega', 1),
('febe', 1),
('wide', 1),
('pwns', 1),
('jush', 1),
('vbox', 1),
('duet', 1),
('thje', 1),
('chop', 1),
('werd', 1),
('yell', 1),
('toop', 1),
('raed', 1),
('hgey', 1),
('gear', 1),
('dood', 1),
('dojn', 1),
('yesh', 1),
('quiz', 1),
('lion', 1),
('tall', 1),
('goof', 1),
('inch', 1),
('woah', 1),
('jess', 1),
('site', 1),
('worl', 1),
('haaa', 1),
('sayn', 1),
('kmpH', 1),
('pair', 1),
('page', 1),
('post', 1),
('ebay', 1),

('caca', 1),
('asss', 1),
('nada', 1),
('benz', 1),
('orta', 1),
('mkay', 1),
('soup', 1),
('evah', 1),
('akon', 1),
('jeff', 1),
('beam', 1),
('tlak', 1),
('yeee', 1),
('caan', 1),
('thot', 1),
('cuss', 1),
('lake', 1),
('uyes', 1),
('spat', 1),
('mena', 1),
('ohwa', 1),
('nads', 1),
('jude', 1),
('whys', 1),
('dawg', 1),
('allo', 1),
('bowl', 1),
('ques', 1),
('bell', 1),
('mami', 1),
('sent', 1),
('syck', 1),
('tide', 1),
('fort', 1),
('dirt', 1),
('frst', 1),
('bomb', 1),
('bein', 1),
('ribs', 1),
('owww', 1),
('scuk', 1),
('brad', 1),
('peek', 1),
('wins', 1),
('blow', 1),
('mame', 1),
('anti', 1),
('rick', 1),

('dick', 1),
('cure', 1),
('poof', 1),
('star', 1),
('fish', 1),
('pork', 1),
('dies', 1),
('ussy', 1),
('ruth', 1),
('dark', 1),
('rats', 1),
('lust', 1),
('yess', 1),
('exit', 1),
('nooo', 1),
('bull', 1),
('clay', 1),
('sexs', 1),
('hots', 1),
('pasa', 1),
('paid', 1),
('plow', 1),
('nawp', 1),
('ther', 1),
('hiom', 1),
('ssid', 1),
('gret', 1),
('crib', 1),
('moms', 1),
('fake', 1),
('kina', 1),
('loss', 1),
('safe', 1),
('bust', 1),
('poop', 1),
('ntmn', 1),
('rape', 1),
('este', 1),
('thah', 1),
('tjhe', 1),
('barn', 1),
('grlz', 1),
('lung', 1),
('puts', 1),
('wild', 1),
('typr', 1),
('wore', 1),
('paul', 1),

('heat', 1),
('tooo', 1),
('whoo', 1),
('troy', 1),
('daft', 1),
('whew', 1),
('boot', 1),
('gift', 1),
('wher', 1),
('nude', 1),
('four', 1),
('tape', 1),
('wire', 1),
('okey', 1),
('hong', 1),
('ltnc', 1),
('surf', 1),
('tits', 1),
('bike', 1),
('wean', 1),
('boed', 1),
('pine', 1),
('aime', 1),
('outs', 1),
('poot', 1),
('rose', 1),
('outa', 1),
('keys', 1),
('coat', 1),
('tips', 1),
('bugs', 1),
('oooh', 1),
('text', 1),
('body', 1),
('drug', 1),
('shup', 1),
('hero', 1),
('heys', 1),
('guts', 1),
('scar', 1),
('cock', 1),
('nerd', 1),
('sean', 1),
('thnx', 1),
('sexi', 1),
('mahn', 1),
('howl', 1),
('soda', 1),

('arms', 1),
('grew', 1),
('knee', 1),
('smax', 1),
('card', 1),
('wubs', 1),
('waht', 1),
('firs', 1),
('maps', 1),
('reub', 1),
('lube', 1),
('peel', 1),
('seat', 1),
('hurr', 1),
('jack', 1),
('poll', 1),
('heee', 1),
('buff', 1),
('eeww', 1),
('dust', 1),
('ssri', 1),
('wyte', 1),
('feat', 1),
('base', 1),
('sink', 1),
('vamp', 1),
('news', 1),
('lyin', 1),
('pope', 1),
('giva', 1),
('dawn', 1),
('guns', 1),
('tere', 1),
('enuf', 1),
('form', 1),
('chik', 1),
('numb', 1),
('bacl', 1),
('yout', 1),
('test', 1),
('boom', 1),
('gags', 1),
('kept', 1),
('nawt', 1),
('menu', 1),
('wood', 1),
('dork', 1),
('bird', 1),

('cams', 1),
('offa', 1),
('waaa', 1),
('givs', 1),
('hazy', 1),
('mofo', 1),
('woof', 1),
('able', 1),
('bong', 1),
('otay', 1),
('idnt', 1),
('dotn', 1),
('urls', 1),
('joey', 1),
('puke', 1),
('addy', 1),
('lots', 1),
('brwn', 1),
('grea', 1),
('eggs', 1),
('eeek', 1),
('kong', 1),
('mark', 1),
('mang', 1),
('bied', 1),
('slam', 1),
('salt', 1),
('coem', 1),
('herd', 1),
('draw', 1),
('ruff', 1),
('pull', 1),
('ahah', 1),
('yoll', 1),
('tory', 1),
('tthe', 1),
('sori', 1),
('xmas', 1),
('meep', 1),
('bloe', 1),
('lapd', 1),
('gees', 1),
('lala', 1),
('acid', 1),
('prep', 1),
('cook', 1),
('push', 1),
('sign', 1),

('hide', 1),
('span', 1),
('mris', 1),
('save', 1),
('geez', 1),
('vent', 1),
('dint', 1),
('fear', 1),
('muah', 1),
('calm', 1),
('tail', 1),
('york', 1),
('lisa', 1),
('iowa', 1),
('anal', 1),
('ogan', 1),
('sets', 1),
('judy', 1),
('mauh', 1),
('mite', 1),
('fock', 1),
('grin', 1),
('cums', 1),
('pray', 1),
('bois', 1),
('ctrl', 1),
('ladz', 1),
('twit', 1),
('mess', 1),
('disc', 1),
('tart', 1),
('dyed', 1),
('wrek', 1),
('nuff', 1),
('serg', 1),
('fawk', 1),
('jeep', 1),
('lois', 1),
('crop', 1),
('bred', 1),
('poem', 1),
('scum', 1),
('gals', 1),
('icky', 1),
('docs', 1),
('term', 1),
('nods', 1),
('gray', 1),

('wuts', 1),
('tenn', 1),
('tend', 1),
('wrap', 1),
('abou', 1),
('ghet', 1),
('hooo', 1),
('fade', 1),
('pimp', 1),
('brat', 1),
('noth', 1),
('nova', 1),
('akst', 1),
('matt', 1),
('sips', 1),
('weed', 1),
('plus', 1),
('pure', 1),
('hogs', 1),
('pigs', 1),
('perk', 1),
('jane', 1),
('uhhh', 1),
('dman', 1),
('chit', 1),
('samn', 1),
('spit', 1),
('boyz', 1),
('tina', 1),
('prob', 1),
('ooer', 1),
('dogs', 1),
('east', 1),
('junk', 1),
('prof', 1),
('vvil', 1),
('toke', 1),
('asks', 1),
('choc', 1),
('dump', 1),
('gooo', 1),
('ally', 1),
('lool', 1),
('seth', 1),
('cepn', 1),
('lazy', 1),
('laid', 1),
('cyas', 1),

```
( 'http', 1),
( 'whou', 1),
( 'fair', 1),
( 'cops', 1),
( 'byes', 1),
( 'sext', 1)]
```

26. What does the following Python code do? `sum(len(w) for w in text1)` Can you use it to work out the average word length of a text

```
In [8]: (sum([len(w) for w in text1])) / len(text1)
```

```
Out[8]: 3.830411128023649
```

28. Define a function `percent(word, text)` that calculates how often a given word occurs in a text, and expresses the result as a percentage.

```
In [9]: def percent(word, text):
        return "{:.2%}".format(text.count(word) / len(text4))
```

```
In [10]: percent('part', text4)
```

```
Out[10]: '0.07%'
```

1.3 Accessing Text Corpora and Lexical Resources

2. Use the corpus module to explore `austen-persuasion.txt`. How many word tokens does this book have? How many word types?

```
In [11]: p2 = nltk.corpus.gutenberg.words('austen-persuasion.txt')
```

```
In [12]: len(p2)
```

```
Out[12]: 98171
```

```
In [13]: len(set(p2))
```

```
Out[13]: 6132
```

4. Read in the texts of the State of the Union addresses, using the `state_union` corpus reader. Count occurrences of `men`, `women`, and `people` in each document. What has happened to the usage of these words over time?

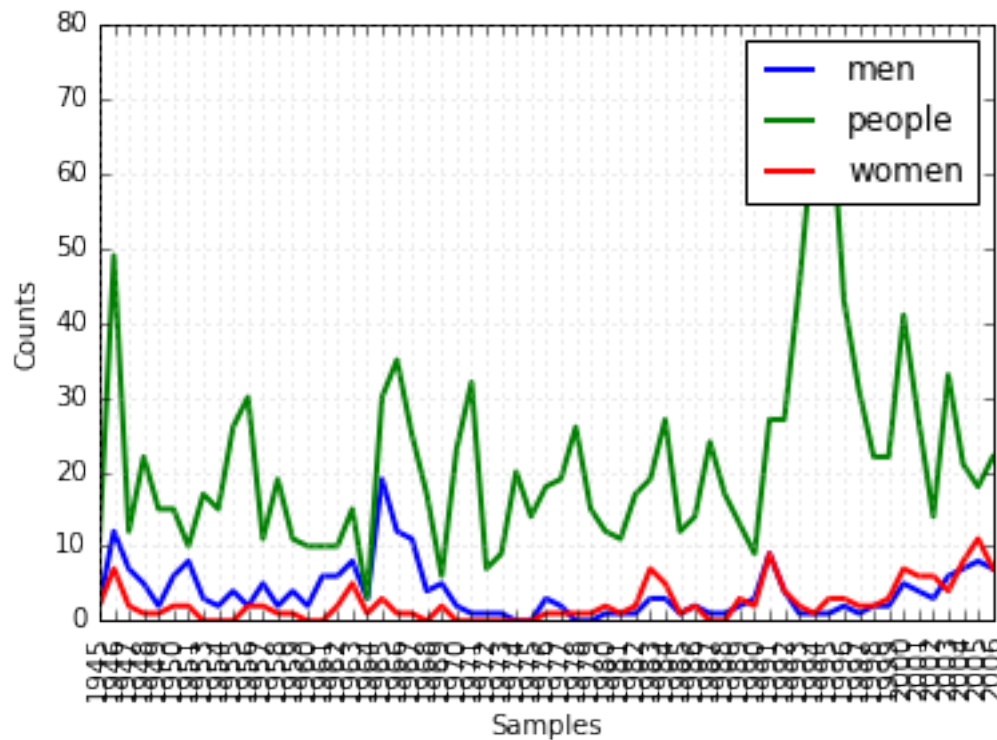
```
In [14]: from nltk.corpus import state_union
```

```
In [15]: cfd = nltk.ConditionalFreqDist((target, fileid[:4])
        for fileid in state_union.fileids()
        for w in state_union.words(fileid)
        for target in ['men', 'women', 'people']
        if w.lower() == target)

        cfd.tabulate()
```

	1945	1946	1947	1948	1949	1950	1951	1953	1954	1955	1956	1957	1958	1959	1960
men	2	12	7	5	2	6	8	3	2	4	2	5	2	4	2
people	10	49	12	22	15	15	10	17	15	26	30	11	19	11	10
women	2	7	2	1	1	2	2	0	0	0	2	2	1	1	0

```
In [16]: cfd.plot()
```



6. In the discussion of comparative wordlists, we created an object called `translate` which you could look up using words in both German and Italian in order to get corresponding words in English. What problem might arise with this approach? Can you suggest a way to avoid this problem?

```
In [17]: from nltk.corpus import swadesh
de2en = swadesh.entries(['de', 'en'])
it2en = swadesh.entries(['it', 'en'])
translate2 = dict(de2en)
translate2.update(dict(it2en))
len(translate2)
```

```
Out[17]: 411
```

```
In [18]: translate2['bianco']
```

```
Out[18]: 'white'
```

```
In [19]: translate2['Hund']
```

```
Out[19]: 'dog'
```
