

2021-I_mcpp_taller_9_Valentina_Cuenca

April 30, 2021

1 Taller 9

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 30-abr-2021 11:59 PM

Valentina Cuenca

norma.cuenca@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_matallana
- 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/>), ejercicios: - Capítulo 1: 22, 26, 28 - Capítulo 2: 2, 4, 11

```
[4]: import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = [18.0, 8.0]
import nltk
```

```
[5]: nltk.download('gutenberg')
nltk.download('nps_chat')
nltk.download('webtext')
nltk.download('treebank')
```

```
nltk.download('inaugural')
nltk.download('genesis')
nltk.download('state_union')
nltk.download('brown')
```

```
[nltk_data] Downloading package gutenber to
[nltk_data] C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data] Unzipping corpora\gutenberg.zip.
[nltk_data] Downloading package nps_chat to
[nltk_data] C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data] Unzipping corpora\nps_chat.zip.
[nltk_data] Downloading package webtext to
[nltk_data] C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data] Unzipping corpora\webtext.zip.
[nltk_data] Downloading package treebank to
[nltk_data] C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data] Unzipping corpora\treebank.zip.
[nltk_data] Downloading package inaugural to
[nltk_data] C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data] Unzipping corpora\inaugural.zip.
[nltk_data] Downloading package genesis to
[nltk_data] C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data] Unzipping corpora\genesis.zip.
[nltk_data] Downloading package state_union to
[nltk_data] C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data] Unzipping corpora\state_union.zip.
[nltk_data] Downloading package brown to
[nltk_data] C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data] Unzipping corpora\brown.zip.
```

[5]: True

```
[7]: from nltk.book import *
nltk.book
```

```
*** 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
```

```
[7]: <module 'nltk.book' from 'C:\\Users\\User\\anaconda3\\lib\\site-packages\\nltk\\book.py'>
```

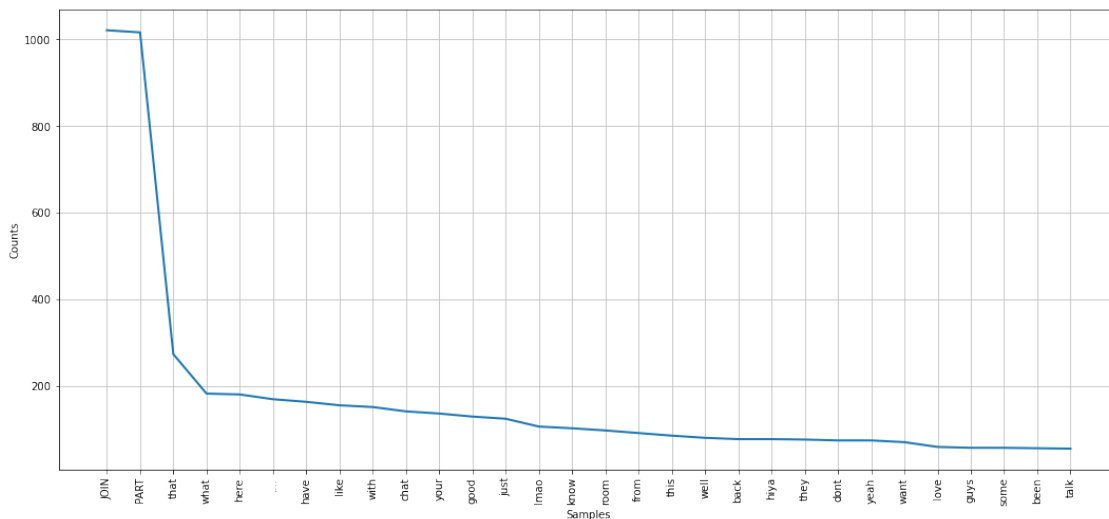
1.2 Capitulo 1

22. 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.

```
[9]: cuatro_palabras = FreqDist([w for w in text5 if len(w) == 4])
print(cuatro_palabras.most_common(100))
```

```
[('JOIN', 1021), ('PART', 1016), ('that', 274), ('what', 183), ('here', 181),
('...', 170), ('have', 164), ('like', 156), ('with', 152), ('chat', 142),
('your', 137), ('good', 130), ('just', 125), ('lmao', 107), ('know', 103),
('room', 98), ('from', 92), ('this', 86), ('well', 81), ('back', 78), ('hiya',
78), ('they', 77), ('dont', 75), ('yeah', 75), ('want', 71), ('love', 60),
('guys', 58), ('some', 58), ('been', 57), ('talk', 56), ('nice', 52), ('time',
50), ('when', 48), ('haha', 44), ('make', 44), ('girl', 43), ('need', 43),
('U122', 42), ('MODE', 41), ('will', 40), ('much', 40), ('then', 40), ('over',
39), ('work', 38), ('were', 38), ('take', 37), ('U121', 36), ('U115', 36),
('song', 36), ('even', 35), ('does', 35), ('seen', 35), ('U156', 35), ('U105',
35), ('more', 34), ('damn', 34), ('only', 33), ('come', 33), ('hell', 29),
('long', 28), ('them', 28), ('name', 27), ('tell', 27), ('away', 26), ('sure',
26), ('look', 26), ('baby', 26), ('call', 26), ('play', 25), ('U110', 25),
('U114', 25), ('NICK', 24), ('down', 24), ('cool', 24), ('sexy', 23), ('many',
23), ('hate', 23), ('said', 23), ('last', 22), ('ever', 22), ('hear', 21),
('life', 21), ('live', 20), ('feel', 19), ('very', 19), ('mean', 19), ('give',
19), ('same', 19), ('must', 19), ('stop', 19), ('LMAO', 19), ('!!!!', 18),
('hugs', 18), ('What', 18), ('find', 18), ('cant', 18), ('left', 17), ('????',
17), ('shit', 17), ('nite', 17)]
```

```
[11]: cuatro_palabras.plot(30);
```



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?

Este código nos da el número de caracteres del `text1`, podríamos utilizar para saber la longitud promedio de las palabras, si lo dividimos por la longitud del texto, pero esto estaría sesgado porque eso incluyen caracteres alfanuméricos, signos de puntuación y demás.

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

```
[14]: 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.

```
[16]: def percent(word, text):  
      porcentaje= 100* text.count(word) /len(text)  
      return porcentaje  
percent("JOIN",text5)
```

```
[16]: 2.2683848033770273
```

1.3 Capítulo 2

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

```
[17]: austen = gutenbergl.words('austen-persuasion.txt')  
austen
```

```
[17]: [' ', 'Persuasion', 'by', 'Jane', 'Austen', '1818', ...]
```

```
[18]: (len(set(word for word in austen if word.isalpha())))
```

```
[18]: 6036
```

```
[19]: len(austen)
```

```
[19]: 98171
```

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?

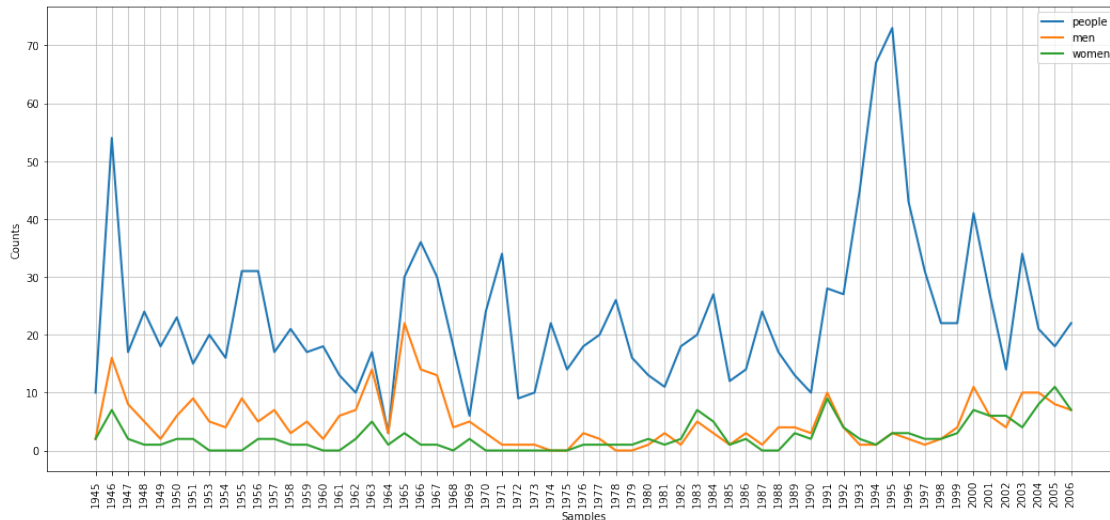
```
[24]: from nltk.corpus import state_union
```

```
[25]: palabras_clave = nltk.ConditionalFreqDist((objetivo, documentos[:4])  
      for documentos in state_union.fileids()  
      for w in state_union.words(documentos))
```

```

for objetivo in ['men', 'women', 'people']
    if w.lower().startswith(objetivo))
palabras_clave.plot();

```



Podemos ver que la palabra people cuando mas se utilizó fue entre 1992 y 1997, mientras que la palabra men se utiliza menos, en comparación a people, a lo largo de todos los años, pero se utiliza más si la comparamos con la palabra women, en el 2005 y 2002 women supera a men, y en el 2006 las dos palabras se utilizan en la misma cantidad. También es interesante entre 1965 y 1968 como men y people aumentaron al mismo tiempo.

- Investigate the table of modal distributions and look for other patterns. Try to explain them in terms of your own impressionistic understanding of the different genres. Can you find other closed classes of words that exhibit significant differences across different genres?

```

[41]: palabras_clave = nltk.FreqDist(( documentos[:100]))
      for documentos in state_union.fileids()
      for w in state_union.words(documentos))
print(palabras_clave.most_common(100))

```

```

[('1946-Truman.txt', 30650), ('1995-Clinton.txt', 10674), ('2000-Clinton.txt',
10640), ('1956-Eisenhower.txt', 9190), ('1999-Clinton.txt', 8762),
('1994-Clinton.txt', 8616), ('1998-Clinton.txt', 8573), ('1967-Johnson.txt',
8221), ('1955-Eisenhower.txt', 8200), ('1993-Clinton.txt', 7961),
('1997-Clinton.txt', 7781), ('1953-Eisenhower.txt', 7777), ('1962-Kennedy.txt',
7603), ('1996-Clinton.txt', 7277), ('1954-Eisenhower.txt', 6774),
('1947-Truman.txt', 6740), ('1961-Kennedy.txt', 6634), ('2006-GWBush.txt',
6515), ('1983-Reagan.txt', 6466), ('1960-Eisenhower.txt', 6311),
('1966-Johnson.txt', 6253), ('2003-GWBush.txt', 6241), ('2004-GWBush.txt',
6240), ('1963-Kennedy.txt', 6218), ('1992-Bush.txt', 6077), ('2005-GWBush.txt',
6053), ('1982-Reagan.txt', 6051), ('1984-Reagan.txt', 5828), ('1974-Nixon.txt',
5805), ('1988-Reagan.txt', 5715), ('1989-Bush.txt', 5708), ('1948-Truman.txt',

```

```
5705), ('1968-Johnson.txt', 5689), ('1950-Truman.txt', 5673), ('1976-Ford.txt',
5617), ('1959-Eisenhower.txt', 5607), ('1958-Eisenhower.txt', 5589),
('2001-GWBush-1.txt', 5384), ('1977-Ford.txt', 5342), ('1978-Carter.txt', 5238),
('1981-Reagan.txt', 5221), ('1970-Nixon.txt', 5019), ('1965-Johnson-1.txt',
4966), ('1985-Reagan.txt', 4935), ('1975-Ford.txt', 4760), ('2002-GWBush.txt',
4689), ('1991-Bush-1.txt', 4661), ('1969-Johnson.txt', 4660),
('1957-Eisenhower.txt', 4633), ('1971-Nixon.txt', 4558), ('1990-Bush.txt',
4549), ('1951-Truman.txt', 4514), ('1972-Nixon.txt', 4512), ('1987-Reagan.txt',
4504), ('1965-Johnson-2.txt', 4192), ('1986-Reagan.txt', 4105),
('1980-Carter.txt', 3871), ('1949-Truman.txt', 3817), ('1979-Carter.txt', 3764),
('1964-Johnson.txt', 3675), ('2001-GWBush-2.txt', 3573), ('1991-Bush-2.txt',
3357), ('1945-Truman.txt', 2138), ('1963-Johnson.txt', 1878), ('1973-Nixon.txt',
1873)]
```

```
[30]: state_union.fileids()
```

```
[30]: ['1945-Truman.txt',
'1946-Truman.txt',
'1947-Truman.txt',
'1948-Truman.txt',
'1949-Truman.txt',
'1950-Truman.txt',
'1951-Truman.txt',
'1953-Eisenhower.txt',
'1954-Eisenhower.txt',
'1955-Eisenhower.txt',
'1956-Eisenhower.txt',
'1957-Eisenhower.txt',
'1958-Eisenhower.txt',
'1959-Eisenhower.txt',
'1960-Eisenhower.txt',
'1961-Kennedy.txt',
'1962-Kennedy.txt',
'1963-Johnson.txt',
'1963-Kennedy.txt',
'1964-Johnson.txt',
'1965-Johnson-1.txt',
'1965-Johnson-2.txt',
'1966-Johnson.txt',
'1967-Johnson.txt',
'1968-Johnson.txt',
'1969-Johnson.txt',
'1970-Nixon.txt',
'1971-Nixon.txt',
'1972-Nixon.txt',
'1973-Nixon.txt',
'1974-Nixon.txt',
```

'1975-Ford.txt',
'1976-Ford.txt',
'1977-Ford.txt',
'1978-Carter.txt',
'1979-Carter.txt',
'1980-Carter.txt',
'1981-Reagan.txt',
'1982-Reagan.txt',
'1983-Reagan.txt',
'1984-Reagan.txt',
'1985-Reagan.txt',
'1986-Reagan.txt',
'1987-Reagan.txt',
'1988-Reagan.txt',
'1989-Bush.txt',
'1990-Bush.txt',
'1991-Bush-1.txt',
'1991-Bush-2.txt',
'1992-Bush.txt',
'1993-Clinton.txt',
'1994-Clinton.txt',
'1995-Clinton.txt',
'1996-Clinton.txt',
'1997-Clinton.txt',
'1998-Clinton.txt',
'1999-Clinton.txt',
'2000-Clinton.txt',
'2001-GWBush-1.txt',
'2001-GWBush-2.txt',
'2002-GWBush.txt',
'2003-GWBush.txt',
'2004-GWBush.txt',
'2005-GWBush.txt',
'2006-GWBush.txt']
