mcpp_taller9_Julian_Ramirez

November 1, 2019

1 Taller 9

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 1-nov-2019 11:59 PM

Julián Santiago Ramírez
julians.ramirez@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/), Exercises: - Chapter 1: 22, 26, 28 - Chapter 2: 2, 4, 11

2 Capítulo 1

```
In [15]: import nltk
```

```
In [6]: from nltk.book import *
```

```
*** Introductory Examples for the NLTK Book ***
Loading text1, ..., text9 and sent1, ..., sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
text7: Wall Street Journal
text8: Personals Corpus
text9: The Man Who Was Thursday by G . K . Chesterton 1908
```

3 Punto 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.

```
In [9]: ### Buscamos las palabras que tengan tamaño 4
        frecuencia = FreqDist(i for i in text5 if len(i) == 4)
        ## En este caso para verlo mas claron imprimo solo 50 de manera decreciente, porque so
        print(frecuencia.most_common(50))
```

```
[('JOIN', 1021), ('PART', 1016), ('that', 274), ('what', 183), ('here', 181), ('...', 170), ('I
```

4 Punto 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?

4.0.1 Rta:

Lo que realiza el código es que recorre cada palabra del texto 1 y luego cuenta el número de caracteres que hay en el texto 1

```
In [10]: total = sum(len(w) for w in text1)
         print("Cantidad de carácteres en el texto 1: ", total)
         ## Necesito primero la cantidad de palabras que hay en el texto 1
         total_palabras = len(text1)
         ## Realizo el promedio
         promedio = total / total_palabras
         print("El promedio del tamaño de una palabra en el texto 1 es: ", promedio)
```

```
Cantidad de carácteres en el texto 1:  999044
El promedio del tamaño de una palabra en el texto 1 es:  3.830411128023649
```

5 Punto 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 [14]: # Funcion percent, dos argumentos
         def percent(word, text):
             ## primero cuento las veces que aparece la palabra en el texto
             cuenta=text.count(word)
             ## Luego miro que tan largo es el texto
             tamaño= len(text)
             ## Porcentaje
             por=(cuenta/tamaño)*100
             return por

         ## Pruebas
         a=percent('God',text3)
         print("Porcentaje de cantidad de veces que aparece 'God' en el texto 3: ",a)
         b=percent('Finance',text7)
         print("Porcentaje de cantidad de veces que aparece 'Finance' en el texto 7: ",b)
```

```
Porcentaje de cantidad de veces que aparece 'God' en el texto 3:  0.5160396747386293
Porcentaje de cantidad de veces que aparece 'Finance' en el texto 7:  0.008939568516826254
```

6 Capítulo 2

7 Punto 2

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

```
In [21]: # Traemos el archivo
         austen=nltk.corpus.gutenberg.words('austen-persuasion.txt')
         print('Número de tokens:',len(austen))
         print('Tipos de palabras:',len(set(austen)))
```

```
Número de tokens: 98171
Tipos de palabras: 6132
```

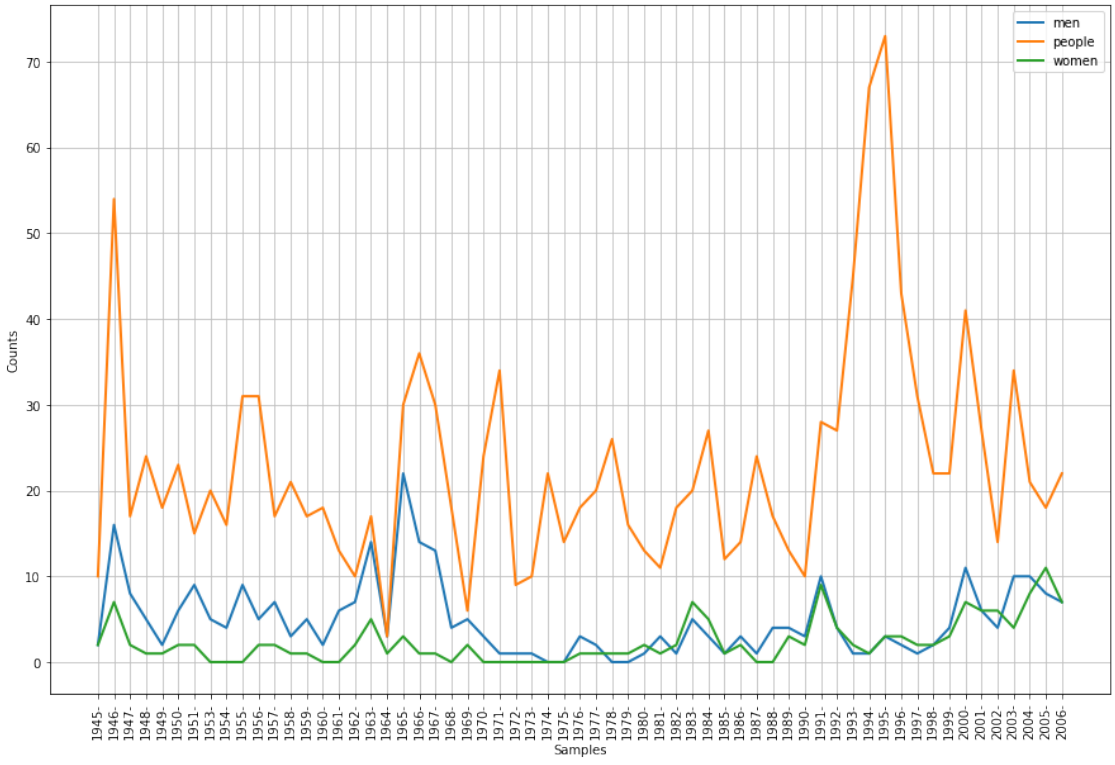
8 Punto 4

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 [25]: from nltk.corpus import state_union
         import matplotlib.pyplot as plt

In [44]: ## Frecuencia Condicional
         # Palabras que queremos buscar
         lista=['men', 'women', 'people']
         ## Realizamos la busqueda por año
         DistF= nltk.ConditionalFreqDist(
             (i, fileid[:4])
             # Recorremos el texto y buscamos que si cada palabra coincide con las que estamos
             for fileid in state_union.fileids()
             for j in state_union.words(fileid)
             for i in lista
             # En caso tal de que si coincida
             if j.lower().startswith(i))

In [45]: plt.figure(figsize=(15,10))
         DistF.plot()
```



9 Punto 11

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?

Segun el texto: "The Brown Corpus is a convenient resource for studying systematic differences between genres, a kind of linguistic inquiry known as stylistics. Let's compare genres in their usage of modal verbs. The first step is to produce the counts for a particular genre. Remember to import nltk before doing the following:" Luego importamos brown

```
In [47]: from nltk.corpus import brown
         ## Tomado del libro los modals serian
         modals = ['can', 'could', 'may', 'might', 'must', 'will']
         ## Tomado del libro
         genres = ['news', 'religion', 'hobbies', 'science_fiction', 'romance', 'humor']
         tabla = nltk.ConditionalFreqDist(
                     (genre, word.lower())
                     for genre in brown.categories()
                     for word in brown.words(categories=genre))
         tabla.tabulate(conditions=genres, samples=modals)
```

	can	could	may	might	must	will
news	94	87	93	38	53	389
religion	84	59	79	12	54	72
hobbies	276	59	143	22	84	269
science_fiction	16	49	4	12	8	17
romance	79	195	11	51	46	49
humor	17	33	8	8	9	13

- Humor: Es el que usa menos los diferentes modales, bajo mi impresion no son muy requeridos en este genero.
- Hobbies: Usa en gran manera "can", "will" y "may" indica que las personas expresan en mayor medida que pueden o realizaran algunas actividades
- News: Es el que usa mas la palabra will, es comun que una noticia estos indican que sucederá despues del acontecimiento.
- Romance: Es el que usa mas la palabra "could" indicando la posible existencia de diferentes cosas que no son seguras que sucedan en el futuro.
- Religion: Se usa las palabras can y may. Tiene sentido es comun que aquí se expresen mas oraciones con mandamientos o cosas que puede realizar una persona.

```
In [59]: # Mis palabras
         julian_pal = ['but','then','also','however','thus','so']
         ## Tomado del libro
         genres = ['news', 'religion', 'hobbies', 'science_fiction', 'romance', 'humor']
         tabla = nltk.ConditionalFreqDist(
                     (genre, word.lower())
                     for genre in brown.categories()
                     for word in brown.words(categories=genre))
         tabla.tabulate(conditions=genres, samples=julian_pal)
```

	but	then	also	however	thus	so
news	283	73	129	49	10	81
religion	175	59	57	19	24	99

hobbies	221	88	110	41	21	121
science_fiction	89	18	2	7	2	26
romance	387	143	16	2	0	192
humor	101	37	13	10	5	56

En este caso use conectores. Podemos ver que en "news" y "romance" es mas comun usar "but" seguramente porque siempre se muestran diferentes puntos de vista. Las palabras "so" y "then" se usa mucho en hobbies y romance indicando que dado que paso algo, esto conllevo a que les guste algo o que dos personas se separaran. En humor tambien se exponen diferentes puntos de vista pero con menor frecuencia

---