IM-UH 1511 **Introduction to Digital Humanities**

HOMEWORK 1

Extraction of Names from Text

50 points totally

In [13]:
```
import time
start_time = time.clock()
import urllib, os, codecs, random, operator, re, string, copy, dateutil.par
from collections import Counter
from string import punctuation, digits
import pathlib
import spacy
from spacy import displacy
nlp = spacy.load('en_core_web_lg')
import inflect
import nltk
from nltk import word_tokenize

import warnings
warnings.filterwarnings("ignore", category=RuntimeWarning)
warnings.simplefilter('ignore')
```

Load Data

In [2]:
```
# get your working directory
home = str(pathlib.Path.cwd())

# create a path to which the file will be written
text_path = os.path.join(home, 'Dracula.txt')

# location of the project gutenberg copy of the moby-dick text file
text_url = 'http://www.gutenberg.org/cache/epub/345/pg345.txt'

urllib.request.urlretrieve(text_url, text_path)

print('Downloaded to:', text_path)
```

```
Downloaded to: /Users/mb7881/WorkPlaces/Python Projects 2/3 NYUAD Digital
Humanities/Homework1 NamesExtraction/Dracula.txt
```

In [3]:
```
f = codecs.open(text_path, "r", encoding="utf-8").readlines()
for line in f:
    if line.startswith("(_Kept in shorthand._)"):
        print(f.index(line)) #198
    if line.startswith("                                    THE END"):
        print(f.index(line)) #15514
```

```
198
15514
```

In [4]:
```
ff=f[194:15514]
ff
```

Out[4]:
```
['CHAPTER I\r\n',
 '\r\n',
 "JONATHAN HARKER'S JOURNAL\r\n",
 '\r\n',
 '(_Kept in shorthand._)\r\n',
 '\r\n',
 '\r\n',
 '_3 May. Bistritz._--Left Munich at 8:35 P. M., on 1st May, arriving at
\r\n',
 'Vienna early next morning; should have arrived at 6:46, but train was a
n\r\n',
 'hour late. Buda-Pesth seems a wonderful place, from the glimpse which I
\r\n',
 'got of it from the train and the little I could walk through the\r\n',
 'streets. I feared to go very far from the station, as we had arrived\r
\n',
 'late and would start as near the correct time as possible. The\r\n',
 'impression I had was that we were leaving the West and entering the\r
\n',
```

In [5]:
```
text="\r\n".join(ff)
text
```

Out[5]:
```
'CHAPTER I\r\n\r\n\r\n\r\nJONATHAN HARKER\'S JOURNAL\r\n\r\n\r\n\r\n(_Kep
t in shorthand._)\r\n\r\n\r\n\r\n\r\n\r\n_3 May. Bistritz._--Left Munich
at 8:35 P. M., on 1st May, arriving at\r\n\r\nVienna early next morning;
should have arrived at 6:46, but train was an\r\n\r\nhour late. Buda-Pest
h seems a wonderful place, from the glimpse which I\r\n\r\ngot of it from
the train and the little I could walk through the\r\n\r\nstreets. I feare
d to go very far from the station, as we had arrived\r\n\r\nlate and woul
d start as near the correct time as possible. The\r\n\r\nimpression I had
was that we were leaving the West and entering the\r\n\r\nEast; the most
western of splendid bridges over the Danube, which is\r\n\r\nhere of nobl
e width and depth, took us among the traditions of Turkish\r\n\r\nrule.\r
\n\r\n\r\n\r\nWe left in pretty good time, and came after nightfall to Kl
ausenburgh.\r\n\r\nHere I stopped for the night at the Hotel Royale. I ha
d for dinner, or\r\n\r\nrather supper, a chicken done up some way with re
d pepper, which was\r\n\r\nvery good but thirsty. (_Mem._, get recipe for
Mina.) I asked the\r\n\r\nwaiter, and he said it was called "paprika hend
l," and that, as it was a\r\n\r\nnational dish, I should be able to get i
t anywhere along the\r\n\r\nCarpathians. I found my smattering of German
very useful here; indeed, I\r\n\r\ndon\'t know how I should be able to ge
```

In [6]:
```
titlename = "Bram Stoker's Dracula"

words = word_tokenize(text)
nuw=len(words)
uw=len(set(words))
print("%s contains %i nonunique and %i unique words"%(titlename,nuw,uw))
```

```
Bram Stoker's Dracula contains 189685 nonunique and 10627 unique words
```

Extraction of Proper Nouns

In [7]:
```
p = inflect.engine()
d_tags = {}

docs_d={"Dracula":text}
for key, value in docs_d.items():
    arr = []
    doc = nlp(value.replace('\n',''))
    #Keep these types of nlp entities
    keep_l = ['PERSON'] #,'NORP','PRODUCT','ORG']
    #Typo/model error + german corrections
    drop_t = []

    #Things inflect library handles poorly or to exclude from touching
    ex_ls = []

    for X in doc.ents:
        s1 = X.text
        if (X.label_ in keep_l) and (s1.lower() not in drop_t) and (s1):
            arr.append((s1, X.label_))
    d_tags[key] = arr
# pprint(d_tags)
names=[]
for k,v in d_tags.items():
    for vv in v:
        if vv[0] not in names:
            p=vv[0].replace("'","")
            p=p.title()
            names.append(p)
names=sorted(set(names))
print(len(names))
names
```

```
385
```

In [8]:
```
rem=[]
for p in names:
    if "_" in p:
        rem.append(p)
    if "--" in p:
        rem.append(p)
    if p not in text:
        rem.append(p)
names=[p for p in names if p not in rem]
pp=[q for q in itertools.product(names,names) if q[0]!=q[1]]
for q in pp:
    if q[0] in q[1]:
        rem.append(q[0])
    if q[1] in q[0]:
        rem.append(q[1])
    w=q[0]+" "+q[1]
    if w in text:
        names.append(w)
        rem.append(q[0])
        rem.append(q[1])
names=[p for p in names if p not in rem]
names=sorted(set(names))
print(len(names))
names
```

```
165
```

In [9]:
```
rem=['Breakfast','Chin','Crucifix','Wafer','Ye','D. Lit','Devil','Draculas'
    'Friend Arthur','H. B. M.','Harker Jonathan','Herr','Harkers','Hun',
    'I. He','Ittin','Lively','Lookin','Lordship','M. D.','M. R. C. S. L. K.
    'Manlike','Mein Gott','Mem','Moneybag','Moon','Nature','Morris Quincey'
    'Nay','Omne','Ounds','Pass','Seein','Telegram','Yus','Arthur Ho','Cszek
    'Stay','Stop','Soh','This Braithwaite Lowrey','Lord God','Robin Ho',
    'Quincey P. Morris','Faugh','Mitchell','Jonathan Harker','Mina Harker',
    'Sister','Agatha',"Joseph","Miss Westenra"]
names=[p for p in names if p not in rem]
names=names+['Robin Hood','Soho','Braithwaite Lowrey','Van Helsing',
            'Mitchell, Sons, & Candy','Jonathan','Mina','Sister Agatha',
            'Count Dracula','Ste. Mary','St. Joseph',"Saxon"]
names=sorted(set(names))
print(len(names))
names
```

```
133
```

In [10]:
```
nfreq=[]
for i in names:
    nfreq.append(text.count(i))
pnf_df = pd.DataFrame(
    {'Names': names,
     'Frequency of Occurrences': nfreq
    })
pnf_df=pnf_df[['Names','Frequency of Occurrences']]
pnf_df=pnf_df.sort_values(by ='Frequency of Occurrences',ascending=False)
# trf_df=trf_df[trf_df["Frequency of Occurrences"]>10]
print(len(pnf_df))
pnf_df[:50]
```

```
133
```

Out[10]:

	Names	Frequency of Occurrences
120	Van Helsing	294
78	Mina	226
59	Jonathan	193
68	Lord Godalming	64
91	Renfield	48
127	Whitby	43
90	Quincey Morris	22
39	Galatz	18
20	Catherine	17
69	Lucy Westenra	17
119	Turk	16
112	Szgany	13
10	Bersicker	9
79	Mina Murray	9
48	Hillingham	9
7	Arthur Holmwood	9
23	Count Dracula	9
75	Mary	8
14	Bukovina	8
58	John Seward	6
100	Sereth	6
24	Crescent	6
110	Swales	6
11	Bistritza	6
13	Buda-Pesth	6
61	Kettleness	5

	Names	Frequency of Occurrences
38	Fundu	5
122	Veresti	5
64	Kukri	5
111	Szekelys	4
36	Esk	4
86	Peter Hawkins	4
104	Sister Agatha	4
93	Roumanian	4
108	Ste. Mary	3
35	Enoch	3
52	Jack Seward	3
25	Demeter	3
123	Vincent	3
115	Thor	3
41	Hampstead Heath	3
71	Magyars	3
19	Castle Dracula	3
99	Saxon	3
125	Wallach	3
92	Robin Hood	3
107	St. Joseph	3
98	Samuel F. Billington	2
80	Mitchell, Sons, & Candy	2
94	Rufus Smith	2

In [11]:
```
pnf_df.to_csv('Names_freqs.csv')
```

In [12]:
```
print("Run in %.2f seconds (%.2f minutes)" %(time.clock() - start_time,(tim
```

```
Run in 44.77 seconds (0.75 minutes)
```