Processing Text Practice Assignment

AUTHOR: 'STUDENT NAME'

DATE: 'INSERT DATE'

```
#Load Libraries
from urllib.request import urlopen
from bs4 import BeautifulSoup
import pandas as pd
from nltk.tokenize import sent_tokenize, word_tokenize
import re
import html
import PyPDF2
import os
```

Step 1: Find a webpage and scrape it

- Find a public html webpage like shown in class and read it into Python using urlopen

```
#Step 1: Websites

links = ['https://chicago.suntimes.com/politics/2024/05/31/donald-trump-hush-money-conviction-illinois-republicans-reactio
texts = []

for link in links:
    page = urlopen(link).read()
    soupified = BeautifulSoup(page, "html.parser")
    html = soupified.find('div', {'class': 'RichTextArticleBody RichTextBody'})
    text = html.get_text().strip() # Option 1
    #text = re.findall("<p>(.*?)</p>", str(html)) #Option 2
    texts.append(text)

print(texts)
```

```
['In a state where Donald Trump has divided many Republicans, Illinois GOP leaders were united in their condemnation of
```

```
#Step 1: Files

# Using csv
df = pd.read_csv('inaugural addresses.csv', encoding='latin1', header=None)
df.rename(columns = {0:'Filename', 1:'Number', 2 :'President', 3:'Term', 4:'Text'}, inplace=True)

# Using pdfs
pdfs_list = []
in_dir = 'PDFs/'
for infile in os.listdir(in_dir):
    infile = in_dir + infile
    in_pdf = open(infile, 'rb')
    pdfReader = PyPDF2.PdfReader(in_pdf)
    pages_obj = pdfReader.pages
    text = []
    for page in pages_obj:
        text.append(page.extract_text())
    pdfs_list.append(text)

print(pdfs_list)
```

```
[['Information Processing and Management 58 (2021) 102674\nAvailable online 21 July 2021\n0306-4573/© 2021 Elsevier Ltd.
```

Step 2: Fix Errors

- Examine the text for errors or problems by looking at the text (done above).
- Deal with the issues involved in HTML.

```
#Step 2: Selecting Text from HTML
# This step is entirely dependant on your specific text and may be skipped in the data is already clean

# for text in range(len(texts)): #Needed to loop through list and change items in the list
#     texts[text] = re.sub('\\n', ' ', texts[text]) #Use extra \ is need to literally match \n
#     texts[text] = re.sub('(References.*$)', ' ', texts[text]) # Put parentheses around text to remove

# x = re.search("The scientific method is an empirical method for acquiring", texts[0]).span(0)[0] #Find the index for the
# texts[0] = (texts[0])[x:] #Extracts the string from the sentence above to the end of the text

# x = re.search("Science fiction \(sometimes shortened to SF or sci-fi\) is a genre of", texts[1]).span(0)[0] #Find the in
# texts[1] = (texts[1])[x:] #Extracts the string from the sentence above to the end of the text

for text in texts:
    print(text)

#Add any further cleaning necessary to create human readable text
```

```
In a state where Donald Trump has divided many Republicans, Illinois GOP leaders were united in their condemnation of th




Illinois Republican Party Chairman Don Tracy speaks during the Republican Day rally at the Illinois State Fair in August


“Everyone sees what happened in Manhattan, the second most Democrat county in the nation — where Democrat prosecutors ar
Related


How will the Biden campaign exploit the Trump hush money guilty verdict?



Even self-declared “Never-Trump” Illinois Republicans slammed the verdict, including former Illinois House Minority Lead
```

```
#Step 2: Selecting Text from PDFs

for pdf in pdfs_list:
    pdf = re.sub('\\\\n', ' ', str(pdf))
    pdf = re.sub('(.*?)ABSTRACT|(.*?)Abstract', '', str(pdf))# The | means find string before or after
    pdf = re.sub('- ', '', str(pdf))
    pdf = re.sub('REFERENCES(.*?)|References(.*?)', '', str(pdf))
    print(pdf)

#Add any further cleaning necessary to create human readable text
```

```
   Ethnicity-targeted hate speech has been widely shown to influence on-the-ground inter-ethnic  conflict and violence,
 The datasets most widely used for abusive language detection contain lists of messages, usually tweets, that have been
 — Whether intentionally or not, many individuals in  society express some form of prejudice or bias in their thinking
```

Step 3: Clean text with regex

- Use the “impurity” function from class to examine the text for potential issues.
- Remove the noise with the regex function.
- Re-examine the impurity to determine if the data has been mostly cleaned.

```
#Step 3

import re

RE_SUSPICIOUS = re.compile(r'[&#<>{}\[\]\\]')

def impurity(text, min_len=10):
    """returns the share of suspicious characters in a text"""
    if text == None or len(text) < min_len:
        return 0
    else:
        return len(RE_SUSPICIOUS.findall(text))/len(text)

def clean_up(text):
    # tags like <tab>
    # find things that are < STUFF > that aren't <<>> because [^<>] means doesn't start with
    text = re.sub(r'<[^<>]*>', ' ', text)
    # markdown URLs like [Some text](https://....)
    text = re.sub(r'\[([^\[\]]*)\]\([^\(\)]*\)', r'\1', text)
    # text or code in brackets like [0]
    text = re.sub(r'\[[^\[\]]*\]', ' ', text)
    # standalone sequences of specials, matches &# but not #cool
    text = re.sub(r'(?:^|\s)[&#<>{}\[\]+|\\:-]{1,}(?:\s|$)', ' ', text)
    # standalone sequences of hyphens like --- or ==
    text = re.sub(r'(?:^|\s)[\-=\+]{2,}(?:\s|$)', ' ', text)
    # get rid of bylines
    # starts with ( any number of characters in the middle and ends with )
    text = re.sub(r'^\(.*\)$', ' ', text)
    # sequences of white spaces
    text = re.sub(r'\s+', ' ', text)
    return text.strip()

#Applying to a list
for pdf in range(len(pdfs_list)):
    pdfs_list[pdf] = clean_up(str(pdfs_list[pdf]))
    print(impurity(str(pdf)))
```

```
0
0
0
```

```
#Applying to column in dataframe
df['impurity'] = df['Text'].apply(impurity)
df['impurity'].sort_values()
```

```
0     0.000000
31    0.000000
32    0.000000
33    0.000000
34    0.000000
35    0.000000
36    0.000000
37    0.000000
38    0.000000
39    0.000000
40    0.000000
41    0.000000
42    0.000000
30    0.000000
43    0.000000
45    0.000000
46    0.000000
47    0.000000
48    0.000000
49    0.000000
50    0.000000
51    0.000000
52    0.000000
53    0.000000
54    0.000000
55    0.000000
56    0.000000
44    0.000000
57    0.000000
29    0.000000
27    0.000000
1     0.000000
2     0.000000
3     0.000000
4     0.000000
5     0.000000
6     0.000000
7     0.000000
8     0.000000
9     0.000000
10    0.000000
11    0.000000
28    0.000000
13    0.000000
15    0.000000
16    0.000000
18    0.000000
19    0.000000
20    0.000000
21    0.000000
22    0.000000
23    0.000000
24    0.000000
25    0.000000
26    0.000000
14    0.000000
58    0.000000
12    0.000085
17    0.000119
Name: impurity, dtype: float64
```

Step 4: Print output

- Print out the first 1000 characters and the number of tokens in the text.

```
import nltk
nltk.download('punkt')

#Applying to list
for text in texts:
    print(str(text[0:1000]))
    print(len(word_tokenize(str(text))))

#Applying to dataframe
df['tokens'] = df['Text'].apply(word_tokenize)
df['tokens'].apply(len)
```

```
In a state where Donald Trump has divided many Republicans, Illinois GOP leaders were united in their condemnation of th
1069
NEW YORK — Donald Trump became the first former president to be convicted of felony crimes Thursday as a New York jury f
1433
Gov. J.B. Pritzker plans to talk about the threat to “fundamental freedoms” that another Donald Trump presidency poses t
501
```

```
0     1524
1      145
2     2579
3     1917
4     2375
5     1263
6     1304
7     3682
8     4880
9     3148
10    1207
11    1264
12    4215
13    6000
14    5170
15    1177
16    3633
17    3073
18    3989
19     777
20    1225
21    1472
22    2702
23    3207
24    1813
25    2130
26    4723
27    4352
28    2438
29    1080
30    5821
31    1887
32    1653
33    3729
34    4437
35    4063
36    2062
37    1988
38    1528
39     633
40    2494
41    2755
42    1869
43    1513
44    1709
45    2388
46    1975
47    1368
48    2744
49    2879
50    2658
51    1813
52    2447
53    1805
54    2326
55    2701
56    2503
57    1651
58    2607
Name: tokens, dtype: int64
```

```
[nltk_data] Downloading package punkt to /home/datalore/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Interpretation

- Write a paragraph explaining the process of cleaning data for an NLP pipeline. You should explain the errors you found in the dataset and how you fixed them. Discuss the importance of text preprocessing to the NLP pipeline.

Answer

- [Expectings 4-6 sentences describing how you had to process this specific set of texts not just the generic pipeline from the slides.]