AIT526 Individual Lab1

Due Date: Please check the class schedule on blackboard.

Text Preprocessing & Word Clouds

Programming Tools:

1) **Python 3**
2) **Jupyter Lab** (Desktop or online) or Desktop **Jupyter Notebook**
3) **NLTK** (https://www.nltk.org/)
   (Install: https://www.nltk.org/install.html
   `pip install --user -U nltk` **or** `conda install -c anaconda nltk`)
4) **Wordcloud** (https://amueller.github.io/word_cloud/)
   (Install: `pip install wordcloud` **or** `conda install -c conda-forge wordcloud`)

Suggested References:

1) Class 2 Lecture slides
2) **NLTK video tutorials** on Blackboard
3) Python NLP tutorial: Using NLTK for natural language processing
4) **Wordcloud** code examples
5) **Code snippets, hints, plots, and running outputs** in this assignment
6) NLTK textbook and other sources on Internet

Note that you must include **reference(s) in the code comments when you refer others' work.*

Data Location: Blackboard/Assignments/Optional Labs/Lab 1/ Harry Potter Book 1.txt

Tasks (10 points):

In the **first** lab, a few very fundamental applied NLP techniques to process and explore the text data is introduced. This **step-by-step** lab-based tutorial is specially designed to help you learn ***fast*** and have ***fun*** with NLP. Please follow the detailed **step-by-step instructions** to use NLTK**,** Python, and Wordcloud for **text preprocessing, basic text analysis,** and **word cloud generation**.

Task 0:

Please review *suggested* **code examples/tutorials** and **watch the videos** to practice NLTK programming. *Note: Task 0 is not for grading. You may skip it if familiar with these but they will help Task 1 & 2.*

Task 1 (5 points): Text Preprocessing and Basic Analysis with NLTK

Please complete the subtasks:

1.1 (0 points) Load one text file.

Hint: The code snippets are:

```
fo = open("Harry Potter Book 1.txt", "r", encoding='utf-8')
mytext = fo.read()
```

1.2 (1 point) Tokenize sentences and words. Print lens of sentences and tokens and only print the first 20 words.

Hint: The output is similar to:

```
# of sentences:  6394
# of words after word tokenizing:  98781
['Harry', 'Potter', 'and', 'the', 'Sorcerer', "'s", 'Stone', 'CHAPTER', 'ONE', 'THE',
'BOY', 'WHO', 'LIVED', 'Mr.', 'and', 'Mrs.', 'Dursley', ',', 'of', 'number']
```

1.3 (1 point) Remove punctuations. Only print the first 20 words.

Hint: The output is similar to:

```
# of words after punctuation removing:  80646
['harry', 'potter', 'and', 'the', 'sorcerer', 's', 'stone', 'chapter', 'one', 'the',
'boy', 'who', 'lived', 'mr', 'and', 'mrs', 'dursley', 'of', 'number', 'four']
```

1.4 (1 point) Remove stop words and count the cleaned words.

Hint: Several different ways can be used to count words. You also can FreqDist(). The output should be similar to:

```
# of words without stop words:  40785
['harry', 'potter', 'sorcerer', 'stone', 'chapter', 'one', 'boy', 'lived', 'mr', 'mr
s', 'dursley', 'number', 'four', 'privet', 'drive', 'proud', 'say', 'perfectly', 'no
rmal', 'thank']
```

```
<FreqDist with 5628 samples and 40785 outcomes>
```

1.5 (1 point) Lemmatize the cleaned words and count the lemmatized words. What's different from 1.4? Please clearly explain.

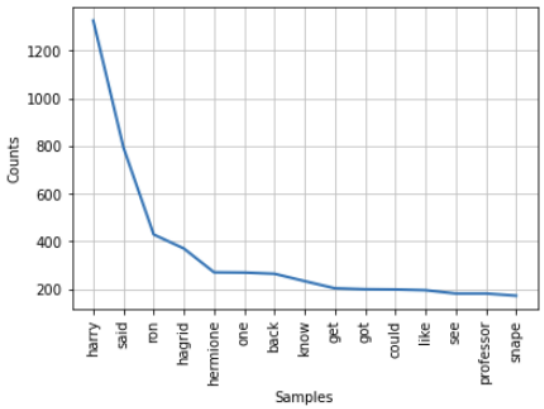
*Hint: Note that you can use NLTK WordNet Lemmatizer. By default, only **nouns** are lemmatized without tagged POS. If so, the output is similar to:*

```
<FreqDist with 5109 samples and 40785 outcomes>
```

1.6 (1 point) Calculate the word distribution and plot and list only top 15 words.

Hint: Use FreqDist(). The outputs is similar to:

```
[('harry', 1327),
 ('said', 794),
 ('ron', 429),
 ('hagrid', 370),
 ('hermione', 270),
 ('one', 269),
 ('back', 264),
 ('know', 233),
 ('get', 203),
 ('got', 199),
 ('could', 198),
 ('like', 195),
 ('see', 181),
 ('professor', 181),
 ('snape', 172)]
```

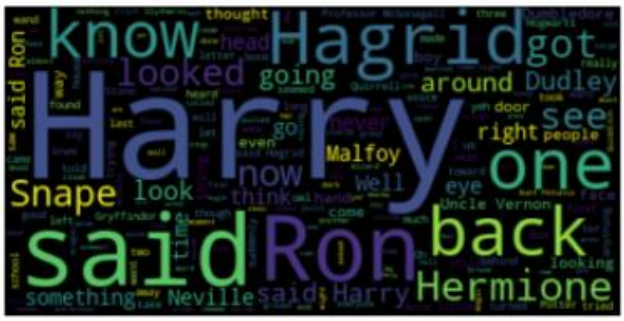


Task 2 (4 points): Word Clouds

In this task, we are going to use a few different ways to generate word clouds using Wordcloud package.

2.1 (1.5 points) Follow the **wordcloud simple example** to generate the word cloud **without stopwords** directly from the original text and show it. Note that you must remove the stopwords.

Hint: Use WordCloud(stopwords=STOPWORDS).generate(mytext). The output is similar to:



2.2 (1.5 points) Generate the word cloud for the processed data in Task 1 and show the image.

Hint: Use WordCloud().generate_from_frequencies(freq_dist). Note that freq_dist is the output of FreqDist(). The output is similar to:

2.3 (1 point) Compare the above two wordclouds and what difference do you find? Please clearly explain why.

Task 3 (1 point):

Please answer the following questions:

3.1 Based on your experience with these NLP tasks, please compare with text data processing and numeric data processing.

SUBMISSION

- Write all your code and answers with explanation in the Notebook.
- In the code file, please do not forget to write your name, course #, and date in the comments.
- **Run ALL Cells**:

Open your IPython file in Jupyter, go to **Run**->**Run All Cells**. Please make sure all of you code has been run and print out the results.

- **Save to HTML**:
  Go to **File**-> **Export Notebook As…**->**Export Notebook to HTML**, and save your work into HTML file.
- **Submission**:
  a. Write your work with two file names **"AIT526_YourName_Lab1.ipynb"** and **"AIT526_YourName_ Lab1.HTML"**.
  b. **Zip** both files to **ONE zipped file** since blackboard does not allow you to submit HTML file separately.
  c. Go to the Blackboard **/Course Content/Optional Individual Labs/** to submit **ONE zipped file.**