Arnau Manzano Ramells
U150397

HEEJUNG
U209834

INFORMATION RETRIEVAL AND WEB ANALYTICS

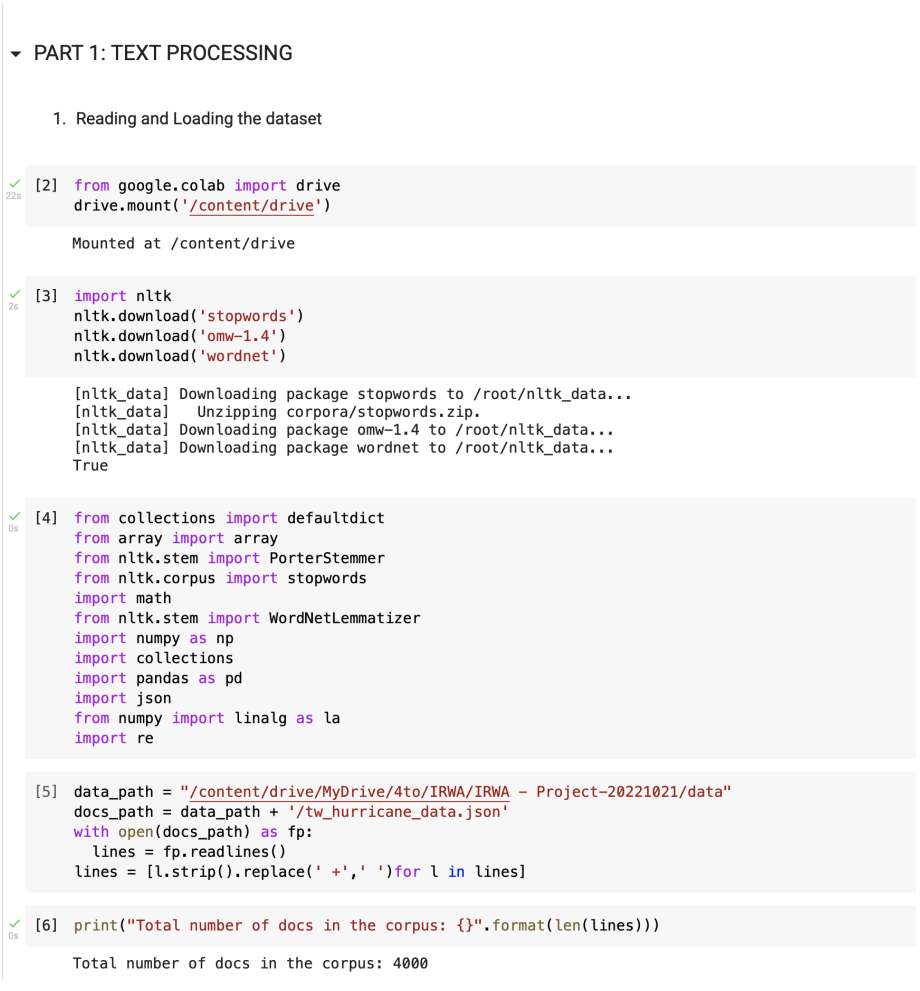
FINAL PROJECT

GitHub

All code for the project is submit to and can be found in the repository at
https://github.com/byunheejung1028/IRWA-2022-part1

PART 1: TEXT PROCESSING

For the first part of the final project we have to preprocess the documents. We've had to import the document corpus which is a set of tweets related to Hurricane Ian (tw_hurricane_data.json), all the libraries necessaries and the suggested library **nlkt.**



PART 1: TEXT PROCESSING

1. Reading and Loading the dataset

```
[2]  from google.colab import drive
     drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
[3]  import nltk
     nltk.download('stopwords')
     nltk.download('omw-1.4')
     nltk.download('wordnet')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data] Downloading package wordnet to /root/nltk_data...
True
```

```
[4]  from collections import defaultdict
     from array import array
     from nltk.stem import PorterStemmer
     from nltk.corpus import stopwords
     import math
     from nltk.stem import WordNetLemmatizer
     import numpy as np
     import collections
     import pandas as pd
     import json
     from numpy import linalg as la
     import re
```

```
[5]  data_path = "/content/drive/MyDrive/4to/IRWA/IRWA - Project-20221021/data"
     docs_path = data_path + '/tw_hurricane_data.json'
     with open(docs_path) as fp:
       lines = fp.readlines()
     lines = [l.strip().replace(' +',' ')for l in lines]
```

```
[6]  print("Total number of docs in the corpus: {}".format(len(lines)))
```

```
Total number of docs in the corpus: 4000
```

Then we have created a series of functions:

```
def getId(tweet):

  return tweet ['id']
```

```
def text(tweet):

  return tweet['full_text']
```

```
def username(tweet):

  return tweet['user']['name']
```

```
def hashtags(tweet):

  return [hashtag['text'] for hashtag in tweet['entities']['hashtags']]
```

```
def url(tweet):

  return 'https://twitter.com/_/status/'+str(tweet['id'])
```

```
def date(tweet):

  return tweet['created_at']
```

```
def likes(tweet):

  k = tweet ['favorite_count']

  if k:
    return k
  else:
    return 0
```

```
def retweets(tweet):

  k = tweet['retweet_count']

  if k:
    return k
  else:
    return 0
```

We'll use these to implement another function to get to the final output format that we are asked: **Tweet | Username | Date | Hashtags | Likes | Retweets | Url**. In this new function we put together all the information from the other functions implemented before and we'll return a dictionary and the text to preprocess.

```
def final_output(tweet):

  output = {}
  tweet = json.loads(tweet)
  output['Tweet'] = text(tweet) #This is not tokenized/stemmed with build_terms()
  output['Id'] = getId(tweet)
  output['Username'] = username (tweet)
  output['Date'] = date(tweet)
  output['Hashtags'] = hashtags(tweet)
  output['Likes'] = likes(tweet)
  output['Retweets'] = retweets(tweet)
  output['Url'] = url(tweet)
  return output, str(output['Tweet'])
```

Then we're going to use the function seen in the first lab, **build_terms** to:

- Removing stop words
- Tokenization
- Removing punctuation marks
- Stemming
- and... anything else you think it's needed (bonus point)

In addition to these steps, we have also removed new lines, hashtags, tags, URLs, empty strings and lemmatized the words. We consider that we'll better approach the final output format.

```
def build_terms(line):
    """
    Preprocess the text removing stop words, stemming,
    transforming in lowercase rfeturn the tokens of the text...
    """

    tweet, line = final_output(line)

    stemmer = PorterStemmer()
    lemmatizer = WordNetLemmatizer()
    stop_words = set(stopwords.words("english"))

    line = line.lower()
    line = re.sub(r'http\S+', ' ', line) # delete urls
    line = re.sub(r'@\S+',' ', line) # delete tags
    line = re.sub(r'[^\w\s]', ' ', line) # delete punctuation
    line = line.split()  # Tokenize the text to get a list of terms
    line = [re.sub(r'\\n', '', x) for x in line]
    line = [x.replace('#', '') for x in line] # delete hashtag symbol
    line = [x for x in line if x not in stop_words]  # delete the stopwords
    empty = ['', ' ']
    line = [x for x in line if x not in empty] #delete empty strings '', ' '
    line = [stemmer.stem(word) for word in line] # perform stemming (HINT: use List Comprehension)
    line = [lemmatizer.lemmatize(x) for x in line] # lemmatize words
    return tweet, line
```

Finally we compare the tweet's id with the document id (tweet_document_ids_map.csv) to see if they are the same. As we can see in the next photo, they are the same.

[57]
```
import pandas as pd
docs_path = '/content/drive/MyDrive/4to/IRWA/IRWA - Project-20221021/data/tweet_document_ids_map.csv'
df = pd.read_csv(docs_path, sep='\t', header = None)
df
```

	0	1
0	doc_1	1575918182698979328
1	doc_2	1575918151862304768
2	doc_3	1575918140839673873
3	doc_4	1575918135009738752
4	doc_5	1575918119251419136
...	...	...
3995	doc_3996	1575856268022992896
3996	doc_3997	1575856245650919424
3997	doc_3998	1575856228886089728
3998	doc_3999	1575856226139017216
3999	doc_4000	1575856225908326400

4000 rows × 2 columns

[62]
```
#checking tweet id with the document ids
print(processed_documents[0]['Id'])
print(processed_documents[1]['Id'])
print(processed_documents[2]['Id'])
print(processed_documents[4]['Id'])
```

```
1575918182698979328
1575918151862304768
1575918140839673873
1575918119251419136
```