scraper_doc_parser

December 12, 2020

[5]:

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

```
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/sahiltyagi/nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/sahiltyagi/nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /Users/sahiltyagi/nltk_data…
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

[5]:

```
True
```

PARSING JANE AUSTEN BOOKS FROM GUTENBERG DOCUMENTS

[6]:

```
import urllib.request
from bs4 import BeautifulSoup
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import os
from nltk.tokenize import RegexpTokenizer

#removing pronouns, prepositions, conjunctions and interjections, verbs and␣
 ↪adverbs
what_to_filter_out =␣
 ↪['CC','EX','IN','PRP','PRP$','UH','WDT','WP','WP$','WRB','VB','VBG','VBD','VBP','VBZ','DT',
 ↪','RB']

jane_austen_homepage = {}
file = open(os.getcwd() + '/jane_austen.txt', 'r')
for line in file:
    jane_austen_homepage[str(line.split(',')[1])] = str(line.split(',')[0])
```

```
file.close()

for url in jane_austen_homepage.keys():
    stop_words = set(stopwords.words('english'))
    tokenizer = RegexpTokenizer(r'\w+')
    html = urllib.request.urlopen(url)
    f = html.read()
    soup = BeautifulSoup(f, 'html.parser')
    book = []
    tags = soup.find_all('p')
    for tag in tags:
        para = ''
        word_tokens = word_tokenize(str(tag.get_text()))
        for word in word_tokens:
            if word not in stop_words and 'CHAPTER' not in word:
                para = para + str(word) + ' '

        #tokenizer.tokenize(para)
        for w in tokenizer.tokenize(para):
            filter_words = word_tokenize(w)
            tagged_words = nltk.pos_tag(filter_words)
            # to remove all 's' from the contractions like it's, I'm etc...
            if tagged_words[0][1] not in what_to_filter_out and␣
 ↪tagged_words[0][0] != 's':
                book.append(str(tagged_words[0][0]))

    file = open(os.getcwd() + '/books/' + jane_austen_homepage[url] + '.txt',␣
 ↪'w')
    for line in book:
        file.write(line + ' ')

    file.close()
    print('did the book ' + str(jane_austen_homepage[url]) + ' so far..')
```

```
did the book Persuasion so far..
did the book The Letters of Jane Austen so far..
did the book Pride and Prejudice so far..
did the book Emma so far..
did the book Sense and Sensibility so far..
did the book Northanger Abbey so far..
did the book Mansfield Park so far..
did the book The Watsons: By Jane Austen and Concluded by L. Oulton so far..
did the book Lady Susan so far..
did the book Love and Freindship so far..
```

USE WORD FREQUENCY FROM EACH BOOK TO GET SOME IDEAS FOR TOPICS

[7]:

```
books_dir = os.getcwd() + '/books/'
files = os.listdir(books_dir)
for file in files:
    if '.txt' in file:
        word_count = {}
        print(books_dir + file)
        f = open(books_dir + '/' + file, 'r')
        for line in f:
            for word in line.split():
                if word in word_count.keys():
                    word_count[word] = word_count[word] + 1
                else:
                    word_count[word] = 1

        f.close()
        all_counts = list(word_count.values())
        all_counts.sort(reverse=True)

        K = 200
        top_K = all_counts[0:K]

        popular_words = []
        for k,v in word_count.items():
            if v in top_K:
                popular_words.append(k)

        topic_file = open(os.getcwd() + '/top_K_topics/' + file, 'w')
        for topic in popular_words:
            topic_file.write(topic + '\n')

        topic_file.close()
```

```
/Users/sahiltyagi/Desktop/gutenberg/books/The Watsons: By Jane Austen and
Concluded by L. Oulton.txt
/Users/sahiltyagi/Desktop/gutenberg/books/Northanger Abbey.txt
/Users/sahiltyagi/Desktop/gutenberg/books/Sense and Sensibility.txt
/Users/sahiltyagi/Desktop/gutenberg/books/Mansfield Park.txt
/Users/sahiltyagi/Desktop/gutenberg/books/Persuasion.txt
/Users/sahiltyagi/Desktop/gutenberg/books/Emma.txt
/Users/sahiltyagi/Desktop/gutenberg/books/Pride and Prejudice.txt
/Users/sahiltyagi/Desktop/gutenberg/books/The Letters of Jane Austen.txt
/Users/sahiltyagi/Desktop/gutenberg/books/Lady Susan.txt
/Users/sahiltyagi/Desktop/gutenberg/books/Love and Freindship.txt
```

[85]:

```
ss = word_tokenize('perhaps')
tw = nltk.pos_tag(ss)
print(tw[0][0])
print(tw[0][1])
```

```
perhaps
RB
```