Implement a simple web crawler

import requests

import lxml

from bs4 import BeautifulSoup

url = "https://www.rottentomatoes.com/top/bestofrt/"

headers = {

'User-Agent':'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36 OPR/50.0.2762.58 (Edition Yx 01)'

}

f = requests.get(url, headers = headers)

movies\_lst = []

soup = BeautifulSoup(f.content, 'html.parser')

movies = soup.find('table', {

'class':'table'

}) .find\_all('a')

print(movies)

num = 0

for anchor in movies:

urls = 'https://www.rottentomatoes.com' +anchor['href']

movies\_lst.append(urls)

print(movies\_lst)

num += 1

movie\_url = urls

movie\_f = requests.get(movie\_url, headers=headers)

movie\_soup = BeautifulSoup(movie\_f.content, 'lxml')

movie\_content = movie\_soup.find('div', {

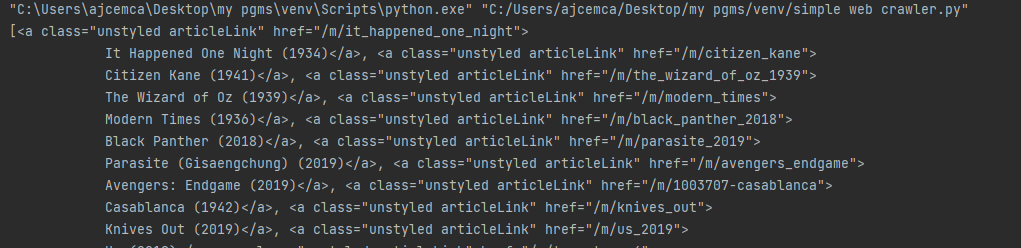
'class':'movie\_synopsis clamp clamp-6 js-clamp'

})

print(num, urls, '/n', 'Movie:'+anchor.string.strip())

print('Movie info:' + movie\_content.string.strip())

OUTPUT





from bs4 import BeautifulSoup

import requests

pages\_crawled = []

def crawler(url):

page = requests.get(url)

soup = BeautifulSoup(page.text, 'html.parser')

links = soup.find\_all('a')

for link in links:

if 'href' in link.attrs:

if link['href'].startwidth('/wiki') and ':' not in link['href']:

if link['href'] not in pages\_crawled:

new\_link = f"https://en.wikepedia.org{link['href']}"

pages\_crawled.append(link['href'])

try:

with open('data.csv') as file:

file.write(f'{soup.title.text}; {soup.h1.text};{link["href"]}\n')

crawler(new\_link)

except:

continue

crawler('https://en.wikipedia.org')

Implement a program to scrap the web page of any popular website

import csv

import requests

from bs4 import BeautifulSoup

url="http://www.values.com/inspirational-quotes"

r=requests.get(url)

print("Content:")

print(r.content)

print("Prettify:")

soup=BeautifulSoup(r.content,'lxml')

print(soup.prettify())

quotes=[]

table=soup.find('div',attrs={'id':'all\_quotes'})

for row in table.find\_all('div',attrs={'class':'col-6 col-lg-3 text-center margin-30px-bottom sm-margin-30px-top'}):

quote={}

quote['theme']=row.h5.text

quote['url']=row.a['href']

quote['img']=row.img['src']

quote['lines']=row.img['alt'].split("#")[0]

quote['author']=row.img['alt'].split("#")[1]

quotes.append(quote)

filename='insipration\_quotation.csv'

with open(filename,'w',newline='')as f:

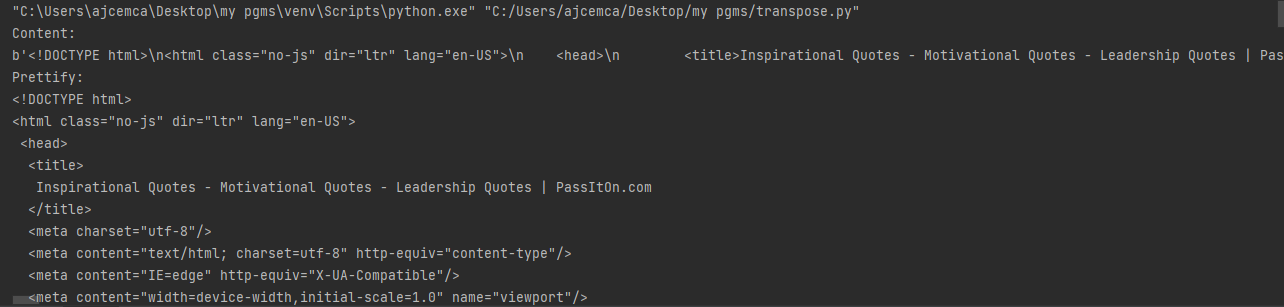
w=csv.DictWriter(f,['theme','url','img','lines','author'])

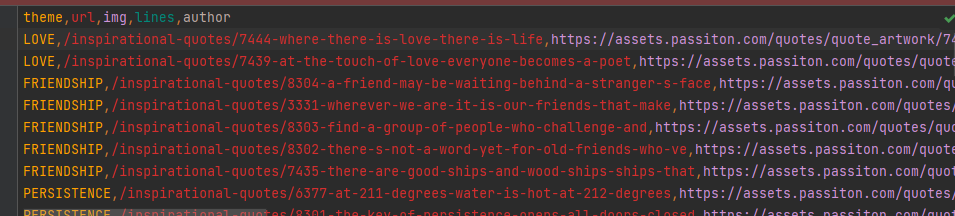
w.writeheader()

for quote in quotes:

w.writerow(quote)

OUTPUT





Python program for natural language processing N-gram and smoothening(Without inbuilt)

def generate\_ngrams(text,WordsToCombine):

words=text.split()

output=[]

for i in range(len(words)-WordsToCombine+1):

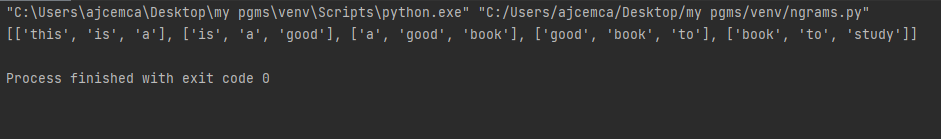
output.append(words[i:i + WordsToCombine])

return output

x=generate\_ngrams(text="this is a good book to study",WordsToCombine=3)

print(x)

OUTPUT



python for natural language processing N-gram and smoothening(With inbuilt)

import nltk

from nltk.util import ngrams

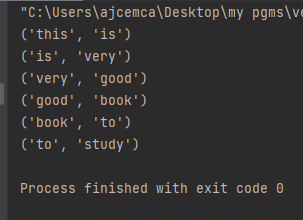
samplText="this is very good book to study"

Ngrams=ngrams(sequence=nltk.wordpunct\_tokenize(samplText),n=2)

for grams in Ngrams:

print(grams)

OUTPUT



Python program for natural language processing Part of Speech tagging

import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize, sent\_tokenize

stop\_words = set(stopwords.words('english'))

txt = "Sukanya, Rajib and Naba are my good friends. " \

"Sukanya is getting married next year. " \

"Marriage is a big step in one’s life." \

"It is both exciting and frightening. " \

"But friendship is a sacred bond between people." \

"It is a special kind of love between us. " \

"Many of you must have tried searching for a friend " \

"but never found the right one."

tokenized = sent\_tokenize(txt)

for i in tokenized:

wordsList = nltk.word\_tokenize(i)

wordsList = [w for w in wordsList if not w in stop\_words]

tagged = nltk.pos\_tag(wordsList)

print(tagged)

OUTPUT

[('Sukanya', 'NNP'), (',', ','), ('Rajib', 'NNP'), ('Naba', 'NNP'), ('good', 'JJ'), ('friends', 'NNS'), ('.', '.')]

[('Sukanya', 'NNP'), ('getting', 'VBG'), ('married', 'VBN'), ('next', 'JJ'), ('year', 'NN'), ('.', '.')]

[('Marriage', 'NN'), ('big', 'JJ'), ('step', 'NN'), ('one', 'CD'), ('’', 'NN'), ('life.It', 'NN'), ('exciting', 'VBG'), ('frightening', 'NN'), ('.', '.')]

[('But', 'CC'), ('friendship', 'NN'), ('sacred', 'VBD'), ('bond', 'NN'), ('people.It', 'NN'), ('special', 'JJ'), ('kind', 'NN'), ('love', 'VB'), ('us', 'PRP'), ('.', '.')]

[('Many', 'JJ'), ('must', 'MD'), ('tried', 'VB'), ('searching', 'VBG'), ('friend', 'NN'), ('never', 'RB'), ('found', 'VBD'), ('right', 'JJ'), ('one', 'CD'), ('.', '.')]