# -\*- coding: utf-8 -\*-

# from konlpy.tag import Okt

import sys

import io

sys.stdout = io.TextIOWrapper(sys.stdout.detach(),encoding='utf-8')

sys.stderr = io.TextIOWrapper(sys.stderr.detach(),encoding='utf-8')

from ckonlpy.tag import Twitter

import pickle

from collections import Counter

import requests

import csv

import os

import re

from bs4 import BeautifulSoup

import pandas as pd

from datetime import datetime,date

from dateutil.relativedelta import relativedelta

from wordcloud import WordCloud

import warnings

# sending email

import smtplib

from email.mime.multipart import MIMEMultipart

from email.mime.base import MIMEBase

from email import encoders

warnings.filterwarnings("ignore")

now = datetime.now()

# 회사별 제거해야할 단어 리스트 [대웅제약, 한미약품, 유한양행, 종근당]

remove0 = ['edaily', 'MoneyToday', '국민일보', '뉴스1', '뉴시스', '데일리안', '디지털타임스', '매경닷컴', '머니S', '서울경제', '세계닷컴', '스포츠조선',

'한경닷컴', '한국경제TV', '코메디닷컴']

remove1 = ['대웅', '제약', '대웅제약', '대웅바이오']

remove2 = ['한미', '약품', '한미약품']

remove3 = ['유한', '양행', '유한양행']

remove4 = ['종근당', '종근당홀딩스', '종근당건강', '종근당바이오', '홀딩스']

removedict = {'대웅제약': remove1, '한미약품': remove2, '유한양행': remove3, '종근당': remove4}

# 날짜 지정

## 기간: 어제~오늘

senddate = '%s.%s.%s' % (now.year, now.month, now.day)

ed = now.strftime('%Y-%m-%d')

sd = (now - relativedelta(days=1)).strftime('%Y-%m-%d')

ed = ed.replace('-', '.')

sd = sd.replace('-', '.')

# print("%s~%s" % (sd, ed))

# 수집할 회사: 대웅, 한미, 유한, 종근당

item = ['대웅제약', '한미약품', '유한양행', '종근당']

# item = ['종근당']

# 불용어 담긴 csv 읽어 리스트 만들기

add = 'C:/Users/User/Desktop/wordcloud/ver4/'

fl = open('%sfixed\_korean\_stopwords.csv' % add, 'r', encoding='UTF8')

files = csv.reader(fl)

stopwords = []

for file in files:

stopwords.extend(file)

stopwords = set(stopwords)

address = 'C:/Users/User/Desktop/wordcloud/ver4/data/'

# 최신순으로 한 네이버 뉴스에서 오늘부터 어제자까지의 뉴스 크롤링

## 시작 링크가 네이버뉴스, 메디파나, 약업신문, 약사공론, 데일리팜인 경우 전부 크롤링 가능

## 그외의 경우 기사 제목과 상위에 뜨는 문장만 끌어오는 것으로 변경

def naver\_crawler(maxpage, query, s\_date, e\_date):

print("<%s>" % query)

s\_from = s\_date.replace('.', '')

e\_to = e\_date.replace('.', '')

page = 1

# maxpage\_t = (int(maxpage) - 1) \* 10 + 1 # 11= 2페이지 21=3페이지 31=4페이지 ...81=9페이지 , 91=10페이지, 101=11페이지

maxpage\_t = 87945

total = []

while page < maxpage\_t:

url = "https://search.naver.com/search.naver?&where=news&query="+ query +"&sm=tab\_pge&sort=1&photo=0&field=0&reporter\_article=&pd=0&ds=&de=&docid=&nso=so:dd,p:all,a:all&mynews=0&start="+ str(page)+"&refresh\_start=0"

# url = "http://search.naver.com/search.naver?where=news&query=" + query + "&sm=tab\_pge&sort=1&photo=0&field=0&reporter\_article=&pd=3&ds=1990.01.01&de=2020.06.17&docid=&nso=so:dd,p:from19900101to20200617,a:all&mynews=0&start=" + str(

# page)+"&refresh\_start=0&related=0"

url = "https://search.naver.com/search.naver?where=news&query="+query+"&sm=tab\_srt&sort=1&photo=0&field=0&reporter\_article=&pd=0&ds=&de=&docid=&nso=so%3Add%2Cp%3Aall%2Ca%3Aall&mynews=0&start="+str(page)+"&refresh\_start=0&related=0"

# print(url)

req = requests.get(url)

cont = req.content

soup = BeautifulSoup(cont, 'html.parser')

# totalpage = soup.select(".title\_desc.all\_my")[0].span.get\_text().split('/')[1].replace(',', '')

# totalpage = re.sub('[가-힣]', '', totalpage)

# maxpage\_t = int(totalpage)

# print(soup)

for content in soup.select('ul.list\_news > li > div > div.news\_area'):

try:

# url이 naver, dailypharm, yakup, medipana, kpanews로 시작하면 url 안으로 들어가 내용까지 전부 크롤링 함

href = content.select('a.news\_tit')[0]['href']

if href.startswith("https://news.naver.com"):

inf = content.select('div.news\_info > div > a.info')[0].get\_text()

tt = content.select('a.news\_tit')[0]['title'].replace('\n', '').strip()

news\_detail = get\_naver(href, query,inf,tt)

# 주식관련 뉴스봇 기사 제외

if news\_detail[0] == '조선비즈':

continue

total.append(news\_detail)

elif href.startswith("http://www.dailypharm.com"):

news\_detail = get\_dailypharm(href, query)

total.append(news\_detail)

elif href.startswith('http://www.yakup.com'):

news\_detail = get\_yakup(href, query)

total.append(news\_detail)

elif href.startswith('http://medipana.com/'):

news\_detail = get\_medipana(href, query)

total.append(news\_detail)

elif href.startswith('https://www.kpanews.co.kr'):

news\_detail = get\_kpanews(href, query)

total.append(news\_detail)

# 그 외의 경우 기사 제목, 회사, 기사 상단 크롤링 진행

else:

news\_detail = ['', '', '', '', '']

news\_detail[0] = content.select('div.news\_info > div > a.info')[0].get\_text()

# 주식 관련 뉴스 사이트 제외

if news\_detail[0] == '글로벌이코노믹':

continue

news\_detail[1] = href

news\_detail[2] = content.select('a.news\_tit')[0]['title'].replace('\n', '').strip()

news\_detail[3] = content.select('div.news\_dsc > div > a')[0].get\_text()

try:

# string = content.select('span.info')[0].get\_text()

# temp = string.split()

# news\_detail[4] = temp[1]

string = content.select('div.news\_info > div > span.info')[0].get\_text()

news\_detail[4] = string.replace(' ','')

except Exception as e:

print(e)

p = re.sub('[0-9]', '', news\_detail[4])

if p == '분' or p == '시간':

news\_detail[4] = ed

elif p == '일':

for i in range(1, 8):

if news\_detail[4] == '%s일' % str(i):

news\_detail[4] = (now - relativedelta(days=i)).strftime('%Y-%m-%d').replace('-', '.')

total.append(news\_detail)

# print(news\_detail)

except Exception as e:

continue

# 한 page에 기사 10개 존재하므로 +10

page += 10

print(page, '/', maxpage\_t)

return total

def get\_kpanews(n\_url, it):

news\_detail = ['', '', '', '', '']

breq = requests.get(n\_url)

bsoup = BeautifulSoup(breq.content, 'html.parser')

news\_detail[0] = '약사공론'

news\_detail[1] = n\_url

title = bsoup.select('div.inr-c2 > p.h1')[0].get\_text()

news\_detail[2] = title

text = bsoup.select("div.bbs\_cont.inr-c2 > div")[0].get\_text()

text = re.sub(r"\[[^)]\*\]", '', text).replace('\t', '').replace('\n', '').strip()

news\_detail[3] = text

date = bsoup.select('p.t1.t\_date > span')[0].get\_text()[:10].replace('-', '.')

news\_detail[4] = date

return news\_detail

def get\_medipana(n\_url, it):

news\_detail = ['', '', '', '', '']

breq = requests.get(n\_url)

bsoup = BeautifulSoup(breq.content, 'html.parser')

news\_detail[0] = '메디파나'

news\_detail[1] = n\_url

title = bsoup.select('div.newView > div.tit')[0].get\_text()

news\_detail[2] = title

text = bsoup.select("div.newView > div.newsCon").get\_text().replace(' ', '')

text = re.sub(r"\[[^)]\*\]", '', text).replace('\t', '').replace('\n', '').strip()

news\_detail[3] = text

date = bsoup.select('div.infor > span.data')[0].get\_text()[:10].replace('-', '.')

news\_detail[4] = date

return news\_detail

def get\_yakup(n\_url, it):

news\_detail = ['', '', '', '', '']

breq = requests.get(n\_url)

bsoup = BeautifulSoup(breq.content, 'html.parser')

news\_detail[0] = '약업신문'

news\_detail[1] = n\_url

title = bsoup.select('div#content > h4.h4Type01')[0].get\_text()

news\_detail[2] = title

text = bsoup.select("div.bodyarea")[0].get\_text().replace(' ', '').replace('트윗하기', '')

text = re.sub(r"\[[^)]\*\]", '', text).replace('\t', '').replace('\n', '').strip()

news\_detail[3] = text

date = bsoup.select('div.clear.tp\_5 > p > span')[0].get\_text()[5:15].replace('-','.')

news\_detail[4] = date

return news\_detail

def get\_dailypharm(n\_url, it):

news\_detail = ['', '', '', '', '']

breq = requests.get(n\_url)

bsoup = BeautifulSoup(breq.content, 'html.parser')

news\_detail[0] = '데일리팜'

news\_detail[1] = n\_url

title = bsoup.select('div.d\_newsHead > div.d\_newsTitle')[0].get\_text().replace(' ', '').replace('\n', '')

news\_detail[2] = title

text = bsoup.select("div.d\_newsContWrap > div.newsContents.font1")[0].get\_text().replace(' ', '')

text = re.sub(r"\[[^)]\*\]", '', text).replace('\t', '').replace('\n', '').strip()

news\_detail[3] = text

date = bsoup.select('div.d\_newsName\_wrap > span.d\_newsDate')[0].text[:10]

news\_detail[4] = date.replace('-','.')

return news\_detail

def get\_naver(n\_url, it, info, title):

news\_detail = ['', '', '', '', '']

news\_detail[0] = info

news\_detail[1] = n\_url

news\_detail[2] = title

breq = requests.get(n\_url)

bsoup = BeautifulSoup(breq.content, 'html.parser')

# bsoup = bsoup.encode('cp949')

# bsoup = bsoup.decode('cp949')

# pcompany = bsoup.select('#footer address')[0].a.get\_text()

# news\_detail[0] = pcompany

# news\_detail[1] = n\_url

# title = bsoup.select('h3#articleTitle')[0].text

# news\_detail[2] = title

# print(title)

print(bsoup.select('#articleBodyContents').text)

\_text = bsoup.select('#articleBodyContents')[0].get\_text().replace('\n', " ").replace('\t', " ")

btext = \_text.replace("// flash 오류를 우회하기 위한 함수 추가 function \_flash\_removeCallback() {}", "").replace(

"[이 기사는 증권플러스(두나무)가 자체 개발한 로봇 기자인", "").replace(' ⓒ 전자신문 & 전자신문인터넷, 무단전재 및 재배포 금지]', '')

btext = btext.replace(", 무단 전재 및 재배포 금지 -", "").replace("조선닷컴 바로가기", "").replace("바로가기", "").replace(

"네이버 메인에서 조선일보 받아보기", "")

btext = btext.replace("▶뉴스레터 '매콤달콤' 구독", "").replace("▶네이버 메인에서 '매일경제'를 받아보세요", "").replace(

"▶'M코인' 지금 가입하면 5000코인 드려요", "").replace(" 무단전재 및 재배포 금지]", "").replace("[김시균 기자]", "")

btext = btext.replace(" 무단전재 및 재배포 금지>", "").replace("<저작권자 ⓒ '성공을 꿈꾸는 사람들의 경제 뉴스' 머니", "").replace("mt.co.kr", "")

btext = btext.replace("▶ 고수들의 재테크 비법 영상", "").replace("▶ 코로나19 속보", "").replace("구독신청하기", "")

btext = btext.replace("경제를 실험한다~ 머니랩 [네이버TV]", "").replace("아침을 여는 뉴스 모닝벨 [#오늘의키워드]", "").replace(

" I&M 무단전재-재배포 금지", "")

text = btext.replace("무단 전재", "").replace("전재", "").replace("재배포 금지", "").replace("재배포금지", "").replace("네이버 메인에서",

"").replace(

"받아보기", "").replace('무단전재', '').replace('일보', '')

text = re.sub('▶[^;]\*', '', text)

# eliminate everything after last reporter name

# eliminate emails

text = re.sub('\S\*@\S\*\s?', '', text)

# eliminate reporter names

text = re.sub('[가-힣]{3} 기자 = [가-힣]{3}', '', text)

# eliminate newspaper names

text = re.sub('[가-힣]+=[가-힣]+뉴스', '', text)

# eliminates ' and ()

text = re.sub("'|\(|\)", '', text)

# eliminates anything between []

text = re.sub("\[(.\*?)\]", '', text)

text = re.sub('이데일리', '', text)

# 무단전재 제거

text = re.sub('구독신청하기', '', text)

btext = re.sub('무단전재', '', text)

for de in remove0:

btext = btext.replace(de, "")

if it == "대웅제약":

for de in remove1:

btext = btext.replace(de, "")

elif it == "한미약품":

for de in remove2:

btext = btext.replace(de, "")

elif it == "유한양행":

for de in remove3:

btext = btext.replace(de, "")

else:

for de in remove4:

btext = btext.replace(de, "")

news\_detail[3] = btext.strip()

pdate = bsoup.select('.t11')[0].get\_text()[:10]

news\_detail[4] = pdate

print(news\_detail)

return news\_detail

def excel\_make(it, results):

df = pd.DataFrame(results, columns=['press', 'link', 'title', 'article', 'date'])

results = df.drop\_duplicates('link', keep='last')

try:

results.to\_csv('../data/%s\_naverresult.csv' % it, index=False, encoding='cp949')

except:

results.to\_csv('../data/%s\_naverresult.csv' % it, index=False)

def today(query):

print(query)

page = 1

check = False

c2 = False

total = []

s\_from = sd.replace('.', '')

e\_to = ed.replace('.', '')

maxpage\_t = 300

while page < maxpage\_t:

url = "https://search.naver.com/search.naver?&where=news&query="+ query +"&sm=tab\_pge&sort=1&photo=0&field=0&reporter\_article=&pd=0&ds=&de=&docid=&nso=so:dd,p:all,a:all&mynews=0&start="+ str(page)+"&refresh\_start=0"

# url = "http://search.naver.com/search.naver?where=news&query=" + query + "&sm=tab\_opt&sort=0&photo=0&field=0&reporter\_article=&pd=3&ds=" + sd + "&de=" + ed + "&docid=&nso=so%3Ar%2Cp%3Afrom" + s\_from + "to" + e\_to + "%2Ca%3A&start=" + str(

# page)+"all&mynews=0&refresh\_start=0&related=0"

url = "https://search.naver.com/search.naver?where=news&query="+query+"&sm=tab\_srt&sort=1&photo=0&field=0&reporter\_article=&pd=0&ds=&de=&docid=&nso=so%3Add%2Cp%3Aall%2Ca%3Aall&mynews=0&start="+str(page)+"&refresh\_start=0&related=0"

req = requests.get(url)

cont = req.content

soup = BeautifulSoup(cont, 'html.parser')

# if c2 is False:

# totalpage = soup.select(".title\_desc.all\_my")[0].span.get\_text().split()

# totalp = totalpage[2].replace('건', '').replace(',','')

# maxpage\_t = int(totalp)\*10

# print('총 기사 수:', totalp)

# c2 = True

# for content in soup.select("ul.list\_news > li > dl"):

for content in soup.select("ul.list\_news > li > div > div.news\_area"):

# print(content)

# content = content.encode('cp949')

try:

# href = content.select('a.\_sp\_each\_url')[0]['href']

href = content.select('a.news\_tit')[0]['href']

try:

href = content.select('div.news\_info > div > a.info')[1]['href']

# print(href)

except:

None

# print(href)

if href.startswith("https://news.naver.com"):

inf = content.select('div.news\_info > div > a.info')[0].get\_text()

tt = content.select('a.news\_tit')[0]['title'].replace('\n', '').strip()

# print(inf, tt)

news\_detail = get\_naver(href, query, inf, tt)

# print(news\_detail)

# 주식관련 뉴스봇 기사 제외

if news\_detail[0] == '조선비즈':

continue

total.append(news\_detail)

elif href.startswith("http://www.dailypharm.com"):

news\_detail = get\_dailypharm(href, query)

elif href.startswith('http://www.yakup.com'):

news\_detail = get\_yakup(href, query)

elif href.startswith('http://medipana.com/'):

news\_detail = get\_medipana(href, query)

elif href.startswith('https://www.kpanews.co.kr'):

news\_detail = get\_kpanews(href, query)

else:

# print(href)

news\_detail = ['', '', '', '', '']

news\_detail[0] = content.select('div.news\_info > div > a.info')[0].get\_text()

# print(news\_detail[0])

# 주식 관련 뉴스 사이트 제외

if news\_detail[0] == '글로벌이코노믹':

continue

news\_detail[1] = href

news\_detail[2] = content.select('a.news\_tit')[0]['title'].replace('\n', '').strip()

news\_detail[3] = content.select('div.news\_dsc > div > a')[0].get\_text()

# print(news\_detail[3])

try:

string = content.select('div.news\_info > div > span.info')[0].get\_text()

news\_detail[4] = string.replace(' ','')

# temp = string.split()

# news\_detail[4] = temp[1]+temp[2]

except Exception as e:

print(e)

# print(content.select('dd.txt\_inline')[0].text)

# sp\_nws21 > dl > dd.txt\_inline

p = re.sub("[0-9]", '', news\_detail[4]) # 한글 추출

# num = re.sub("[^0-9]", '', news\_detail[4]) # 숫자 추출

if p == '분전' or p == '시간전':

news\_detail[4] = ed

elif news\_detail[4] == '1일 전':

news\_detail[4] = sd

# print(news\_detail)

if news\_detail[4] not in [sd, ed]:

check = True

break

# print(news\_detail[1])

total.append(news\_detail)

# print(news\_detail)

except Exception as e:

# print(e)

continue

page += 10

if check:

break

# print(total)

return total

def addnews(it):

try:

df = pd.read\_csv('%s%s\_naverresult.csv' % (address, it),encoding='cp949')

except:

df = pd.read\_csv('%s%s\_naverresult.csv' % (address, it))

results = pd.DataFrame(today(it), columns=['press', 'link', 'title', 'article', 'date'])

total\_results = pd.concat([results, df])

total\_results = total\_results.drop\_duplicates('link', keep='last')

try:

total\_results.to\_csv('%s%s\_naverresult.csv' % (address, it), index=False,encoding='cp949')

except:

total\_results.to\_csv('%s%s\_naverresult.csv' % (address, it), index=False)

def navernews():

for it in item:

filename = '%s%s\_naverresult.csv' % (address, it)

if os.path.isfile(filename):

addnews(it)

if (datetime.fromtimestamp(os.path.getmtime('%s%s\_naverresult.csv' % (address, it))).strftime('%G%m%d')) == date.today().strftime('%G%m%d'):

continue

addnews(it)

else:

data = naver\_crawler(100, it, sd, ed)

excel\_make(it, data)

navernews()