Open API를 이용한 News Crawling

Naver와 한국인터넷진흥원(KISA) Open API를 활용한 뉴스 크롤링 코드입니다. 기존 부서에서 사용하는 크롤링 코드 활용해서 작성했습니다.

제가 코드에 써둔 API Key를 사용해도 괜찮지만, 차후에 재발급하면 사용이 어려울 수 있으므로 코드 사용 전에 아래 사이트에서 Open API키를 새로 발급받아 사용하는 것을 추천드립니다.

NAVER Developers

네이버 오픈 API들을 활용해 개발자들이 다양한 애플리케이션을 개발할 수 있도록 API 가이드와 SDK를 제공합니다. 제공중인 오픈 API에는 네이버 로그인, 검색, 단축URL, 캡차를 비롯 기계번역, 음성인식, 음성합성 등이 있습니다.

https://developers.naver.com/main/

한국인터넷진흥원_인터넷주소(도메인이름, 아이피) 정보 검색 서비스

인터넷주소자원인 도메인, IP 주소, AS 번호의 등록정보 및 할당정보를 조회 할 수 있는 서비스(WHOIS 오픈API)

https://www.data.go.kr/data/15094277/openapi.do

0. Library for Import

```
## 0. Library for Import
import requests
import pandas as pd
import urllib3
from datetime import datetime

urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning) # SSL인증서 오류 무시

# Naver Developer에서 발급받은 Open API Key
client_id = "muRJ15UeEhqBN14xyuXR"
clientKey = "DlEuaSNLMj"

# 한국인터넷진흥원(KISA)에서 발급받은 Open API Key
KISA_enc = "ezYdB3TSFLXri3DA4wBTmOdhgS7mIzuIcnAb2e%2BABCJM3L7vBWFnP%2FYq7v34%2Bq%2BxTvAnh0JqU3rYjb44kwOVqQ%3D%3D"
KISA_dec = "ezYdB3TSFLXri3DA4wBTmOdhgS7mIzuIcnAb2e+ABCJM3L7vBWFnP/Yq7v34+q+xTvAnh0JqU3rYjb44kwOVqQ=="
```

urllib3 라이브러리는 Open API 요청 시 발생하는 오류 메세지를 무시하기 위해 사용했습니다.

1. NewSearch 함수

```
## 1. NewSearch 함수 생성
def NewSearch(query, output="json", display=100, start=1, sort="date"):

    # 요청 url 생성
    url = "https://openapi.naver.com/v1/search/"\
          "news.{}"\
          "?"\
          "query={}"\
          "&display={}"\
          "&start={}"\
          "&sort={}".format(output, query, str(display), str(start), sort) # sort를 sim으로 바꾸면 관련도순으로 검색이 가능

    # header 생성 - Open API Key
    headers = {'X-Naver-Client-Id': client_id,
               'X-Naver-Client-Secret' : clientKey}

    # GET Method API 요청
    res = requests.get(url, headers=headers,verify=False)

    return(res)
```

- query 검색 키워드를 입력하는 변수입니다.

- header에는 Naver에서 발급받은 Open API키를 입력합니다.
- url 요청 방식은 GET Method로 사용했습니다. XML방식을 사용해도 되지만… 제가 잘 몰라서 그냥 JSON으로 요청했습니다.
- 네이버 API의 경우에는 1번에 100개 목록(display arg)만 요청 가능하고, 대신에 시작하는 순번(start arg)을 선택 가능합니다.
  - 따라서 100개 이상의 목록을 요청할 때는 1번에서 100개를 요청하고, 101번부터 다시 100개를 요청하는 방법을 반복해야합니다.
  - 제가 사용권한을 받은 기능은 **검색** 기능으로 요청 횟수를 기준으로 count되며 일 25,000회까지 API 호출이 가능합니다.
- NewSearch 함수는 요청해서 받은 json 응답을 반환합니다.

2. JsonToExcel 함수

```
## 2. Json 형태의 데이터를 DataFrame으로 변경해주는 함수 생성
def JsonToExcel(inputdata):

    # JSON을 dictionary로 반환 후 필요 데이터(items 출력변수) 추출
    INFO = inputdata.json() # json을 딕셔너리 형태로 반환
    INFO = pd.DataFrame(INFO['items'])

    # 인용문으로 사용된 따옴표('&quot;' -> '"') 및 태그('<b>,</b>' -> '') 정제
    INFO.replace('&quot;','"',regex=True,inplace=True)
    INFO.replace('<b>','',regex=True,inplace=True)
    INFO.replace('</b>','',regex=True,inplace=True)

    # 이후 날짜를 인덱싱할 수 있도록 현지 시간대 정보 제거
    INFO['pubDate']=pd.to_datetime(INFO['pubDate']).dt.tz_localize(None)
    del INFO['link']

    return(INFO)
```

- JsonToExcel 함수는 NewSearch를 통해 받은 json 응답결과를 DataFrame형태로 바꾸어주는 함수입니다.
- json을 딕셔너리 형태로 반환한 후 DataFrame 형태로 변환합니다. 이중 저희가 원하는 정보인 'items'만을 추출해서 변환합니다.
- API 요청의 경우 인용구(큰따옴표, ")를 '&quot'로, 검색 키워드를 , 태그로 감싼 형태로 출력하므로 replace 함수를 이용해 이를 정제합니다.
- API 요청의 경우 뉴스기사가 네이버에 등록된 시간을 pubDate column으로 반환하는데, 이때 pubDate는 현지시간대 정보(localtime)를 포함합니다.
  - 이후 원하는 시간대의 기사만을 조회하기 위해서는 해당 현지시간대 표현 방법을 제거해주어야 합니다.
  - datetime.tz_localize 함수의 arg를 None으로 바꾸어주면 localtime 정보를 제거할 수 있습니다.

3. DomainSearch 함수

```
## 3. 도메인 요청시 도메인 등록자 명을 반환해주는 함수 생성
def DomainSearch(link,key_id,output="json"):

    # 요청 url 생성
    url = "http://apis.data.go.kr/B551505/whois/domain_name"\
          "?"\
          "serviceKey={}"\
          "&query={}"\
          "&answer={}".format(key_id, link, output)

    # GET Method API 요청
    try:
        res = requests.get(url, verify=False)
        res = res.json()
        Name = res['response']['whois']['krdomain']['regName']
    except:
        Name = str(link)

    return(Name)
```

- 해당 함수는 사실 큰 의미가 없는 함수입니다. Naver Open API의 경우 해당 기사의 original link를 출력 변수로 제공해주는데, 이 link의 도메인만을 조회해서 언론사 및 정보를 알 수 있을까해서 만든 정보였습니다.
- 한국인터넷진흥원(KISA)는 국내 KISA에서 관리하는 도메인의 경우 해당 도메인 정보를 제공하는 Open API를 제공하고 있습니다.

- 네이버 요청 시와 동일하게 json응답결과를 받아 정리하지만, 대다수 인터넷 신문의 경우에는 해외 도메인을 사용하고 있어 조회가 되지 않는 경우들이 많았습니다.

4. NewsAPI 함수

```
## 4. 기사 추출
def NewsAPI(keyword,start_date,end_date):

    result = pd.DataFrame() # output을 저장할 데이터프레임 생성

    # API 반복 시행 - 네이버 API의 경우 한번에 100개 데이터만 요청이 가능
    for i in range(6):
        response = NewSearch(keyword,start=i*100+1) # 뉴스기사 조회 함수
        output = JsonToExcel(response) # 응답JSON을 DataFrame으로 변환하는 함수
        result = pd.concat([result,output])

    # 중복된 기사 제거
    result = result.drop_duplicates(subset='originallink')
    result = result.drop_duplicates(subset='title')
    result = result.drop_duplicates(subset='description')

    # 시작일의 00시 00분 00초부터 종료일의 23시 59분 59초까지의 time index(boolean) 생성
    index = result['pubDate'].between(str(start_date)+' 00:00:00',str(end_date)+' 23:59:59')

    # 도메인 조회
    result = result[index] # time index에 해당하는 특정 일자 및 기간 데이터 조회
    result['domain']=result['originallink'] # 도메인을 추출하기 위한 새로운 열 추가
    result['domain'].replace('https://','', regex = True, inplace = True)
    result['domain'].replace('http://','', regex = True, inplace = True)
    result['domain'].replace('www.','', regex = True, inplace = True)
    result['name'] = result.domain.str.split('/').str[0]
    result.drop(['domain'], axis='columns', inplace=True)

    for x in result['name']:
        try:
            Name = DomainSearch(x,KISA_enc)
            if Name == '후이즈 도메인 관리자':
                result['name'].replace(x,str(x),inplace=True)
            else:
                result['name'].replace(x,Name,inplace=True)
        except:
            Name = str(x)
            result['name'].replace(x,Name,inplace=True)

    return(result)
```

- 뉴스를 검색하는 함수입니다. range를 6으로 설정해서 총 700개까지 뉴스를 조회한 결과값을 result에 저장합니다.
  - 이때 앞서 만들었던 NewSearch와 JsonToExcel 함수를 이용해서 DataFrame형태로 전환해서 저장합니다.
- 이후 drop_duplicates함수를 이용해서 중복된 기사를 제거합니다.
  - originallink 변수만으로도 대부분의 기사가 정리가 될 것 같지만,
  - 발간된 기사가 연합뉴스에 동시에 발간되는 경우도 있어 title과 description도 이용해서 제거해줍니다.
- 검색 시작일자의 경우에는 해당일의 00시 00분 00초부터 종료일자의 경우 23시 59분 59초까지로 변경한 timeindex를 생성합니다.
  - 하루동안 발간된 기사만 조회할 때 올바른 timeindex를 적용하기 위해서이며
  - 네이버에서 반환해준 pubDate에서 아까 localtime 정보를 제공해준 데이터와 같은 형식으로 맞추어 인덱싱이 가능하게끔 만들어주기 위함입니다.
- 도메인 조회의 경우 ‘originallink’의 도메인만을 추출해서 KISA에 조회요청하는 단계입니다.
  - KISA의 응답결과가 오류이거나, 없는 경우에는 해당 도메인 링크를
  - 조회가 되는 경우에는 해당 도메인 회사명을 반환하게끔 작성했습니다.
  - 하지만, 개인사업자의 경우에는 개인의 이름이 반환되는 경우도 있고, 생각 외로 해외도메인이 많아서 잘 조회되지 않았습니다.
  - 그냥 없는 함수라고 생각해주세요ㅠ

5. Main 함수

```
def main():

    keyword = input("keyword : ")
    start = input("start date (YYYY-MM-DD) : ")
```

```
    end = input("end date (YYYY-MM-DD) : ")
    monitor = NewsAPI(keyword,start,end)
    now = datetime.now()
    # 파일 이름
    outputFileName = 'NEWS_API_%s_%s.%s.%s %sh.%sm%ss.xlsx' % (keyword, now.year, now.month, now.day, now.hour, now.minute, now.second
    # 해당 코드가 저장된 폴더에 위의 이름으로 엑셀파일 저장
    monitor.to_excel(outputFileName,sheet_name='sheet1')

    return(monitor)

main()
```

- Main 함수의 경우에는 기존 크롤링 함수에서 사용자가 콘솔창에 입력하면 자동으로 진행해주는 기능이 좋아서 복붙했습니다.
- 위 함수를 실행하면 콘솔창 안내에 따라 편리하게 뉴스기사 조회가 가능합니다.
- 현재는 출력문제로 spyder나 jupyer IDE에서 이용하는 경우 오류가 발생할 수 있습니다.
  - 가상환경에서 실행하면 input기능을 이용할 수 있습니다.
  - 가상환경 설정이 어려운 경우에는 다음의 형식에 맞춰 함수를 작성하고 코드를 실행하면 동일한 결과를 얻을 수 있습니다.

```
monitor = NewsAPI('소비 심리','2022-07-07','2022-07-08') # 키워드, 시작일, 종료일 순
now = datetime.now()
outputFileName = 'NEWS_API_%s_%s.%s.%s %sh.%sm%ss.xlsx' % (keyword, now.year, now.month, now.day, now.hour, now.minute, now.second
monitor.to_excel(outputFileName,sheet_name='sheet1')
```