Python 트위터 크롤링

키워드: 웹 크롤러 (https://ko.wikipedia.org/wiki/%EC%9B%B9_%ED%81%AC%EB%A1%A4%EB%9F%AC), Selenium (https://www.seleniumhq.org/)

이번 장에서는 강의에서 살펴본 트위터 크롤링 코드에 대해서 살펴보겠습니다. 트위터의 정보는 굉장히 방대하고, 다양한 카테고리의 정보가 축적되어 있습니다. 때문에 소셜미디어 채널 중에서도 특히 크롤링이 많이 진행되며, 오픈 소스 형태로 크롤링 코드가 많이 공개되어 있습니다. 인스타그램 크롤링과 전체적으로 중복되는 크롤링 코드 구문이 많으므로, 중복되지 않는 부분을 위주로 알아보도록 하겠습니다.

```
import time
from selenium import webdriver
from openpyxl import load_workbook # 엑셀 파일 처리를 위한 모듈
from konlpy.tag import Okt # 말뭉치 분석을 위한 모듈

URL = "https://twitter.com/search?q={}&src=typd"
DRIVER_DIR = '/Users/temp/Project_FC/chromedriver'
SAVE_DIR = 'test.xlsx' # 엑셀 파일 경로

def twitter_scrap(keyword):
    try:
        driver = webdriver.Chrome(DRIVER_DIR)
        driver.implicitly_wait(10) # 암묵적으로 웹 자원을 (최대) 10초 기다리기
        driver.get(URL.format(str(keyword)))

        no_page = 0
        while no_page < 10:
            driver.execute_script('window.scrollTo(0,
document.body.scrollHeight)')
            no_page += 1
            time.sleep(1.5)
        # 모든 트윗 리스트
        content = driver.find_elements_by_css_selector('div.content')
        print("content-length: ", len(content))

        result = []
        # 반복문으로 각각의 트윗 뽑아오기
        for i in content:
            # 트윗 텍스트
            cont = i.find_element_by_css_selector('p.tweet-text')
            # 트윗 업로드 시간
            timestamp = i.find_element_by_css_selector('a.tweet-timestamp')
            result.append([cont.text.strip(),
timestamp.get_attribute("title")])
            # result = [[텍스트, 시간], [텍스트, 시간], [텍스트, 시간], ...]
            print(cont.text.strip(), timestamp.get_attribute("title"))
            print('----------------------------------------')
        save_excel(result)
    except Exception as e:
        print(e)
```

```
    finally:
        driver.quit()

# 엑셀 파일에 데이터를 저장할 함수 선언
def save_excel(result):
    try:
        wb = load_workbook(SAVE_DIR) # 엑셀 파일을 불러오고, 워크북 객체 선언
        ws = wb.create_sheet(title='twitter') # 시트의 타이틀 할당 및 워크시트 객체 선언
        for idx, re in enumerate(result): # 데이터를 담고 있는 리스트의 크기만큼 반복
(idx: 인덱스(0부터 시작), re: 실제 값(리스트)
            # 주의 하실 점은, 엑셀 파일의 Cell은 1부터 시작하기 때문에 인덱스에 1을 더해줍니다.
            ws['A' + str(idx + 1)] = re[0] # A열에 트윗 텍스트를 저장합니다.
            ws['B' + str(idx + 1)] = re[1] # B열에 트윗 시간을 저장합니다.
        wb.save(SAVE_DIR) # 엑셀 파일을 저장합니다.
    finally:
        wb.close() # 엑셀 파일을 닫습니다.

# 엑셀 파일에 저장된 값을 불러올 함수 선언
def get_excel():
    result = [] # 불러온 엑셀 파일 정보를 담아서 반환할 리스트 선언
    try:
        wb = load_workbook(SAVE_DIR, read_only=True) # 읽기 전용으로 워크북 객체 선언
        ws = wb['twitter'] # 시트 선택
        for row in ws.rows: # 모든 행 반환
            result.append(row[0].value) # 0(A열의 정보인 트윗 텍스트 값만 반환하여 리스트에
할당합니다.)
    finally:
        wb.close() # 워크북 객체를 닫습니다.
    return result # 결과를 반환합니다.

# 말뭉치를 분석합니다.
def get_content(result):
    ok = Okt() # Okt 객체 선언
    content = {} # 단어의 빈도수를 계산할 딕셔너리 선언
    for re in result:
        temp = ok.pos(re) # 값 -> ('단어', '품사'), ('단어', '품사'), ('단어', '품사'),
...
        for t in temp: # 찾은 단어의 크기만큼 반복합니다.
            if t[1] == 'Hashtag': # 해시태그?
                if not (t[0] in content): # 이미 결과 값이 초기화 되어있는지?
                    content[t[0]] = 0 # 없다면 초기화
                content[t[0]] += 1 # 1을 더하는 연산
    content = sorted(content.items(), key = lambda x:x[1], reverse = True) #
단어의 빈도수를 기준으로 내림차순으로 정렬합니다.

    for k, v in content:
        print("({}){}".format(k, v), end = ' ') # 키와 값을 출력합니다.

if __name__ == "__main__":
    keyword = input('keyword?')
    twitter_scrap(keyword)
    get_content(get_excel()) # 저장된 엑셀 파일을 불러오고, 빈도수 분석
```

위의 코드는 twitter 함수를 통해 크롤링할 데이터를 엑셀 파일로 저장할 수 있도록 하고, 저장된 데이터를 불러와 해시태그의 빈도수를 계산하는 프로그램 코드입니다. 다양한 정보를 담고 있는 트위터 게시물의 정보를 크롤링하고 분석하면, 굉장히 재밌는 결과를 많이 만들어낼 수 있습니다. 트위터 크롤링 코드를 많이 작성해보세요. :)