Python Programming and Practice

Development daily news keyword summary program

Progress Report : 1

Date : 2023.11.26
Name : 지연우
ID : 224491

1. Introduction

1) Background

It's time consuming and very cumbersome to check out all the tons of news pouring out every day. But we can't completely stop paying attention to current events, so we need a more efficient way to get information. To solve this problem, a program that can easily check current events information in short sentences or keywords is needed.

2) Project goal

Aim to create a program that summarizes articles uploaded every day in short sentences or words.

3) Differences from existing programs

The article summary function previously provided by Naver News only summarizes each article. This, of course, helps save time, but it is difficult to easily find out various current events information. We analyze all articles to show the user a few key sentences or keywords, making it easy to get current events information.

2. Functional Requirement

1) Collect articles

- function to collect articles uploaded daily

(1) Collect information on popular articles by media company

- Crawl the titles, body text of the top five popular articles of each media company

2) Summarize the articles

- Summarize each article in short sentences and words.

(1) Create article summary

- Summarize the article in one sentence.

(2) Extract keywords

- Extract one keyword of articles.

3) Show keywords

- Show keywords to user.

(1) Sort keywords in order of frequency

- Show the keywords that appear in more articles first.

4) Show summary of articles by keyword

- When a user selects a keyword, it shows a summary of the article in which that keyword appears.

3. Progress

1) Function Implementation

(1) Collect articles

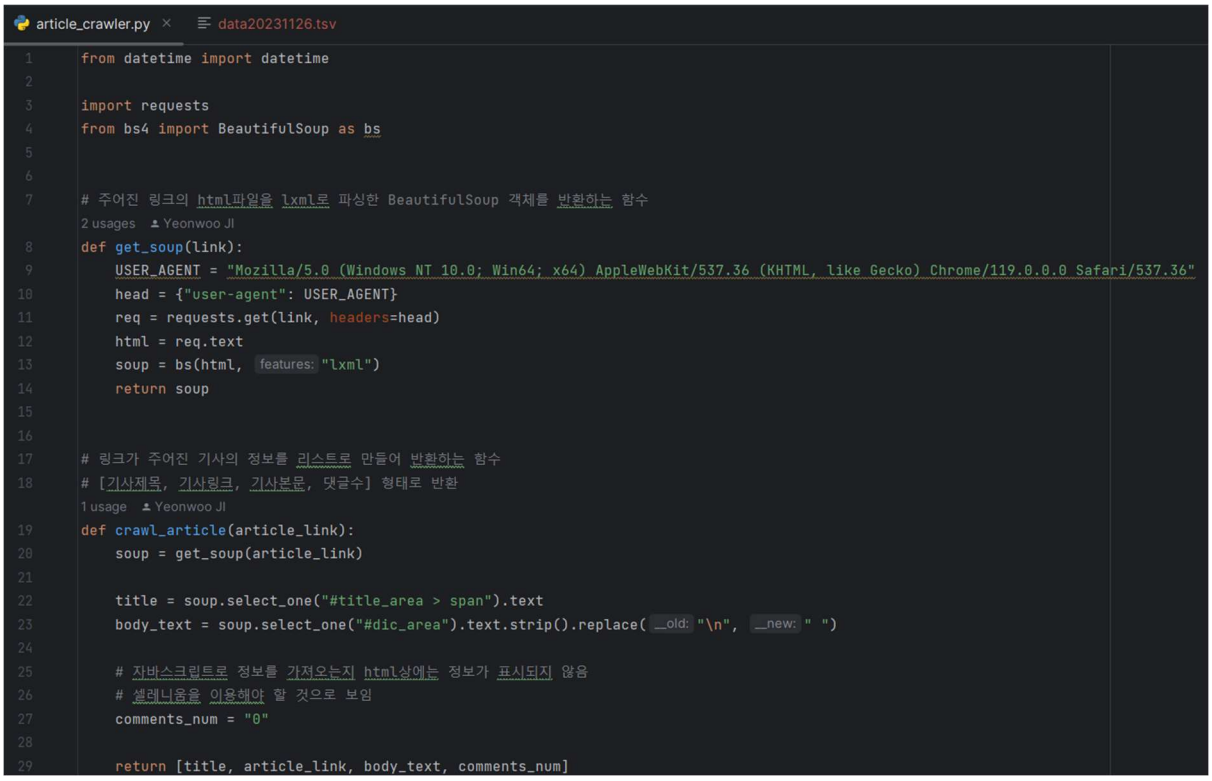
- Input: Link to Naver News Ranking page

- Output: TSV file containing information of the articles

- Description: Crawl the titles, body text and url of the top five popular articles from each media company.

- Applied concepts: loops, functions, string manipulation, exception handling, file output, __name__ == "__main__".

- Code Screenshot



article_crawler.py × data20231126.tsv

```python
from datetime import datetime

import requests
from bs4 import BeautifulSoup as bs


# 주어진 링크의 html파일을 lxml로 파싱한 BeautifulSoup 객체를 반환하는 함수
2 usages  Yeonwoo Jl
def get_soup(link):
    USER_AGENT = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36"
    head = {"user-agent": USER_AGENT}
    req = requests.get(link, headers=head)
    html = req.text
    soup = bs(html, features: "lxml")
    return soup


# 링크가 주어진 기사의 정보를 리스트로 만들어 반환하는 함수
# [기사제목, 기사링크, 기사본문, 댓글수] 형태로 반환
1 usage  Yeonwoo Jl
def crawl_article(article_link):
    soup = get_soup(article_link)

    title = soup.select_one("#title_area > span").text
    body_text = soup.select_one("#dic_area").text.strip().replace( _old: "\n",  _new: " ")

    # 자바스크립트로 정보를 가져오는지 html상에는 정보가 표시되지 않음
    # 셀레니움을 이용해야 할 것으로 보임
    comments_num = "0"

    return [title, article_link, body_text, comments_num]
```

article_crawler.py × data20231126.tsv

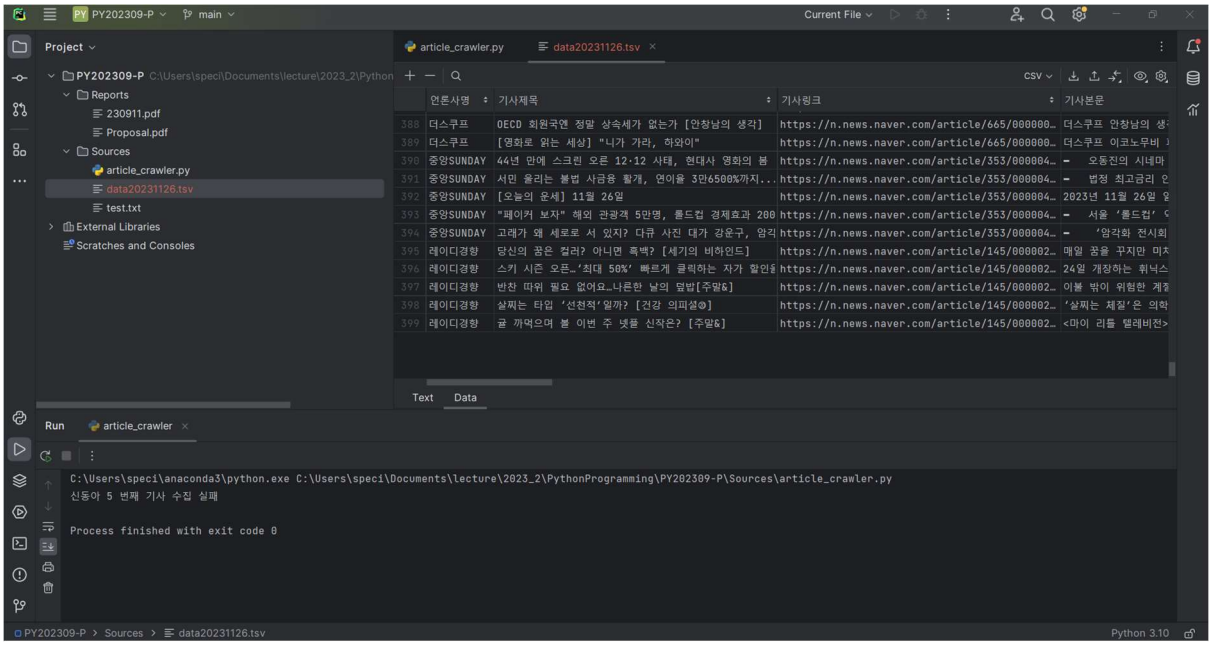
```python

# 네이버 뉴스의 랭킹뉴스 페이지에서 언론사별 조회수 상위 5개 기사의 정보를 수집하는 함수
# 각 기사에 대해 crawl_article 함수를 사용하여 기사 정보를 얻어오고, tsv파일에 저장
1 usage  Yeonwoo Jl
def crawl_rankings(file_name):
    rankings_link = "https://news.naver.com/main/ranking/popularDay.naver"
    soup = get_soup(rankings_link)
    rankingnews_boxes = soup.select(".rankingnews_box")  # 언론사별 랭킹 5위까지의 기사 정보가 담긴 박스들 선택

    with open(file_name, 'w', encoding="utf8") as fp:
        fp.write("언론사명\t기사제목\t기사링크\t기사본문\t댓글수\n")

        for box in rankingnews_boxes:
            company_name = box.select_one(".rankingnews_name").text
            articles = box.select("li")

            for i, article in enumerate(articles):
                try:
                    # 링크는 https://n.news.naver.com/article/언론사코드/기사번호?ntype=RANKGIN 형태
                    article_link = article.select_one(".list_title")["href"]
                    article_data = crawl_article(article_link)
                    fp.write(company_name + "\t" + "\t".join(article_data) + "\n")
                # 집계기준에 해당하는 기사가 없어 랭킹 5위까지 기사가 존재하지 않을 때
                except:
                    print(company_name, i+1, "번째 기사 수집 실패")


if __name__ == "__main__":
    file = f"./data{str(datetime.today().date()).replace( _old: '-',  _new: '')}.tsv"
    crawl_rankings(file)
```

2) Test Results

(1) Collect articles

- Description: Successfully crawled information from 399 articles across 80 media outlets on Naver News Ranking page. For 'Shindonga' (a specific media outlet), only articles up to the 4th rank were available, resulting in a failure message when attempting to collect the 5th article.

- Test Results Screenshot:



4. Changes in Comparison to the Plan

1) Removal of Comment Count-related Feature

- Previously: Comment count was collected in Function 1, and when showing article summaries in Function 4, they were displayed in descending order of comment count.

- Now: Comment count is no longer collected in Function 1. In Function 4, article summaries are displayed randomly.

- Reason: Comment count information was not present in the crawled HTML documents. Although considering using Selenium to collect comment counts was an option, it was decided to exclude this feature as it was not deemed crucial, and the time required for article collection became excessively long.

5. Schedule

- Indicating Progress

업무	11/3	11/17	12/1	12/15	12/22
Write a proposal	Complete				
Function 1		Complete			
Function 2		In progress			
Function 3				Not Started	
Function 4				Not Started	
Final Report					Not Started