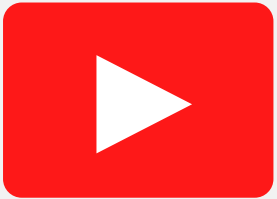


YOUTUBE 댓글 수집 시각화

Selenium, Konlpy, WebDriver_Manager, Pandas, BeautifulSoup4,



대우직업능력개발원 4팀 **TEAM** 임대



TEAM 임대

개발기간 (2023.02.20~2023.02.28)

대우직업능력개발원 4팀 TEAM 임대



임서인 (Seoin Lim)

- PM (프로젝트 기획 및 총괄)
- 문서 작성, 라이브러리 리서치
- 크롤링 기능 개발



황대명 (Daemyoeng Hwang)

- 라이브러리 리서치
- 크롤링 기능 개발
- 시각화 개발



목차



1. 프로젝트 소개
2. 구글 트렌드, 유튜브를 선택한 이유
3. 프로세스 흐름도
4. 사용한 라이브러리
5. 프로젝트 주요 기능
6. 테스트 결과
7. 프로젝트 기대효과
8. 후기



[프로젝트 자동화로 간편하게 가능]

comm_collect.exe

visualize.exe

1. exe파일로 실행

인기 검색어 주제 탐색 일별 인기 급상승 검색어

2023년 3월 2일 목요일

- 1 현대자동차 채용
현대자동차, 기술직 신입사원 채용 현대자동차그룹 · 17시간 전
- 2 근로장려금
2022년 귀속 하반기분 근로장려금 15일까지 신청해야 연합
- 3 올리브영
Z세대가 픽한 '코링코 톡톡하라 속눈썹', 이제 올리브영에서

2. 구글 트렌드에서 인기 검색어 1위 추출

YouTube KR 현대자동차 채용 -Shorts 🔍 🎤

관련 동영상

현대차 기술직 자소서! 2023년! 현대차공자 기술직

2023년 현대자동차 '모빌리티 기술인력' '모빌리티 기술인력' 강점? 샘플 공개! 2:05

2023년 현대자동차 '모빌리티 기술인력' 기술직(생산직) 자기소개서 강점 작성법 조회수 2.1천회 · 10시간 전

SKY TEAM Finding I

2023년 현대자동차 '모빌리티 기술인력' 기술직(생산직) 자기소개서 강점 작성법 실제 샘플 전격 공개 새 동영상

'현대차 생산직' 채용에 홈페이지도 액 연봉' 진실은? / 이포커스 조회수 672회 · 4시간 전

이포커스 뉴스

3. 유튜브에 키워드 검색



댓글 817개 정렬 기준

댓글 추가...

YOON IK 1일 전
회사는 다르지만 완성차 업계 14년 경력임.
좋은 대학 못나오고 어릴때 중소기업 갈때는 대기업 생산직이
좋아요.

46 답글

↓ 답글 2개

대한민국 5일 전
시대를 잘타고 나아 된다. 옛날에는 공돌이라고 무시받던 현대차
군인, 버스기사 등도 옛날에는 천대받았지만 지금은 괜찮은 직업이
아니냐?

197 답글

↓ 답글 66개

이창원 1일 전
7-8년전하고 지금하고 공무원 월급은 거의 비슷한데 대기업들만
되버렸음. 7-8년전은 소득이 공무원vs대기업 이정도 차이는 아니
였어?

19 답글

↓ 답글 4개

호갑 5일 전
초봉만 5천에 휴대폰보면서 일할수있다는 전설의 현대차..

140 답글

↓ 답글 28개

니 17시간 전
별봉 연봉 워라벨 현대자동차 압승 주변
요즘 악성민원도 너무 많고 민원인한테
래서 정보장이라도 들어가서 정년때까지 버티는게 낫고 쉽
지않아

4 답글



4. csv파일로 데이터 저장

3. 상단 3개의 영상의 댓글 추출
(이후로는 영상 정보만 추출)

5. visualize.exe로
시각화 결과 추출



는 매일 갱신되는 검색어 트렌드 데이터를 제공하며,
이를 통해 사용자들의 관심사나 선호도, 트렌드 등을 파악할 수 있습니다.

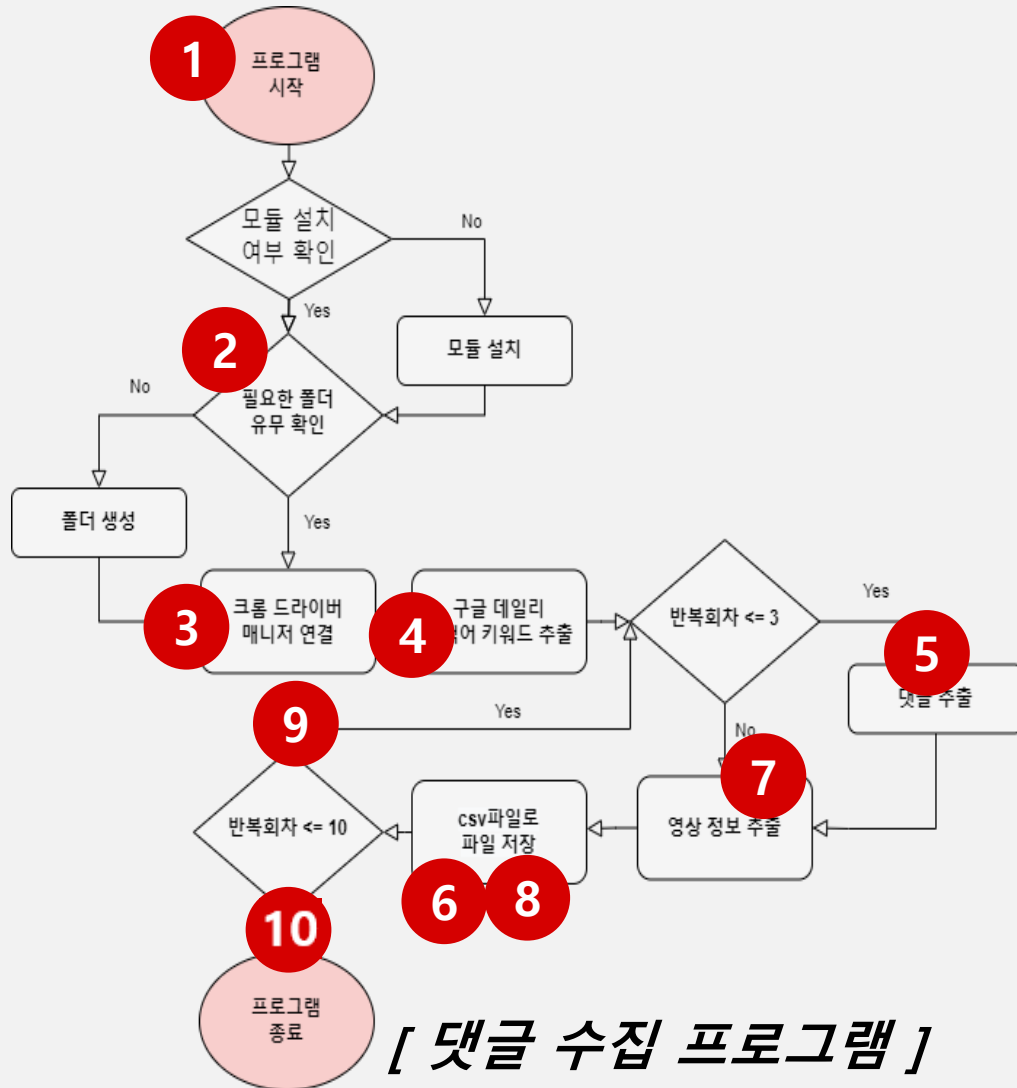
+



는 세계적으로 인기 있는 동영상 공유 플랫폼 중 하나이며,
유저들이 업로드한 수많은 동영상과 댓글이 있습니다. 이러한 동영상과 댓글은 다양한 분야의
정보와 의견을 담고 있으며 빅데이터 분야에서 중요한 자료로 인식되고 있습니다.

=

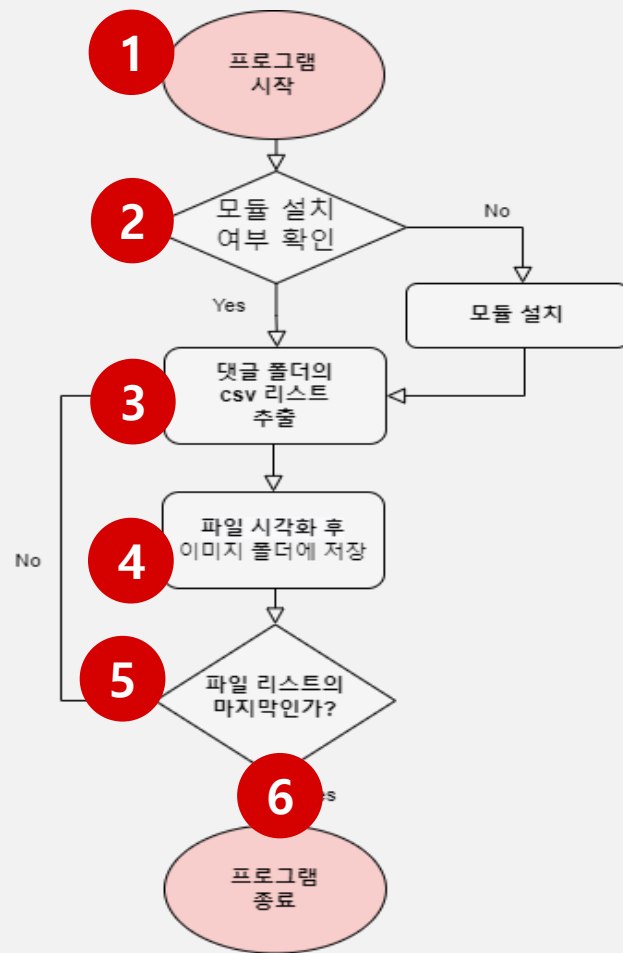
유튜브와 같은 온라인 미디어에서는 사용자들의 관심사와 선호도가 빠르게 변화할 수
있으므로, 실시간으로 업데이트되는 **구글 일간 트렌드** 검색어를 활용하여, **핫한 키워드나**
트렌드를 파악하는 것이 중요합니다.



[댓글 수집 프로그램]

프로세스 흐름도 요약

1. 댓글 수집 프로그램 시작
2. 라이브러리 설치, 필수 폴더 유무 확인
3. 크롬 드라이버 매니저 연결
4. 구글 데일리 검색어 키워드 추출
5. 영상 정보 및 댓글 수집, 정제
6. 추출된 영상정보와 댓글 저장
7. 영상 정보 수집
8. 수집된 영상정보 저장
9. 댓글 수집 프로그램 회차 반복
10. 댓글 수집 프로그램 종료



[시각화 프로그램]

프로세스 흐름도 요약

1. 시각화 프로그램 시작
2. 라이브러리 설치,
필수 폴더 유무 확인
3. 댓글 폴더의 csv리스트 추출
4. 파일 시각화 후 이미지 폴더에 저장
5. 시각화 프로그램 반복
6. 시각화 프로그램 종료



Selenium

*포터블 프레임워크

동적으로 생성되는 사이트의 데이터를
크롤링 할 때 매우 유용하게 사용

Konlpy

라이브러리

한국어 정보처리를 위한
파이썬 패키지

Webdriver_Manager

라이브러리

크롬 드라이버를 크롬 브라우저
버전에 맞게 자동으로 다운로드

Pandas

라이브러리

쉽고 직관적인 분류된 데이터로
작업할 수 있도록 데이터 구조를 제공

bs4

라이브러리

HTML정보로 부터 데이터를
가져오기 쉽게, 데이터 별로 나누어 줌

Wordcloud

라이브러리

텍스트데이터를 가지고
워드클라우드를 생성하기 쉽게 도와줌



```
def checker_module():
    import sys
    import subprocess
    import os

    try:
        # 없는 모듈 import시 에러 발생
        import selenium
    except:
        print("selenium 모듈을 설치합니다.")
        subprocess.check_call([sys.executable, '-m', 'pip',
                                'install', '--upgrade', 'pip'])
        subprocess.check_call([sys.executable, '-m', 'pip',
                                'install', '--upgrade', 'selenium'])

    try:
        import webdriver_manager
    except:
        print("webdriver-manager 모듈을 설치합니다.")
        subprocess.check_call([sys.executable, '-m', 'pip',
                                'install', '--upgrade', 'pip'])
        subprocess.check_call([sys.executable, '-m', 'pip',
                                'install', '--upgrade', 'webdriver-manager'])

    try:
        import pandas
    except:
        print("pandas 모듈을 설치합니다.")
        subprocess.check_call([sys.executable, '-m', 'pip',
                                'install', '--upgrade', 'pip'])
        subprocess.check_call([sys.executable, '-m', 'pip',
                                'install', '--upgrade', 'pandas'])
```

라이브러리 설치 유무 확인 후
설치 하는 함수

```
try:
    import bs4
except:
    print("bs4 모듈을 설치합니다.")
    subprocess.check_call([sys.executable, '-m', 'pip',
                            'install', '--upgrade', 'pip'])
    subprocess.check_call([sys.executable, '-m', 'pip',
                            'install', '--upgrade', 'bs4'])

try:
    import konlpy
except:
    print("konlpy 모듈을 설치합니다.")
    subprocess.check_call([sys.executable, '-m', 'pip',
                            'install', '--upgrade', 'pip'])
    subprocess.check_call([sys.executable, '-m', 'pip',
                            'install', '--upgrade', 'konlpy'])

# wordcloud 설치 파일 이름 찾아서 저장
file_name = [file for file in os.listdir() if file.endswith('.whl')][0]

try:
    import wordcloud
except:
    print("WordCloud 모듈을 설치합니다.")
    subprocess.check_call([sys.executable, '-m', 'pip',
                            'install', '--upgrade', 'pip'])
    subprocess.check_call([sys.executable, '-m', 'pip',
                            'install', '--upgrade', file_name])

try:
    import matplotlib
except:
    print("matplotlib 모듈을 설치합니다.")
    subprocess.check_call([sys.executable, '-m', 'pip',
                            'install', '--upgrade', 'pip'])
    subprocess.check_call([sys.executable, '-m', 'pip',
                            'install', '--upgrade', 'matplotlib'])

return True
```



```
# 필수 저장 폴더 생성 유무 체크
def checker_required_folder():
    import os
    # comment폴더 유무 확인후 생성
    if not os.path.exists('./comment'):
        os.mkdir('./comment')

    # nocomment폴더 유무 확인후 생성
    if not os.path.exists('./nocomment'):
        os.mkdir('./nocomment')

    # img폴더 유무 확인후 생성
    if not os.path.exists('./img'):
        os.mkdir('./img')
```

필수 디렉토리 유무 확인 후,
디렉토리가 없다면 생성하는 함수

```
# 파일 위치를 담고있는 변수
path = os.getcwd()+'\comment'

# .csv 파일 만 추출하여 리스트에 저장
file_names = [file for file in os.listdir(path) if file.endswith('.csv')]

for name in file_names:
    # 데이터 프레임 생성
    df = pd.read_csv(path + '\\ ' + name, names=['word', 'count'], skiprows=[1,2,3,4,5])

    # 생성된 데이터 프레임을 딕셔너리 형태로 변환
    wc = df.set_index("word").to_dict()["count"]

    wordCloud = WordCloud(
        font_path = "malgun", # 폰트 지정
        width = 400, # 워드 클라우드의 너비 지정
        height = 400, # 워드클라우드의 높이 지정
        max_font_size=100, # 가장 빈도수가 높은 단어의 폰트 사이즈 지정
        background_color = 'white' # 배경색 지정
    ).generate_from_frequencies(wc) # 워드 클라우드 빈도수 지정

    plt.figure() # figure 생성
    plt.imshow(wordCloud) # 터미널에 이미지 보여주기
    plt.axis('off') # axis 끄기

    save_name = name.rstrip('.csv') + '.png'
    # 이미지 저장 경로 및 파일 이름 설정
    save_file = os.path.join( os.getcwd()+'\img', save_name)
    plt.savefig(save_file) # img 폴더에 이미지 저장
```




```
# 필수 저장 폴더 생성 유무 체크
def checker_required_folder():
    import os
    # comment폴더 유무 확인후 생성
    if not os.path.exists('./comment'):
        os.mkdir('./comment')

    # nocomment폴더 유무 확인후 생성
    if not os.path.exists('./nocomment'):
        os.mkdir('./nocomment')

    # img폴더 유무 확인후 생성
    if not os.path.exists('./img'):
        os.mkdir('./img')
```

 2023.02.28 - 02.45.28_국민연금공단

 2023.02.28 - 02.48.55_국민연금공단

 2023.02.28 - 02.49.06_국민연금공단

[csv 파일 내부]

| | | | |
|----|-------|-----------------------|--|
| 2 | 수집 날짜 | 2023.02.28 - 03.11.41 | |
| 3 | 제목 | 대표팀 새로운 감독 확정.. | |
| 4 | 채널 | 스포츠타임 | |
| 5 | 조회수 | 239,962회 | |
| 6 | 게시일 | 2023. 2. 27. | |
| 7 | 하다 | 608 | |
| 8 | 감독 | 411 | |
| 9 | 클린스만 | 205 | |
| 10 | 축구 | 196 | |
| 11 | 없다 | 148 | |
| 12 | 보다 | 147 | |
| 13 | 베다 | 146 | |
| 14 | 되다 | 141 | |
| 15 | 한국 | 128 | |
| 16 | 으로 | 122 | |
| 17 | 있다 | 113 | |

 comment

 img

 nocomment

[comment 디렉토리]

- 영상의 정보와 가공된 댓글을
저장한 csv 파일

[comment 디렉토리 내부]



```
# 필수 저장 폴더 생성 유무 체크
def checker_required_folder():
    import os
    # comment폴더 유무 확인후 생성
    if not os.path.exists('./comment'):
        os.mkdir('./comment')

    # nocomment폴더 유무 확인후 생성
    if not os.path.exists('./nocomment'):
        os.mkdir('./nocomment')

    # img폴더 유무 확인후 생성
    if not os.path.exists('./img'):
        os.mkdir('./img')
```

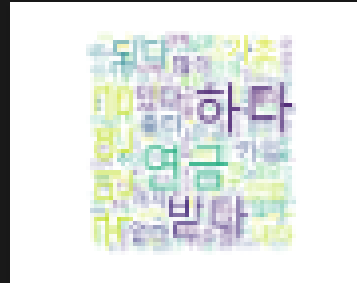
 comment

 img

 nocomm

[img 디렉토리]

- comment 디렉토리의 csv파일을
시각화한 이미지 파일



2023.02.28 -
02.45.28_국민연
금공단



2023.02.28 -
02.48.55_국민연
금공단



2023.02.28 -
02.49.06_국민연
금공단

[img 디렉토리 내부]



```
# 필수 저장 폴더 생성 유무 체크
def checker_required_folder():
    import os
    # comment폴더 유무 확인후 생성
    if not os.path.exists('./comment'):
        os.mkdir('./comment')

    # nocomment폴더 유무 확인후 생성
    if not os.path.exists('./nocomment'):
        os.mkdir('./nocomment')

    # img폴더 유무 확인후 생성
    if not os.path.exists('./img'):
        os.mkdir('./img')
```

[nocomment 디렉토리]

- 댓글을 수집하지 않은,
영상의 정보만 수집한 csv파일

```
2023.02.28 - 02.49.17_국민연금공단
2023.02.28 - 02.49.18_국민연금공단
2023.02.28 - 02.49.20_국민연금공단
2023.02.28 - 02.49.22_국민연금공단
2023.02.28 - 02.49.23_국민연금공단
2023.02.28 - 02.49.25_국민연금공단
2023.02.28 - 02.49.27_국민연금공단
```

[nocomment 디렉토리 내부]

[csv 파일 내부]

| | | | | |
|-------|--------------------------------|--|--|--|
| 수집 날짜 | 2023.02.28 - 02.49.17 | | | |
| 제목 | 국민연금 60세 이후 계속 납부할까말까?(임의계속가입) | | | |
| 채널 | 연금부자연구소 | | | |
| 조회수 | 584,952회 | | | |
| 게시일 | 2022. 10. 20. | | | |

| | | | | |
|-------|---------------------------------|--|--|--|
| 수집 날짜 | 2023.02.28 - 02.49.18 | | | |
| 제목 | 저소득 지역가입자 국민연금 보험료 지원제도를 소개합니다. | | | |
| 채널 | 국민연금TV | | | |
| 조회수 | 1,183회 | | | |
| 게시일 | 2022. 12. 9. | | | |

| | | | | |
|-------|----------------------------------|--|--|--|
| 수집 날짜 | 2023.02.28 - 02.49.20 | | | |
| 제목 | [대학생 홍보대사_숏무비] 우리 국민연금 못 받는다면데?! | | | |
| 채널 | 국민연금TV | | | |
| 조회수 | 2,496회 | | | |
| 게시일 | 2022. 7. 15. | | | |

| | | | | |
|-------|-----------------------------------|--|--|--|
| 수집 날짜 | 2023.02.28 - 02.49.22 | | | |
| 제목 | 내 곁에 국민연금 어플이 있다면? 모바일에서 연금청구하세요! | | | |
| 채널 | 국민연금TV | | | |
| 조회수 | 61,635회 | | | |
| 게시일 | 2021. 12. 17. | | | |



```
import func_list
import time

# 모듈 설치 여부 체크
if (func_list.checker_module()):
    from selenium import webdriver
    from selenium.webdriver.common.keys import Keys
    from selenium.webdriver.common.by import By
    from selenium.webdriver.chrome.service import Service
    from selenium.webdriver.chrome.options import Options

    from webdriver_manager.chrome import ChromeDriverManager

# 필수 폴더 생성 여부 체크
func_list.checker_required_folder()
```

모듈 설치 여부 체크 후
main()실행

```
try:
    # 접속 url
    keywordurl = 'https://trends.google.co.kr/trends/trendingsearches/daily?geo=KR&hl=ko'
    url = "https://youtube.com/"

    # 크롬 드라이버 매니저 연결
    driver = webdriver.Chrome(service=Service(
        ChromeDriverManager().install()), options=Options())
    # 윈도우 전체화면으로 실행
    driver.maximize_window()

    # 구글 데일리 인기 검색어 1위 키워드 추출 후 키워드로 입력
    driver.get(keywordurl)
    driver.implicitly_wait(10)
    keyword = driver.find_element(
        By.XPATH, '/html/body/div[3]/div[2]/div/div[2]/div/div[1]/ng-include/div/div/div/div/md-list[1]/feed-item/ng-include/div/div/div/div/a').text

    # 유튜브 접속
    driver.get(url)
    driver.implicitly_wait(10)
    time.sleep(0.5)

    # 키워드 입력하여 검색
    search = driver.find_element(By.NAME, "search_query")
    search.send_keys(keyword + ' -Shorts')
    driver.implicitly_wait(10)
    search.send_keys(Keys.ENTER)
    driver.implicitly_wait(10)

    # 10개의 영상 정보를 추출
    for i in range(1, 11):
        func_list.get_comment(driver, i, keyword)

    # 드라이브 완전 종료
    print('프로그램을 종료합니다.')
    driver.quit()

except Exception as e:
    print(e)
```

webdriver_manager를 통해
크롬 드라이버 버전 맞춤 자동
다운로드

keywordurl에서 가져온 keyword를
url의 검색창에서 검색



댓글을 수집하는 함수

```
# 댓글 추출(웹 드라이버, 인덱스, 검색 키워드)
def get_comment(driver, index, keyword):
    from selenium.webdriver.common.by import By
    from bs4 import BeautifulSoup
    import time

    comment_dict = {}
    # index번째 영상의 xpath값 저장
    xpath = '''/html/body/ytd-app/div[1]/ytd-page-manager/ytd-search/div[1]/ytd-two-column-search-results-renderer/
    div/ytd-section-list-renderer/div[2]/ytd-item-section-renderer/div[3]/ytd-video-renderer[{}]/div[1]'''.format(index)
    # xpath 요소 찾기
    contents = driver.find_element(By.XPATH, xpath)
    driver.implicitly_wait(10)

    contents.click()
    driver.implicitly_wait(10)

    # 영상 일시정지
    driver.find_element(
        By.XPATH, '''/html/body/ytd-app/div[1]/ytd-page-manager/ytd-watch-flexy/div[5]/
        div[1]/div/div[1]/div[2]/div/div/ytd-player/div/div/div[1]/video''').click()
    driver.implicitly_wait(10)

    # 영상 설명 더보기 클릭
    driver.find_element(
        By.XPATH, '''/html/body/ytd-app/div[1]/ytd-page-manager/ytd-watch-flexy/div[5]/
        div[1]/div/div[2]/ytd-watch-metadata/div/div[3]/div[1]''').click()
    driver.implicitly_wait(10)

    video_info = info_collect(driver) # 영상 정보를 json형태로 리턴 받음

    video_info['검색어'] = keyword
```

```
for key, value in video_info.items():
    print(key + " : " + value)

# index 3이하의 영상만 댓글 추출
if index <= 3:
    print("스크롤 시작 : " + time.strftime('%H:%M:%S'))
    scroll_down(driver)

    print("대댓글 열기 시작 : " + time.strftime('%H:%M:%S'))
    open_reply(driver)

    print("댓글 추출 시작 : " + time.strftime('%H:%M:%S'))

    # html source 불러와서 저장
    html_source = driver.page_source
    # bs4를 이용한 html parsing
    soup = BeautifulSoup(html_source, 'html.parser')
    # 댓글의 text 추출
    comment_list = soup.select("yt-formatted-string#content-text")

    # 댓글의 형태소 분석
    item = mrphl_anlys(comment_list)

    # 리스트를 딕셔너리 형태로 변환
    comment_dict = data_prfct(item)

# 영상정보를 csv로 저장
save_data(comment_dict, video_info, index)
print("")
print(comment_dict)

driver.back()
driver.implicitly_wait(10)

driver.maximize_window()
driver.implicitly_wait(10)
```




```
def scroll_down(driver):
    import time

    # 화면 크기 조정 (유튜브 댓글 추출 안정성 증가)
    driver.set_window_size(800, 1100)

    # scrollHeight = 화면 바깥으로 빠져나간 부분까지 포함한 전체 길이
    # 0부터 전체길이(맨 아래)까지 스크롤한다.
    driver.execute_script(
        "window.scrollTo(0, document.documentElement.scrollHeight)")
    time.sleep(1.5)

    # 스크롤 이전 높이
    last_height = driver.execute_script(
        "return document.documentElement.scrollHeight")

    while True:
        # 스크롤의 y좌표를 가장아래(scrollHeight)까지 내림
        driver.execute_script(
            "window.scrollTo(0, document.documentElement.scrollHeight);")
        time.sleep(1.5)

        # 스크롤 후 높이 구하기
        new_height = driver.execute_script(
            "return document.documentElement.scrollHeight")
        # 끝까지 스크롤 한 뒤 멈추기
        if new_height == last_height:
            break
        last_height = new_height

        time.sleep(1.5)

    return 0
```

```
if index <= 3:
    print("스크롤 시작 :" + time.strftime('%H.%M.%S'))
    scroll_down(driver)
```

get_comment 함수에서 사용됨

댓글 수집을 위해

자동으로 최하단으로 스크롤하는 함수



```
# 형태소 분석 (댓글 배열)
def mrphl_anlys(arr):
    from konlpy.tag import Okt
    import re

    # 한국어 형태소 분석
    okt = Okt()
    str_list = []

    for text in arr:
        temp_comment = text.text

        temp_comment = temp_comment.replace('\n', ' ')
        temp_comment = temp_comment.replace('\r', ' ')
        temp_comment = temp_comment.replace('\t', ' ').split(" ") # 단어로 쪼갬

        # filter함수를 이용하여 '@', 'https'를 포함한 단어 제거
        temp_comment = list(filter(lambda n: n.find('@') != 0, temp_comment))
        temp_comment = list(
            filter(lambda n: n.find('https') != 0, temp_comment))

        # re.sub() 정규표현식을 통해 문자열을 치환하는 함수
        # 한글, 숫자, 영어, 일본어, 한자를 제외한 댓글 정리
        temp_comment = list(map(lambda n: re.sub(
            r"[^\uAC00-\uD7A30-9a-zA-Zぁ-うァ-ヴー々ヅメー-籲]", "", n).replace(u'\xa0', u' '), temp_comment))

        # 1차원 배열을 문자열로 정렬
        temp_comment = " ".join(temp_comment)
        # okt객체를 이용한 형태소 분석
        temp_comment = okt.morphs(temp_comment, stem=True)
        # '' 빈 요소 값 제거
        temp_comment = list(filter(None, temp_comment))

    str_list.append(temp_comment)
```

```
print("댓글 추출 시작 : " + time.strftime('%H.%M.%S'))
html_source = driver.page_source
soup = BeautifulSoup(html_source, 'html.parser')
comment_list = soup.select("yt-formatted-string#content-text")

item = mrphlAnlys(comment_list)
```

get_comment 함수에서 사용됨

수집된 댓글의 데이터 처리를 위해
형태소를 분석하는 함수

6. 테스트 결과



제목 : 수면마취제 프로포폴에 중독되는 이...

조회수 : 92,206회

수집 날짜 : 2023.02.28 - 08.18.15



제목 : [자막뉴스] 20대 여성 "프로포폴 투..

조회수 : 14,947회

수집 날짜 : 2023.02.28 - 08.22.33



제목 : "유아인, 2021년에만 프로포폴 73회

조회수 : 268,228회

수집 날짜 : 2023.02.28 - 08.23.47

[Index 4]

제목 : [단독]프로포폴 중독돼서...하루

조회수 : 653,706회

수집 날짜 : 2023.02.28 - 09.08.58

[Index 5]

제목 : 마취에 대한 모든 궁금증/연예인,

조회수 : 27,410회

수집 날짜 : 2023.02.28 - 09.08.59

[Index 6]

제목 : 이영돈PD 프로포폴 주사를 직접

조회수 : 457,782회

수집 날짜 : 2023.02.28 - 09.09.00

[Index 7]

제목 : [#알쓸범잡2] (1시간) 한국은 더

조회수 : 785,576회

수집 날짜 : 2023.02.28 - 09.09.01

[Index 8]

제목 : 제2의 프로포폴 '에토미데이트'

조회수 : 1,002,878회

수집 날짜 : 2023.02.28 - 09.09.03

[Index 9]

제목 : 전지현, 미모 비결 "프로포폴 매

조회수 : 672,703회

수집 날짜 : 2023.02.28 - 09.09.04

[Index 10]

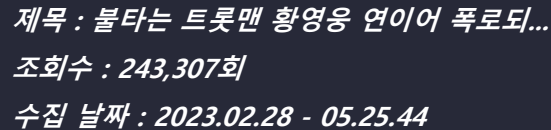
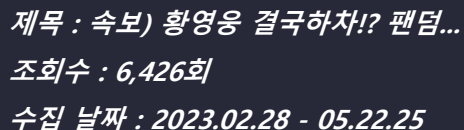
제목 : [자막뉴스] 20대 여성 "프로포폴

조회수 : 14,949회

수집 날짜 : 2023.02.28 - 09.09.05



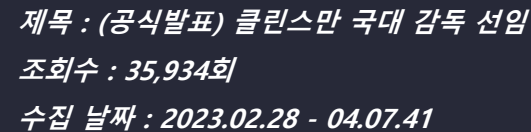
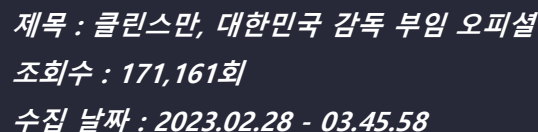
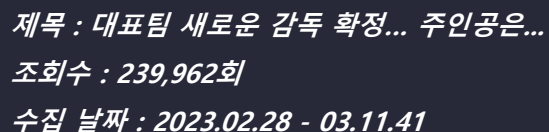
검색 키워드 : 프로포폴 (2023-02-09일 구글 트렌드 인기 검색어)



수집 날짜 : 2023.02.28 - 06.19.00



검색 키워드 : 황영웅 (2023-02-24일 구글 트렌드 인기 검색어)



[Index 7]

제목 : [독점] 브버지, 클린스만 한국 ...

조회수 : 49,093회

수집 날짜 : 2023.02.28 - 04.14.58





1. 대중의 관심사 파악

- 인기 있는 키워드 수집 후 검색된 동영상의 댓글 및 정보 분석
- 대중의 수요와 선호도 파악하여 제품, 서비스 개발 및 마케팅 전략 수립 가능

2. 제품/서비스 개선 방향성 제시

- 제품/서비스에 대한 고객의 만족도와 불만족 요소 파악
- 제품/서비스 개선 방향성 제시하여 고객 만족도 높이기

3. 경쟁사 분석

- 경쟁사 제품/서비스에 대한 고객의 반응과 불만족 요소 파악하여 경쟁사의 강점과 약점 파악
- 기업이나 단체의 경쟁력 강화 가능

4. 소셜미디어 마케팅에 활용

- 제품/서비스와 관련된 이슈 파악
- 소셜미디어에서 홍보 및 마케팅에 활용하여 더 많은 고객들에게 제품/서비스 알리고 홍보 가능



임서인

“여태껏 빅데이터라고 함은 대량의 데이터를 정보화하여 사용하는 것, 이라고만 생각했었는데 아무리 적은 양의 데이터라도 그것을 어떻게 가공하고 처리하는지에 따라 사용할 수 있는 가능성이 천차만별이라는 것을 이번 기회를 통해 경험할 수 있었습니다. 프로젝트 기간 내에 하지 못했던 한국어 전처리 과정을 더 정교하게 하여 완성도 있는 프로젝트를 만들고 싶습니다.”



황대명

“데이터를 수집하여 정제하는 방법에 따라 여러 가지 결과가 나올 수 있다는 점을 알 수 있었습니다. 한국어의 형태소 분석을 여러 가지 라이브러리를 통해 분석하며, 형태소 분석이 많은 양의 데이터와 예외 처리 과정을 통해 분석되고 있다는 것을 알 수 있었습니다.

개발을 하면서 여러 가지 오류에 대하여 예외 처리 과정에서 시행착오를 겪으며 여러 변수에 대한 대처 능력을 기를 수 있었습니다.

감사합니다.