적응형 맞춤 기사 포스팅 어플리케이션

가반 – 팀01

20170404 한종수
20170391 이찬서

Aim & Background

01

내용

파이썬을 이용한 기사 크롤링 및 정보 분석

+

안드로이드 스튜디오를 이용한

어플리케이션 개발

+

읽은 기사 목록을 통한 기사 추천

⬇

“CIY P1 어플리케이션”

02

중심 목표

“ 사용자의 선호도를 바탕으로 사용자에게 기사를 추천한다. ”

03

핵심성

1. 기사에서 키워드 추출
2. 기사의 가중치 분석
3. 사용자가 읽은 기사 기록
4. 정보 조합하여 추천 목록 생성
5. 추천된 기사의 내용 웹뷰로 HTML을 보여줌

Aim & Background

04

배경 설명, 사례 분석

유튜브를 시청할 때 알고리즘이 사람들에게 재미있는 영상을 추천하는 것을 보며 그런 알고리즘을 뉴스에 적용하면 어떨까라는 생각에서 시작

기존에도 설정을 통해 원하는 분야의 뉴스를 보여주는 어플이나 웹은 존재했지만 사용자가 읽었던 기사의 정보를 통해 사용자가 원하는 기사를 추천해주는 어플은 아님

05

문제 정의

위의 어플들은 현재 인기 많은 기사나 특정 주제의 기사들을 추천함.

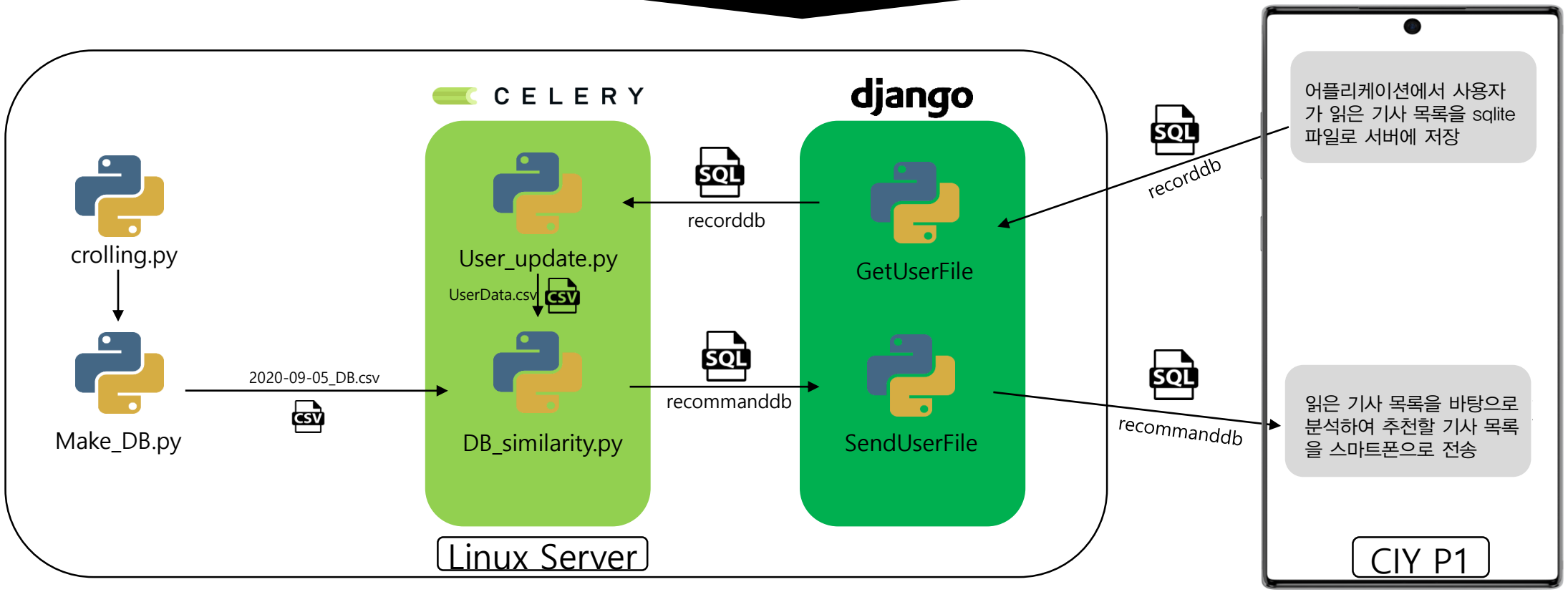
사용자가 읽은 기사를 토대로 추천하긴 하지만 기사의 세부내용이나 키워드가 아닌 카테고리를 분석하여 추천함

06

극복 방안

기사들의 키워드와 가중치를 분석하여 저장하고 사용자가 읽은 기사들을 통해 사용자의 키워드 선호도를 분석하여 사용자의 선호도와 비슷한 기사들을 추천한다.

전체적인 앱 구조(sw 관점)



크게 보아 두가지로 나눌 수 있다. 서버 그리고 어플리케이션으로. 서버에서는 데이터 분석, 수집의 핵심적인 과정을 수행하고, 어플리케이션은 이를 기반으로 사용자에게 의미있는 데이터를 보여준다. 핵심 구성 요소로 기사를 모으는 크롤링, 기사 텍스트의 자연어 처리, 기사와 사용자 관심도와의 유사성 계산 및 추천이 있다.

구현 방법 설명

서버

파이썬의 크롤링 패키지 beautiful soup를 이용하여 기사들을 텍스트형태로 수집한다. Page rank기법을 사용하는 키워드 추출 패키지 gensim을 사용하여 키워드와 가중치를 계산한 결과값을 데이터프레임에 저장한다.

크롤링되어 데이터프레임 형태로 저장된 데이터들을 다양한 방법(코사인 유사도, 피어슨 유사도)을 통해 분석하고 관계가 높은 순서대로 정렬한다.

위의 과정은 비동기로 수행하기 위해 비동기 서버 도구인 celery를 사용한다. 동기 서버는 django를 사용하며 어플로부터 실시간으로 올라오는 기록 파일을 저장하고 어플 실행시 실시간으로 기사 목록을 전달한다.

안드로이드 스튜디오

안드로이드 스튜디오에서 목록을 받기 위해 Asynctask를 통해 별도의 스레드로 서버에 연결하여 추천목록을 받는다. 추천목록을 받은 뒤 sqlite데이터 베이스를 사용하여 목록을 읽고 Adapter View의 형태로 리스트를 생성하여 기사 목록을 띄운다.

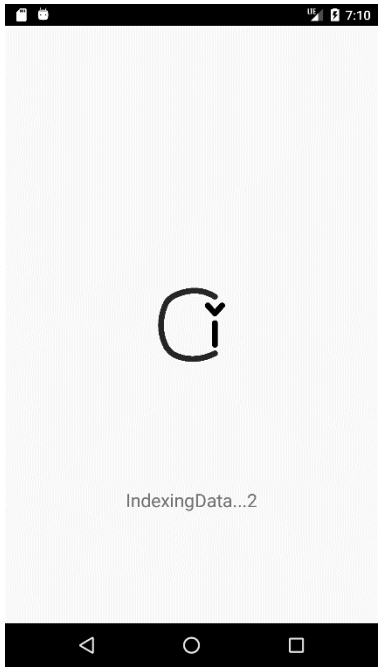
기사를 누르면 인텐트를 통해 어댑터 뷰에서의 기사 위치와 링크, 등을 함께 보낸다. 읽기 모드 액티비티에서 별도로 기사를 html 형태로 파싱하여 webview를 통해 보여주게 된다.

북마크버튼을 누르면 기기 내부에 Realm 형태의 NoSQL형태로 북마크 목록을 저장한다. 공유 버튼을 누르면 기사의 링크를 클립보드에 복사한다.

앱을 종료시키면 실행시킬때와 마찬가지로 Asynctask로 별도 스레드를 생성하여 서버로 데이터를 전송한다.

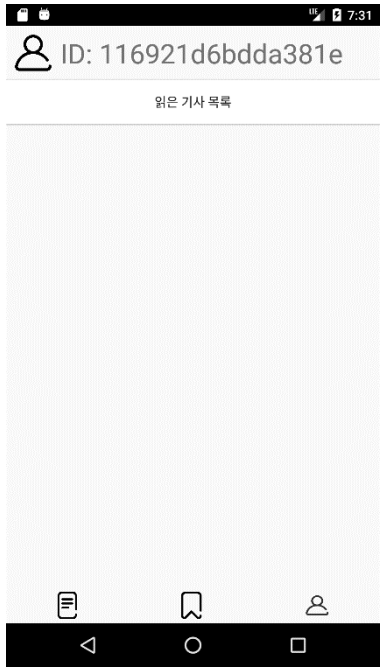
주요 결과 및 토의

1. 성공적으로 서버에서 추천 목록을 생성하여 스마트폰에서 보여줄 수 있었다.



로딩화면, 추천목록, 기사 조회 화면

2. 이미 더 완성도 있는 어플리케이션 뉴썸이 있지만, 더 많은 부가 기능을 넣어 뉴썸 보다 정확하고 빠른 어플리케이션을 만들 수 있지 않을까?



부가기능(북마크, 링크 클립보드 복사, 읽은 기사 목록) 화면

Reference

[1] newspaper, https://www.flickr.com/photos/troybaker/2283643909/in/photolist-J1R8E-4tNgLR-7neWRZ-5QDJoS-8n7HZo-93bzF7-2hno8s9-YN8nvc-4bZYoq-wmej-93K4FM-9tpcH5-adrd5t-9WDc71-9W2vYR-6QZZuU-bkx3pd-8EqApX-3T3L-aJGeri-9vxePZ-ejiLv3-a9g9d-RLc5YY-2aaaAK

[2] 나에게 맞는 뉴스를 추천해주는 뉴스 큐레이팅 어플 4, 삼성반도체 이야기(2018.11.16), https://www.samsungsemiconstory.com/1911

[3] sqlite logo, https://worldvectorlogo.com/logo/sqlite

[4] python logo, https://worldvectorlogo.com/logo/python-3

[5] android logo, https://worldvectorlogo.com/logo/android-1

[6] Djangologo, https://pixabay.com/ko/vectors/%EC%9E%A5-%EA%B3%A0-%EB%A1%9C%EA%B3%A0-%EC%9E%A5-%EA%B3%A0-%ED%94%84%EB%A1%9C%EC%A0%9D%ED%8A%B8-339744/

[7] celery logo with text, https://github.com/celery/celery

[8] Note 10, https://www.searchpng.com/2019/09/28/galaxy-note-10-pro-mockup-png-image-free-download/

Appendix

- 평가 시 참고할 수 있는 모든 자료
- 원칙적으로는 “5장으로 요약된 내용”으로만 평가되지만,
  내용 이해가 어려운 부분에 대해서는 부록을 참고할 수 있음
- 완결성을 파악하는 데 도움

Appendix

- 코드 Git Hub 링크
  -https://github.com/CodeStrich/CIY_Project_1

- 시연 영상 링크
  -Git Hub README.md 참고

Appendix

적용 기술

Appendix

- **코사인 유사도**

  : 코사인 유사도는 두 특성 벡터간의 유사 정도를 코사인 값으로 표현한 것이다.

  -1~1의 값을 가지며 -1은 반대되는 경우, 0은 독립적인 경우, 1은 같은 경우이다.

$$x \cdot y = |x||y| \cos\theta$$

$$\cos\theta = \frac{x \cdot y}{|x||y|} \qquad \text{cosine\_sim}(i,j) = \frac{\sum_{u \in U_{ij}} r_{ui} \cdot r_{uj}}{\sqrt{\sum_{u \in U_{ij}} r_{ui}^2} \cdot \sqrt{\sum_{u \in U_{ij}} r_{uj}^2}}$$

- **피어슨 유사도**

  : 피어슨 유사도는 두 벡터의 상관계수를 의미한다.

$$\text{pearson\_sim}(i,j) = \frac{\sum_{u \in U_{ij}} (r_{ui} - \mu_i) \cdot (r_{uj} - \mu_j)}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \mu_i)^2} \cdot \sqrt{\sum_{u \in U_{ij}} (r_{uj} - \mu_j)^2}}$$

Peer review (타 분반에서 3명씩 평가)

- 신규성
  - 새로운 부분이 있는지? (+1)
  - 얼마나 창의적인지? (+1)
  - 실제로 수요가 있을지? (+1)
- 진보성
  - 비교/분석이 충분히 이루어졌는지? (+1)
  - 해당 분야에서 얼마나 경쟁력 있는지? (+1)
  - 다른 응용에도 파급력이 있는지? (+1)
- 완결성
  - 얼마나 잘 구현하였는지? 제시된 방법이 실현 가능성이 있을지? (+1)
  - 핵심 내용 및 세부 사항들을 이해하기 쉽게 잘 전달하였는지? (+1)
  - 기술 난이도가 충분히 도전적이었는지? (+1)
  - 창의/융합/공동체/의사소통/리더쉽/글로벌 역량 관련 기타 요소 (+1)