**최종 보고서**

**Machine Learning을 이용한**

**인공지능 뉴스 어플리케이션**

**(Today News)**

Ver. 1.0

2019.12.03

한국외국어대학교

정보통신공학과

Team MUD

**문서 정보**

|  |  |  |  |
| --- | --- | --- | --- |
| **구분** | **소속** | **성 명** | **날 짜** |
| **작성자** | **한국외국어대학교** | **박주영(T)** | **2019.12.03** |
| **한국외국어대학교** | **김아연** | **2019.12.03** |
| **한국외국어대학교** | **김혜원** | **2019.12.03** |
| **한국외국어대학교** | **홍승환** | **2019.12.03** |
| **한국외국어대학교** | **이산가 비두샤** | **2019.12.03** |
| **검토자** | **한국외국어대학교** | **박주영(T)** | **2019.12.03** |
|  |  |  |
|  |  |  |
| **사용자** |  |  |  |
|  |  |  |
|  |  |  |
| **승인자** | **한국외국어대학교** | **홍진표** | **2019.12.03** |

**머리말**

Today News는 뉴스 요약 및 추천 서비스를 제공하는 AI 뉴스 어플리케이션이다. Machine Learning을 사용하여 개인화 맞춤 추천 서비스, 뉴스 요약본 및 헤드라인 생성 시스템을 구현하였으며 이에 대한 소개와 설계 방법, 적용 방안을 자세히 기술하고 있다.

**개정 이력**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **이력** | **작성자** | **개정일자** | | **개정내역** |
| **1.0** | **박주영** | **2019.11.20** | | **초안 작성** |
| **김아연** |
| **검토자** | **박주영** | | |
| **1.1** | **김아연** | **2019.11.25** | | **서비스 개요, 기대효과 작성** |
| **검토자** | **박주영** | | |
| **1.2** | **홍승환** | **2019.11.30** | | **시스템 설명, 기능 설명** |
| **김아연** |
| **검토자** | **박주영,** | | |
|  |  |  | |  |
|  |
| **검토자** | **박주영,** | | |
| **2.0** |  | **2019.12.02** | **최종 보완 및 수정** | |
|  |
|  |
|  |
|  |
| **검토자** | **박주영** | | |

**목차**

[**1. 개요** 8](#_Toc26027213)

[1.1 목적 8](#_Toc26027214)

[1.2 참고 문서 8](#_Toc26027215)

[**2. 서비스 개요** 9](#_Toc26027216)

[2.1 서비스 기획 배경 9](#_Toc26027217)

[2.2 서비스 기능 소개 9](#_Toc26027218)

[**3. 시스템 설명** 11](#_Toc26027219)

[3.1 전체 시스템 구성도 11](#_Toc26027220)

[3.2 세부 시스템 구성도 11](#_Toc26027221)

[3.2.1 뉴스 생성 시스템 11](#_Toc26027222)

[3.2.2 뉴스 제공 시스템 11](#_Toc26027223)

[3.2.3 뉴스 추천 시스템 12](#_Toc26027224)

[3.3 소프트웨어 12](#_Toc26027225)

[**4.시스템 상세 설계** 13](#_Toc26027226)

[4.1 App & web Server 13](#_Toc26027227)

[4.1.1 ‘(자동)로그인/로그아웃/회원가입기능’ 요구사항(REQ\_100\_10)에 대한 상세 설계 13](#_Toc26027228)

[ 4.1.2 스크랩 기능 요구사항(REQ\_100\_17)에 대한 상세 설계 14](#_Toc26027229)

[ 4.1.3 TTS 기능 요구사항(REQ\_100\_23)에 대한 상세 설계 15](#_Toc26027235)

[ 4.1.4 잠금 화면 기능 요구사항(REQ\_100\_26)에 대한 상세 설계 15](#_Toc26027240)

[4.2 News Produce System 16](#_Toc26027241)

[4.3 News Clustering System 19](#_Toc26027242)

[ 4.3.1 Clustering 사용 과정 19](#_Toc26027243)

[ 4.3.2 최적의 K값찾기 19](#_Toc26027244)

[ 4.3.3 Clustering의 중심 정하기 20](#_Toc26027245)

[ 4.3.4 전처리 과정 21](#_Toc26027246)

[ 4.4.4 K\_means 적용 22](#_Toc26027247)

[4.4 News Provide System 23](#_Toc26027248)

[ 4.4.1 Headline summary Model 생성(REQ\_100\_21)에 대한 상세 설계 23](#_Toc26027249)

[ 4.4.2 ML processing 24](#_Toc26027250)

[4.5 News recommending System 25](#_Toc26027251)

[ 4.5.1 추천 시스템 상세 설계 25](#_Toc26027252)

[ 4.5.2 아이템 기반 추천 25](#_Toc26027253)

[ 4.5.3 사용자 기반 추천 시스템 26](#_Toc26027254)

[4.6 Database 28](#_Toc26027255)

[**5. 기능 설명** 29](#_Toc26027256)

[5.1 인터페이스 및 기능 설명(여기는 앱의 인터페이스!) 29](#_Toc26027257)

[ 4.4.1 로그인 및 회원가입 29](#_Toc26027258)

[ 4.4.2 홈 화면 29](#_Toc26027259)

[ 4.4.3 카테고리 창 29](#_Toc26027260)

[ 4.4.4 프로필 창 29](#_Toc26027261)

[5.2 Sequence Diagram 29](#_Toc26027262)

[5.3 User flow chart (다른 다이어그램으로 추가도 고려) 29](#_Toc26027263)

[5.4 예상 시나리오 29](#_Toc26027264)

[**6. 적용방안 및 기대효과** 33](#_Toc26027265)

[**7. 프로젝트 세부 추진 계획 및 일정** 34](#_Toc26027266)

[**8. 팀원 담당 업무** 35](#_Toc26027267)

**그림 목차**

# **1. 개요**

뉴스 어플리케이션 ‘Today news’는 인공지능 기술을 이용하여 오늘의 뉴스를 요약하여 제공하고 사용자별로 뉴스를 추천하여 준다. 본 장에서는 해당 시스템에 대한 목적과 범위, 참고문서를 소개한다.

## **1.1 목적**

본 프로젝트에서는 인공지능 뉴스 편집 기술을 구현하여 기존에 사람이 직접 편집하는 방식보다 중요한 정보를 신속하게 제공하고 사용자의 관심사를 파악하고 원하는 뉴스를 제공하는 시스템을 구축하는데 목적을 둔다. 이 프로젝트를 진행하기 위해 다음 사항을 구체적으로 명시하고 구현하였다.

* K-means 클러스터링을 이용하여 실시간 뉴스 군집화
* Lexrank 알고리즘(추출적 요약)과 Attention mechanism RNN 모델(추상적 요약)을 이용한 본문 요약과 헤드라인 추출
* content-based filtering과 collaborated user-based filtering을 사용하여 사용자에게 뉴스 추천

## **1.2 참고 문서**

|  |  |
| --- | --- |
| **문서** | **문헌 제목** |
| **한국정보과학회** | **Self-attention 기반의 다중 문서 인코더를 통한 추상적 다중 문서 요약 생성** |
| **한국정보과학회** | **자가 주의 메커니즘을 활용한 seq-to-seq 기반 문서 생성 요약** |
| **고려대학교 산업경영공학과/서울대학교 산업공학과** | **추천 시스템 기법 연구동향 분석Review and Analysis of Recommender Systems** |
| **한국정보과학회** | **lexrankr: LexRank 기반 한국어 다중 문서 요약** |
| **한국정보과학회** | **그래프 군집화기반의 다중문서요약 기법 Multi-Document Summarization Based on Graph Clustering** |

# **2. 서비스 개요**

Today News 서비스는 뉴스를 보고 싶어 하나, 뉴스를 찾아보거나 뉴스 기사 하나하나를 읽어보기 귀찮아하는 사람들을 위해 기존 뉴스들보다 접근성 있고, 시각적으로 한눈에 오늘의 뉴스를 알려주는 어플리케이션이다. 즉, 뉴스를 봐야겠다고 생각은 하나, 뉴스를 보지 않는 타겟에게 오늘 하루의 뉴스를 브리핑해주는 어플리케이션이다.



## **2.1 서비스 기획 배경**

최근 Time-Poor족 즉, 시간이 부족하다고 생각하는 사람들이 많이 늘어나고 있다. 그래서 자신이 원하는 정보만을 빠른 시간 내 습득할 수 있는 요약 시장 또는 개인화 맞춤 서비스 시장이 활발해 지고 있다. 사람들은 빅데이터 시대 속에서 빠른 시간 내에 자신이 원하는 정보만을 효율적으로 얻어야 한다. 그 중 방대한 정보를 담고 있는 뉴스를 통계에 따르면 하루에 67개의 매체에서 평균 25,866개의 기사를 보내고 있다. 이러한 기사들을 전부 읽을 수는 없으며 간략화 하여 접근성을 낮출 필요가 있다고 생각하여 개인 맞춤 요약 뉴스를 제공하는 Today News 서비스를 기획하게 되었다.

## **2.2 서비스 기능 소개**

**(1) 뉴스 요약본 제공**

Today News는 사회, 정치, 경제, IT 총 4개의 카테고리로 분류된 뉴스들을 실시간으로 보여준다. 뉴스의 전문을 보여주는 것이 아닌 요약해서 단 3줄만 보여준다.

**(2) 주제별 뉴스**

실시간으로 제공되는 뉴스들은 비슷하거나 같은 내용의 뉴스들이 많으므로 사용자가 한 번에 주제를 파악하고 그 주제의 다양한 뉴스들을 제공받기 위해 뉴스를 주제별로 군집하여 묶어서 보여준다. 같은 주제로 묶인 뉴스들이 하나의 내용으로 요약되어 새로운 요약본을 사용자에게 보여주며 그 군집에 맞는 헤드라인을 제공하여 사용자에게 한 눈에 주제를 파악할 수 있도록 보여준다. .

**(3) 개인화 추천 기능**

Today News는 사용자의 경험을 중요시하고 편리함을 보장하기 위해 개인화 맞춤 추천 기능을 제공한다. 뉴스를 제공하는 홈 피드는 추천 알고리즘을 통해 사용자의 취향에 맞는 뉴스들로 채워지게 된다. 사용자가 뉴스에 평점을 매기면 그 점수를 기반으로 알고리즘이 돌아가게 된다.

**(4) 북마크 기능**

사용자는 나중에 다시 읽고 싶은 기사를 북마크(스크랩)기능을 통해 저장해 둘 수 있다. 스크랩한 뉴스는 프로필창에서 확인 할 수 있다.

**(5) 뉴스 브리핑 기능**

홈 피드 상단에 스피커 아이콘을 누르면 오늘 뉴스의 주제들을 브리핑 기능을 통하여 음성으로 들을 수 있다.

# **3. 시스템 설명**

## **3.1 전체 시스템 구성도**

스크린샷이(가) 표시된 사진

자동 생성된 설명

[Today’s News 시스템 구성도]

## **3.2 세부 시스템 구성도**

Today News는 크게 3개의 서버로 구성되어 진다. 클라이언트와 서버(Today server)간 중계역할을 하는 REST API 서버인 linux server, Today News의 전체적인 서비스 및 기능을 제공하는 Today server, 딥 러닝 모델을 생성하는 ml\_headline server 3개의 서버로 구성된다.

Today News APP은 정보를 요청 할 때, Django rest frame work를 사용하여 linux server와 통신한다. Today server는 News Produce System, News Cluster System, Headline & Multi-summary System, News Recommend System 총 4개의 시스템으로 구성되어 있으며 각각의 System을 거쳐 생성된 데이터는 linux server를 거쳐 DB에 저장된다.

**3.2.1 News Produce System**

News Produce System에서는News Crawling과 News Summary을 통해 실시간으로 다양한 뉴스들을 크롤링하여 가져와 뉴스 본문을 3줄로 요약한다. 각 뉴스마다 ‘뉴스 생성 시간, 헤드라인, 요약 내용, 언론사, 카테고리’ 정보를 가지고 있으며 이 데이터는 리눅스 서버를 거쳐 DB에 저장된다.

**3.2.2 News Cluster System**

News Cluster System은 요약된 후 DB에 저장된 뉴스들을 군집화 시켜주는 News clustering과 군집이 완료된 뉴스를 지워주는 News delete로 구성되어진다. 클러스터링이 완료된 뉴스들은 DB의 Cluster table에 저장된다.

**3.2.3 Headline & Multi-summary System**

Headline & Multi-summary에서는 News multi-summary를 통해 같은 주제로 군집화 된 뉴스들을 대표하는 새로운 요약 뉴스를 생성한다. News headline processing에는 딥러닝 학습을 거쳐 미리 저장된 모델이 있으며 이 모델은 새로운 요약 뉴스에 맞는 새로운 제목을 생성하여 준다.

**3.2.4 News Recommend System**

뉴스 추천 시스템은 앞의 시스템들을 거쳐 재가공 되어진 뉴스를 사용자에게 개인 맞춤으로 추천해주는 시스템이다. 뉴스 데이터 양에 따라 contents-based recommendation과 Collaborative recommendation을 각자 적합한 환경에서 사용하게 된다.

**3.2.4 Database**

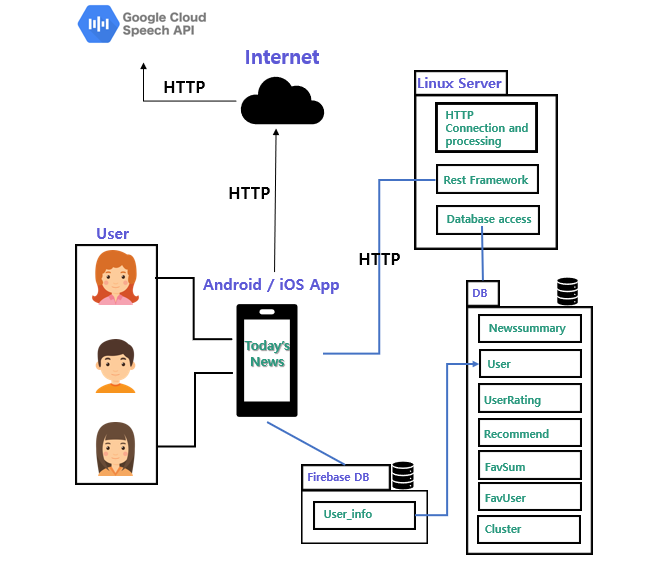
|  |  |
| --- | --- |
| **newssummary** | News crawling과 summary한 정보가 들어있다. (제목, 요약 내용, 카테고리 이름, 클러스터 아이디, 뉴스 업로드 날짜) |
| **User** | 회원 정보에 대한 내용이 들어있다(User id) |
| **UserRating** | 회원이 뉴스를 점수 평가한 정보가 들어 있다 |
| **FavSum** | 모든 회원들이 스크랩한 해당 뉴스들이 들어있다(이때, 이 뉴스들의 정보들은 영원히 지워지지 않는 뉴스이다) |
| **FavUser** | 해당 회원이 스크랩한 뉴스에 대한 정보가 들어있다. |
| **Cluster** | 뉴스 군집화된 뉴스들의 cluster\_id를 부여한 정보와 해당 cluster\_id에 속한 정보가 포함되어있다 |

## **3.3 소프트웨어**

|  |  |
| --- | --- |
| **구성요소** | **설명** |
| Tensorflow | 구글이 개발한 오픈소스로 공개한 머신러닝 라이브러리이다. |
| MySQL |  |
| Django |  |
|  |  |
|  |  |
|  |  |

# **4.시스템 상세 설계**

## **4.1 App & web Server**



[시스템 구성 요소 1: App & Linux server]

**4.1.1 ‘(자동)로그인/로그아웃/회원가입기능’ 요구사항(REQ\_100\_10)에 대한 상세 설계**

|  |  |
| --- | --- |
| **회원가입 기능** | 사용자가 Today’s News APP에서 Email과 Password와 Name을 입력하고 회원 가입을 하면 data를 Firefase로 보낸다. Firefase에서 유효ID를 Database ORM을 전송하여 DB를 만들고 데이터를 저장한다 |
| **로그인 기능(자동)** | -사용자가 Today’s News APP에서 Email과 Password를 입력하고 로그인 누르면 data를 Firebase로 보낸다. Firebase를 통해 유효한 Email과 password인지 검사한다. 유효한 정보라면 로그인 성공된다.  -한 번 로그인에 성공 했으면 다음에 앱을 실행 시킬 시 자동 로그인 된다[local DB에 저장되기 때문이다]. |
| **로그아웃 기능** | 사용자가 로그아웃 버튼을 누르면 로그아웃 된다[local DB에서 저장되어 있던 User 정보를 삭제한다] |

**4.1.2 스크랩 기능 요구사항(REQ\_100\_17)에 대한 상세 설계**

|  |  |
| --- | --- |
|  | 사용자는 뉴스 옆의 ☆버튼을 눌러 원하는 뉴스를 스크랩할 수 있다. 이때 스크랩한 뉴스에 대한 ID값을 보내면 해당 뉴스 ID에 해당되는 제목과 요약 내용을 전달한다. 전달 받은 총 3가지의 정보를 FavSum table에 저장하고, User\_Id와 뉴스 ID를 FavUser에 저장한다. |
|  | 사용자는 스크랩한 뉴스들은 보기 위해서 Linux server로 FavUser table에 포함된 news\_id를 통해 해당 user가 스크랩한 뉴스를 알 수 있다.  스크랩한 뉴스의 id를 통해 FavSum에서 뉴스 제목과 요약본을 받아 자신의 스크랩 공간에서 확인 할 수 있다 |

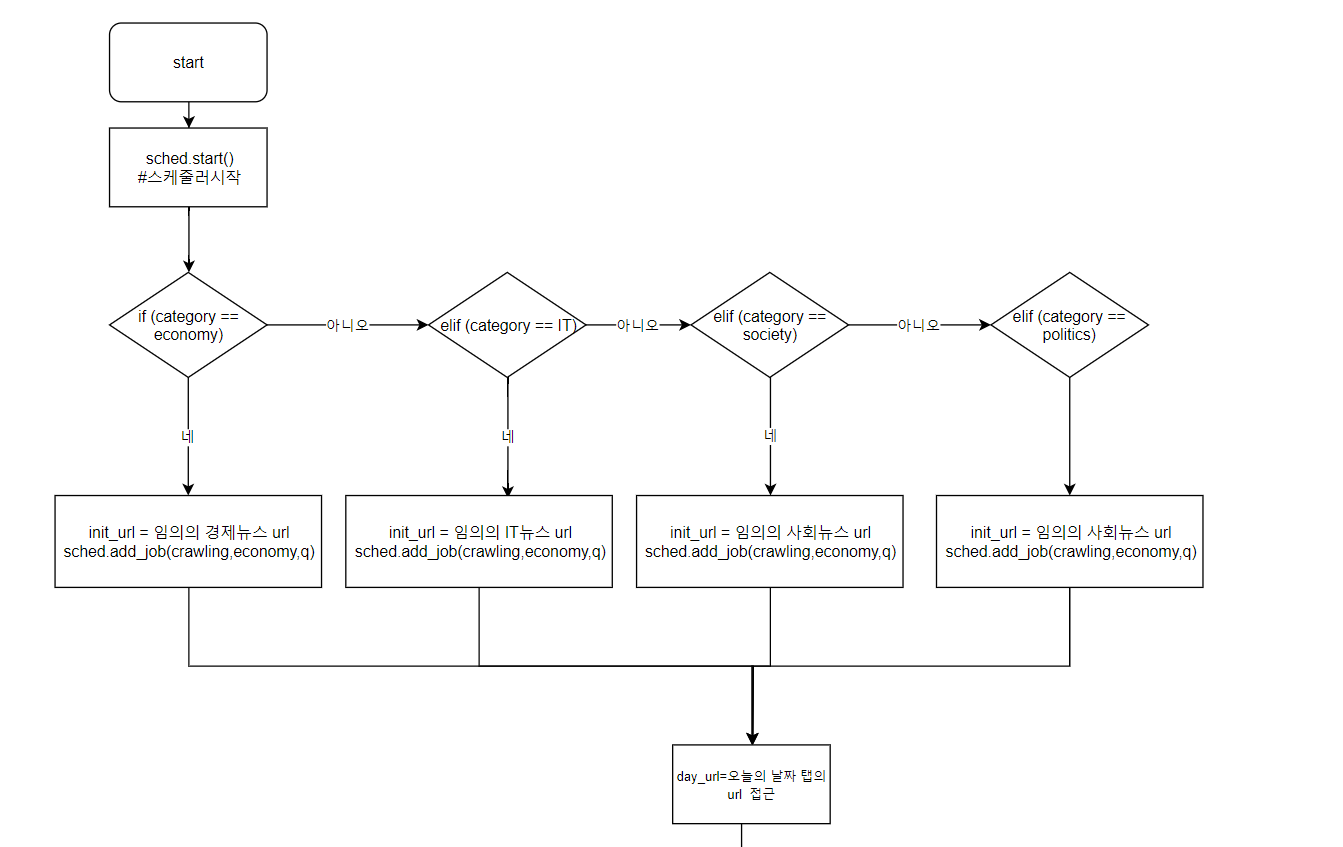
**4.1.3 TTS 기능 요구사항(REQ\_100\_23**)**에 대한 상세 설계**

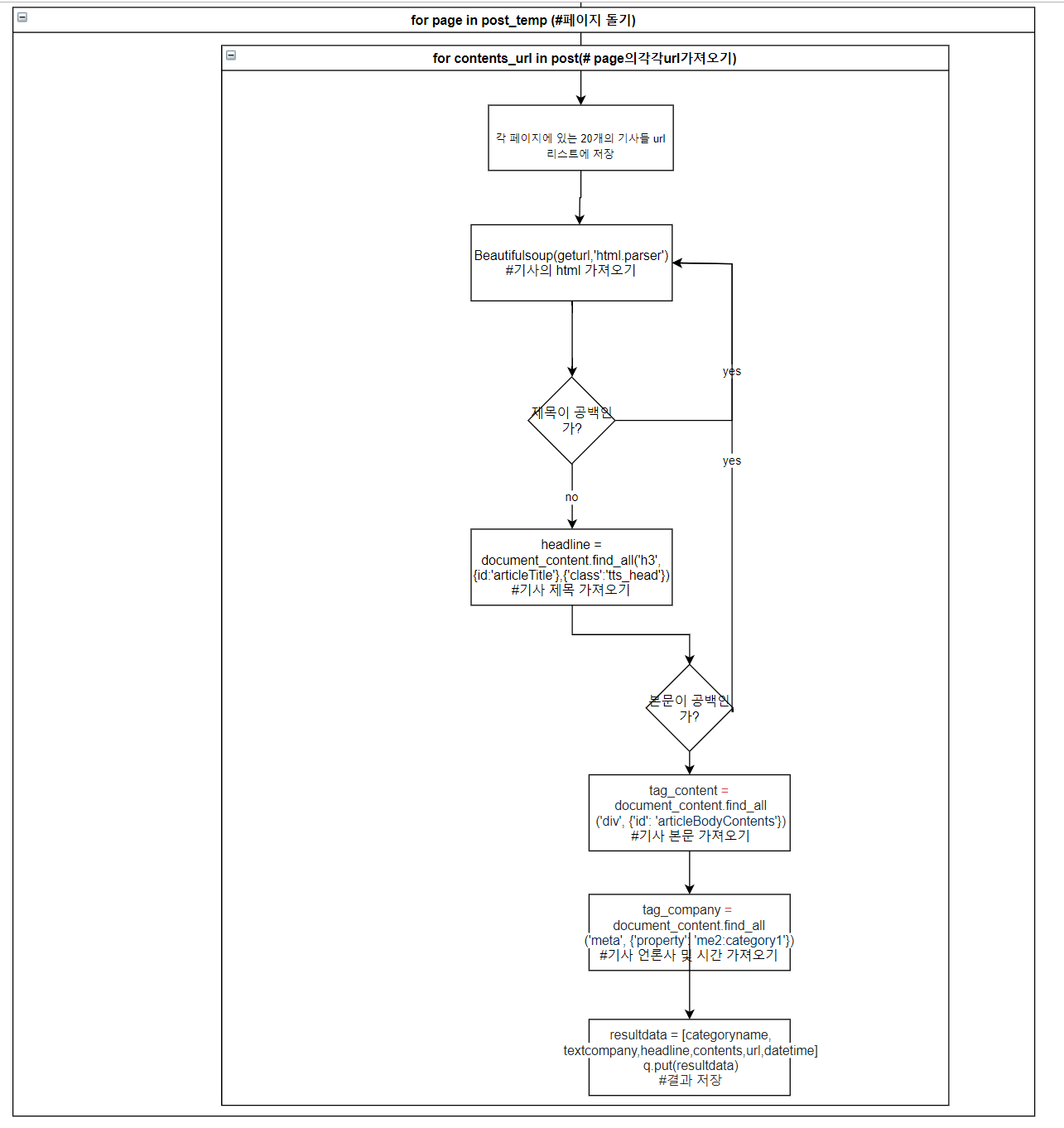
|  |  |
| --- | --- |
|  | 1. 사용자가 Linux server로 군집화된 뉴스의 다중 문서 요약본을 요청하면, Linux는 다중 문서 요약본을 제공하여 APP에서 UI로 띄운다.  2. 사용자가 ‘스피커’ 버튼을 누르면 Google TTS API로 Data[다중 문서 요약본]을 요청한다.  3. Google TTS API는 Streaming 방식으로 APP으로 Data의 음성을 전달한다. |

**4.1.4 잠금 화면 기능 요구사항(REQ\_100\_26)에 대한 상세 설계**

|  |  |
| --- | --- |
|  | Todaynews의 setting(profile)창에서 잠금 화면 위젯으로 설정하면 사용자는 뉴스를 잠금 화면에서 바로 확인 가능하다. 이때, 뉴스는 가장 최신 뉴스를 반영한다. |

## **4.2 News Produce System**





[시스템 구성 요소 3.2: News crawling]

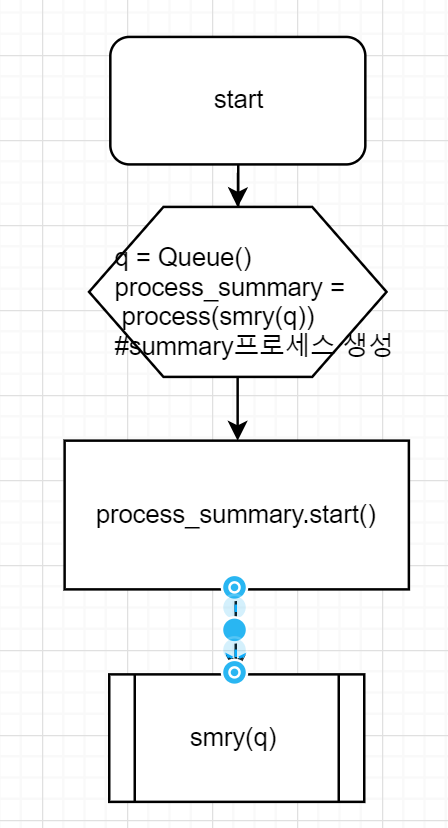
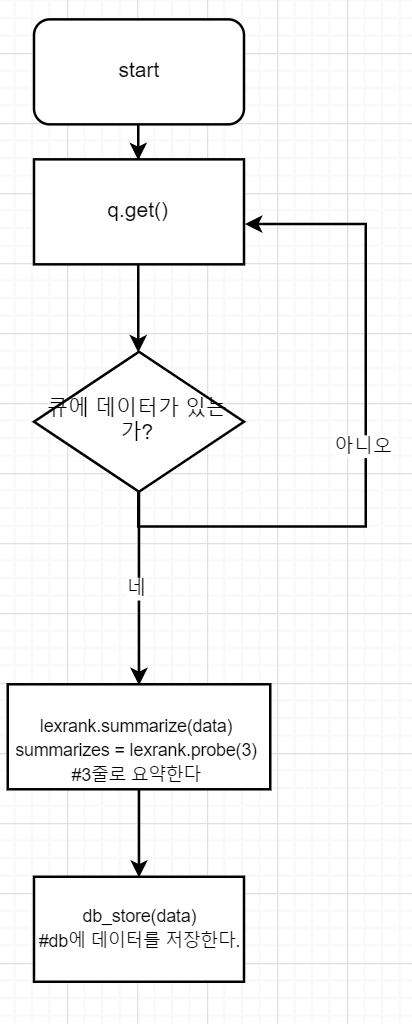
1) 네이버뉴스의 속보란에서 경제, 사회, 정치, IT 카테고리내의 오늘의 뉴스를 수집한다.

2) 스케줄러를 이용하여 일정 시간 간격으로 계속하여 crawler함수를 실행시킨다. 경제: 5분, 사회: 3분, 정치: 7분, IT:15분 간격으로 크롤링을 실행한다.



3)크롤러의 url 접근 순서 : 오늘의 날짜 탭 url 가져오기 > 페이지 url 가져오기>페이지 내 각 기사의 url 가져오기 > html을 이용하여 기사 제목,본문,날짜,언론사 가져오기

4) 이때 크롤링된 내용은 resultdata = [news title, news contents, category, date, URL, publication]로 리스트 형식으로 저장되며 이 값은 q.put(resultdata)로 큐에 저장된다.

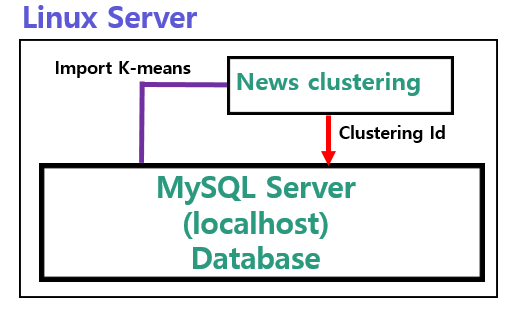
 

[시스템 구성 요소 3.3: News Contents summary]

1) 멀티프로세싱을 이용하여 crawler()와 smry()함수를 동시에 실행시킨다. 큐를 사용하여 데이터를 전달할 것이기 때문에 q = Queue()를 생성한 후 summary 프로세스를 생성한다.

2) Queue에 데이터가 있는 경우, 데이터를 data = q.get() 으로 가져온다. Lexrank를 이용하여 3줄로 요약한다. 그 후 db\_store()함수를 통해 데이터를 summarynews 디비 테이블에 [category\_name, text\_company, text\_headline, summary,url, news\_datetime]값을 업데이트 한다.

## **4.3 News Clustering System**



[시스템 구성 요소 4: News Clustering]

**4.3.1 Clustering 사용 과정**

위의 구성도 그림과 같이 *Today News App* 에서 **Clustering**은

Crawling된 원본 기사들이 Contents summary를 통해 1차 요약이 된 후 DB에 저장이 되는데

이를 DB에서 불러와 사용한다. 이때 DB속 요약된 신문기사들을 요약기사라고 하겠다.

요약기사들은 사용자에게 각각 개별로 보여지는 것이 아니라, 비슷한 소재의 기사들을 그룹화하여 대표적인 것으로 축약하여 보여 줄 것이다. 따라서 이 요약 기사들이 비슷한 소재끼리 묶어 주어야 한다. 묶는 다는 것 즉, Clustering 그룹화를 해야하는데 그 중 K\_mean clustering기법을 이용한다.

Pycharm의 module인 K-means를 사용하여 요약 기사들을 clustering 하겠다.

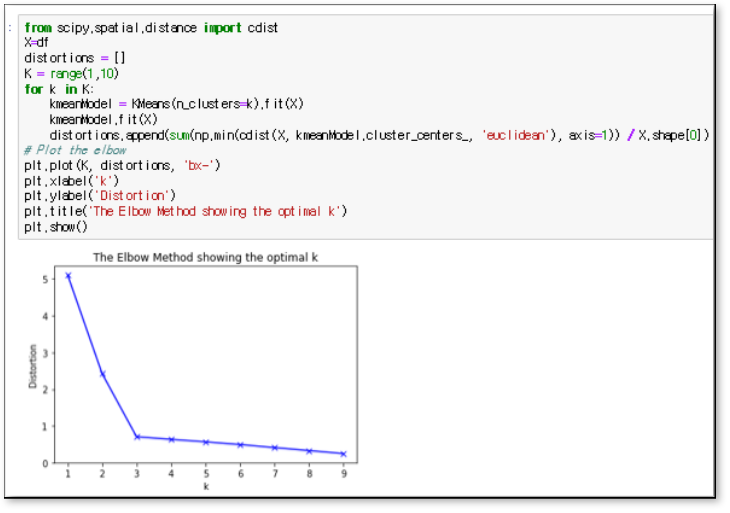
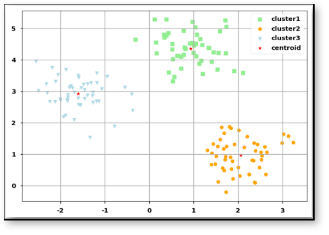
Clustering이 끝나면, 각 요약 기사들은 각각 clustering ID가 부여 받는다. 해당 cluster의 기사 개수로 cluster의 크기를 측정하여 크기 순으로 Headline에 들어갈 기사의 주제를 정한다.

**4.3.2 최적의 K값찾기**

최적의 K값, 즉 cluster의 개수는 K-means 모듈을 사용하여 K-means를 선언 시에 꼭 넣어 줘야하는 값이다. 이는 k-mean기법의 단점이기도 하다. 하지만 아래 그림과 같이 최적의 K값을 찾아 넣어주면 해결 될 수 있는 단점이다.

그림에서 오른쪽 그래프는 data들을 벡터화 시켰을 때 x축, y축을 기준으로 어떻게 분포 되어 있는 지를 알 수 있다. 이 분산 정도를 통해 분산 시 특정 몇 곳을 중심으로 분산되어 있다면 그림과 같이 k=3에서 적절한 k값임을 보여준다. 최적의 k값 찾기 함수에서 기울기가 급격히 감소되었을 때 가 최적의 K값이다.

이렇게 구해진 K를 KMeans를 선언 시, n\_clusters에 대입한다.



[Figure 3.4] 1\_최적의 K값 찾기

**4.3.3 Clustering의 중심 정하기**

k-means 는 initial points 가 제대로 설정되지 않으면 불안정한 군집화 결과를 학습한다고 알려졌다. 사실 k-means 의 학습 결과가 좋지 않는 경우는 initial points 로 비슷한 점들이 여러 개 선택 된 경우이다. 이 경우만 아니라면 k-means 는 빠른 수렴 속도와 안정적인 성능을 보여준다.



▷**randomly select centroid**

▷**randomly select centroid**

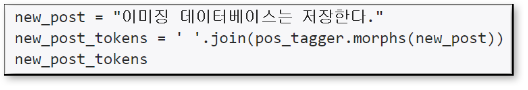
: 임의 data에 c1지정, c1에서 가장 먼 것, c3는 c1과 c2와 가장 먼 것 (cN은 N번째 centroid)

이전 점들과 멀리 떨어진 점들이 선택되다 보면 자연스레 서로 떨어진 점들이 선택될 것.

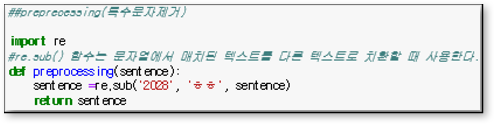
그러나 문서 군집화 과정에서 k-means++ 을 이용한다는 것은 **“매우 비싼 random sampling”**을 수행하는 것이다. 간단한 수치 data를 이용했을 땐 random과 k-means++은 차이가 없었다. 문서에서도 결과값의 성능에 차이가 없다면 random을 쓸 예정이다.

**4.3.4 전처리 과정**

**▷형태소 분석 처리**

****

**▷불 용어 제거**



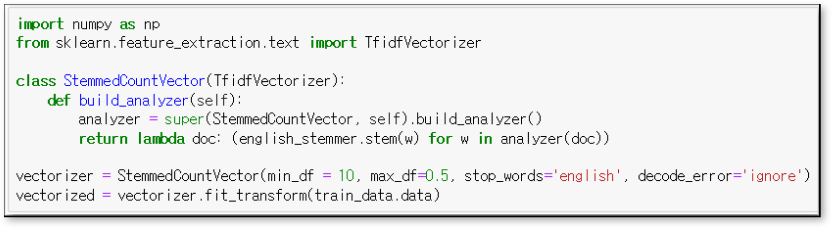
**▷Vectorization**

군집을 만들기 위해 가장 적절한 방법은 게시물마다 등장하는 단어의 빈도수를 파악해 하나의 카운트 벡터로 만듭니다. 이를 단어 주머니 접근 법이라고 합니다. 카운트 벡터 생성 후 해당 게시물과 다른 게시물 사이의 벡터 거리를 계산하여 게시물 사이의 유사도를 파악하면 됩니다.

▶TfidfVectorizer :

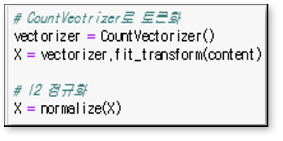
CountVectorizer랑 비슷하지만 TF-IDF방식으로 단어의 가중치를 조정한 BOW 벡터를 만든다.

(CounterVectorizer의 서브클래스로 CountVectorizer를 이용해 BOW를 만들고 TfidTransformer를 사용해 tf-idf로 변환)



▶CountVectorizer :

문서 집합에서 단어 토큰을 생성하고 각 단어의 수를 세어 BOW 인코딩한 벡터를 만든다.

1.  문서를 토큰 리스트로 변환한다.
2. 각 문서에서 토큰의 출현 빈도를 센다.
3. 각 문서를 BOW 인코딩 벡터로 변환한다.

**4.4.4 K\_means 적용**

KMeans module을 통해 아래와 같이 K개의 n\_cluster로 분류할 수 있다.

먼저 정해진 군집 개수 K , init 등 변수 값을 넣고 초기 설정을 한다.

이후 앞서 걸러진 전처리된 dataset을 대입시켜 K개의 군집화를 진행한다.

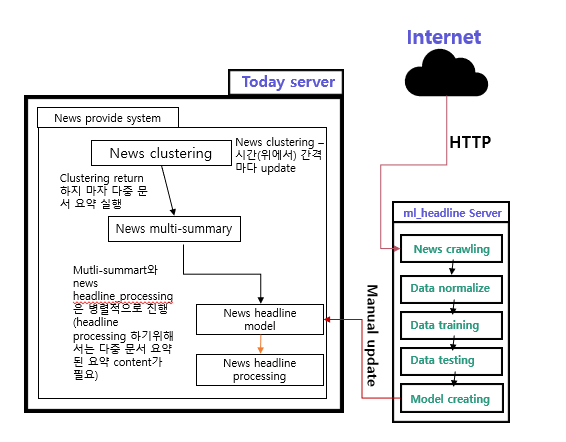
고려해야 할 점은 앞으로 새로운 뉴스 군집 설정할 때이다.



위 테스트는 이미 군집화 된 label을 새로운 data가 들어오면 어떤 cluster에 속할지 거리로 예측하는 것이다.

앞으로 새로운 data가 들어왔을 때 새 군집을 만들어야 할 때를 명시하고 이전 군집들과 중복되지 않게 군집화 하는 방법을 테스트해봐야 한다.

## **4.4 News Provide System**



[시스템 구성 요소 3.6: headline summary]

Headline과 summary를 설명하기 앞서, 앞 3.4와 3.5에서 news clustering(군집화)가 완료되면 군집화된 뉴스들을 다중 요약(multi-summary)한다. 다중 요약과 ml\_headline\_processing을 병렬적으로 진행하기 위해서 둘 사이에 queue를 이용한다. 다중 요약된 cotents들은 queue에 put하고 ml\_headline\_processing처리할 때 queue 에서 다중 요약된 contents들을 get하여 처리한다.

**4.4.1 Headline summary Model 생성(REQ\_100\_21)에 대한 상세 설계**

1. Headline summary Server에서 모델 생성 시 필요한 dataset을 news crawling을 통하여 얻는다. 이때 news crawling 코드는 위 news crawling에서 하루치 대신 한달 치로 바꾸어 crawling한다
2. crawling된 뉴스들은 먼저 data 전처리 과정을 거치게 된다. 그 다음 정수 인코딩 해주고 마지막으로 padding해준다

|  |  |
| --- | --- |
|  |  |
| **Data 전처리 과정** | **정수 인코딩** |

1. embedding\_attention\_seq2seq는 encoder와 embedding attention decoder를 만들어준다

|  |
| --- |
|  |

1. model train 시킨다

|  |
| --- |
|  |

1. model train 완료하면 test하여 headline model을 생성한다

|  |
| --- |
|  |

1. 위 과정을 모두 마치면 생성된 모델을 Linux server로 전송하여 model update 시킨다

**4.4.2 ML processing**

headline summary server에서 받은 model에 의해 처리되어 군집화 된 뉴스들을 MySQL에서 가져온다. 군집화 된 뉴스들의 Headline을 추출하기 위해 미리 정의된 headline\_summary\_mo del.py를 불러와 군집화 된 뉴스들을 input값으로 넣고 output값으로 headline을 얻는다. 추출된 headline은 MySQL에 저장된다.

## **4.5 News recommending System**

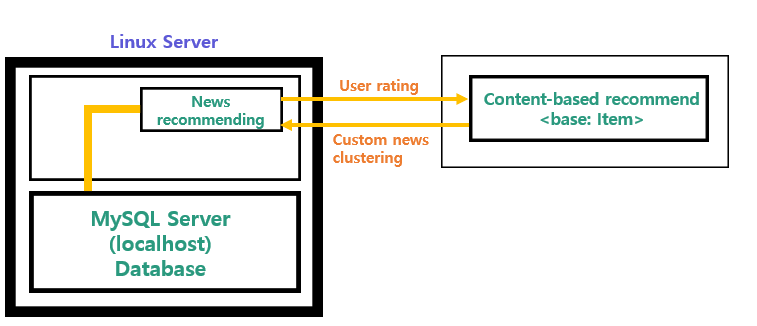
**4.5.1 추천 시스템 상세 설계**

추천 시스템은 콘텐츠 기반과 협업 필터링으로 나뉘며 ‘투데이 뉴스’는 각각의 장단점을 보완한 하이브리드 추천 시스템을 채택한다. 초기는 콘텐츠 기반 알고리즘을 사용하고 일정량의 사용자 데이터가 모이면 협업 필터링 알고리즘을 사용한다.

▷ **평점과 추천 뉴스**

사용자는 클러스터링 된 뉴스 묶음 내에 있는 하나의 뉴스에 대해 열람 시 무조건 1에서 5점(1점 단위)까지의 평점을 매겨야 한다. 이 평점 데이터를 기준으로 추천되는 콘텐츠는 낱개의 뉴스가 아닌 클러스터링 된 뉴스 묶음이어야 한다. 추천 뉴스 다음과 같은 순서로 나열되어 보여진다. (1) 사용자가 아직 보지 않은 뉴스 군집을 예측 평점 순으로 나열, (2) 사용자가 이미 보았던 뉴스 군집을 기존 평점 순으로 나열.사용자가 뉴스에 평점을 부여하면 rating DB에 UserID, news\_ID, score가 업데이트 된다.

**4.5.2 아이템 기반 추천**

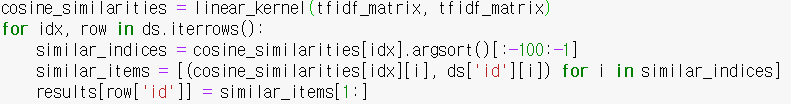


[시스템 구성 요소 6.1: content-based recommend]

초기의 사용자가 평가한 뉴스 기사 데이터가 없을 시 Collaborative filtering을 사용할 수 없기 때문에 사용자의 뉴스 기사 평점 데이터가 어느정도 쌓이기 전까지는 Content-based recommend 중 아이템 기반 추천 알고리즘을 사용하여 사용자가 평점을 매긴 뉴스와 유사한 뉴스 찾아 그 뉴스가 해당하는 Cluster와 그 Cluster 안의 뉴스 기사들을 보여준다.



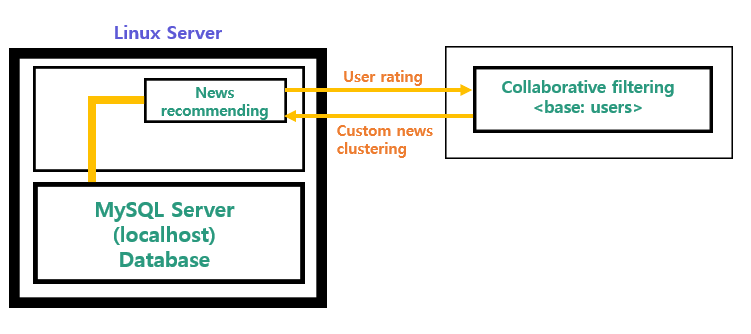
[TF-IDF]



[코사인 유사도]

TF-IDF로 만든 행렬로 코사인 유사도를 검사하여 뉴스 하나에 대해 유사한 정도가 높은 순으로 각 뉴스들이 대응 할 수 있게 리스트에 저장한다. 사용자가 해당 뉴스에 평점을 매기면 그 뉴스와 유사도가 높은 순으로 뉴스들의 Cluster를 검사해 평점을 매긴 뉴스와 Cluster가 같지 않으면 그 뉴스의 cluster\_id를 recommend DB에 저장한다.

**4.5.3 사용자 기반 추천 시스템**



[시스템 구성 요소 6.2: collaborative filtering]

본 기능은 협업 필터링 (Collaboration Filtering)중 사용자 기반 추천 알고리즘(User based)을 사용하며 개인 맞춤 뉴스 추천 기능을 제공한다. 사용자의 뉴스 평점 데이터가 어느 정도 모인 이후 사용 가능하며 데이터가 없는 초기 사용자는 사용할 수 없다.

사용자는 클릭한 뉴스에 1~5점(1점 단위)까지 평점을 매긴다. rating DB 테이블에 User ID, News ID, Score가 저장된다. Target 사용자와 유사한 k명 사용자들을 찾아 유사 사용자들의 평점을 이용하여 Target 사용자 평점을 예측한다. 예측 평점을 이용하여 사용자에 대한 cluster\_id를 recommend DB에 저장한다.

∎ **피어슨 상관계수**

사용자간의 유사도 측정으로는 피어슨 상관계수를 사용한다. 두 사용자가 공통으로 평점을 매긴 뉴스의 점수를 사용해 측정한다. 유사도는 -1에서 1까지 표현 되며 0.5 이하로 나온다면 유사도가 없다고 판단하여 사용하지 않는다.

∎ **평점 예측**

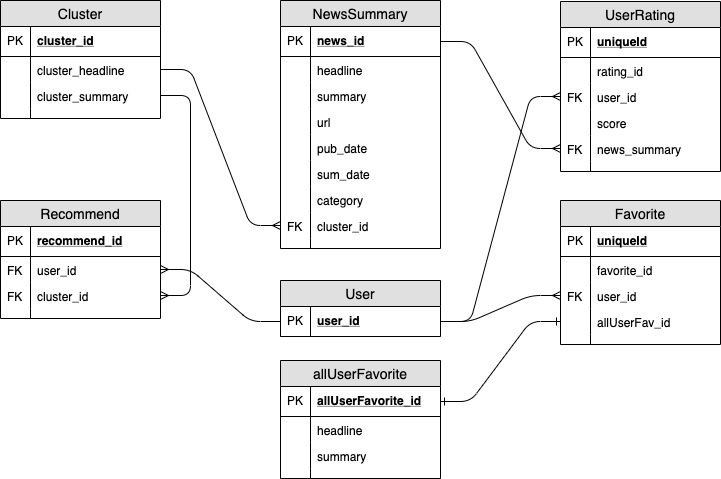
뉴스 추천 받을 사용자를 제외한 모든 사람들의 뉴스평점과 유사도를 곱해 추측 평점을 구한 후 모두 더한 다음 유사도 총합을 나눠 나온 점수로 사용자의

∎ **추천 뉴스 기준**

recommen\_news\_list() 함수에 추천 뉴스 리스트에 들어가는 뉴스들을 정의한다. 뉴스는 사용자가 이전에 보지 않아 평점을 매기지 않은 뉴스이어야 하며 예측 평점이 4점 이상이여야 한다. target 유저가 이미 평가했던 뉴스 외의 뉴스를 추천 받기 위해 target 유저의 평점이 0이면, 가중 평균값을 넣고 그렇지 않으면 0을 넣는다.

## **4.6 Database**

**Linux Server(MySQL)**



[시스템 구성 요소 3.8: Database 설계도]

news crawling과 news clustering, ML processing, news recommending 각 REST API를 통해 받은 data를 수집하여, Database에 갱신한다. Django의 models.py에서 위의 table들을 생성한다.

1. Database에 연결

|  |
| --- |
|  |

1. Database 생성

|  |
| --- |
|  |

models.py에서 table과 field들의 attribute를 정의하고 python manage.py migrate를 통해 Database에 table을 생성한다.

# **5. 기능 설명**

## **5.1 인터페이스 및 기능 설명(여기는 앱의 인터페이스!)**

**5.1.1 로그인 및 회원가입**

**5.1.2 홈 화면**

**5.1.3 카테고리 창**

**5.1.4 프로필 창**

## **5.2 Sequence Diagram**

**5.2.1 News Provide System Sequence Diagram**

스크린샷이(가) 표시된 사진

자동 생성된 설명

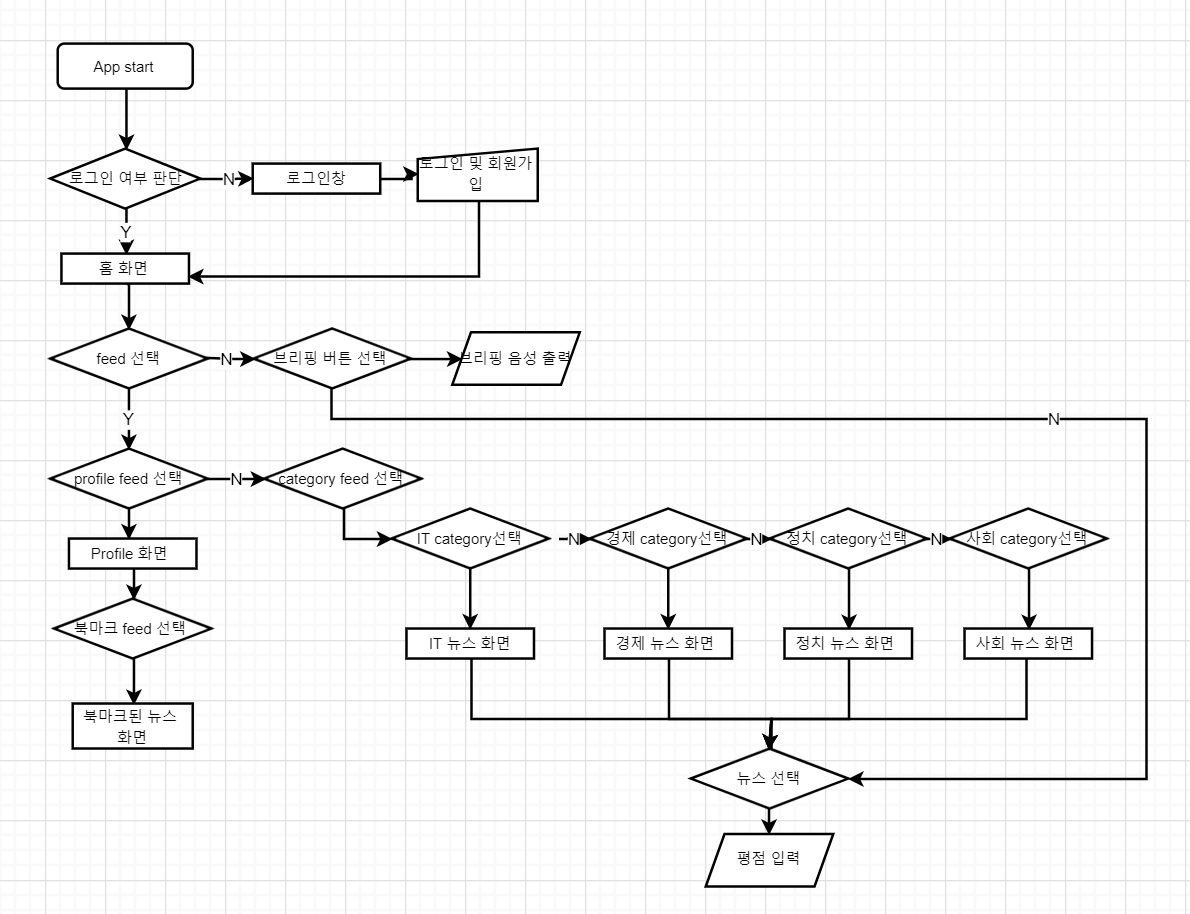
**5.2.1 News Cluster System Sequence Diagram**

**5.2.1 Headline & Multi-summary System Sequence Diagram**

**5.2.1 News Recommend System Sequence Diagram**

## **5.3. Class Diagram**

## **5.4 App flow chart**



# **6. 적용방안 및 기대효과**

**(1) 빠른 시간 내 뉴스 편집 및 제공 가능**

오늘날 생성되는 정보량이 많아지면서 다양한 정보의 뉴스가 범람하게 되었다. 이로 인해 방대한 양의 무분별한 정보가 난무하여 사람들은 정보를 수용하는데 혼란을 겪고 있다. Today News는 인공지능을 기반으로 한 자동화 뉴스 제공 시스템으로 사람의 개입 없이 많은 양의 뉴스를 신속하고 정확하게 편집 (중복 제거 및 요약, 주제별 군집화)하여 사용자에게 제공한다.

**(2) 주제별로 다양한 뉴스 기사를 제공하여 사용자의 뉴스 수용 및 활용 능력의 향상을 도움**

뉴스 전체 내용이 아닌 요약된 뉴스들을 주제별로 보여주며 중복된 뉴스를 제거하고 비슷한 주제를 가진 뉴스들을 클러스터링 해주어 사용자가 같은 주제를 가진 다양한 뉴스를 한 번에 볼 수 있게 한다. 주제의 헤드라인을 추출하여 제공하기 때문에 현재 뉴스 정보의 흐름을 빠르게 파악하는 것을 도우며 사용자가 한 주제에 하나의 뉴스만 접하는 것이 아닌 여러 뉴스를 접하는 것을 도와 주기 때문에 뉴스 수용 및 활용 능력을 높여줄 것으로 기대된다.

**(3) 개인화 맞춤 추천 서비스로 사용자 니즈 충족 및 콘텐츠 소비량의 증가**

Today News는 사용자에게 맞춤형 경험을 제공하고자 사용자의 관심사를 고려한 뉴스 추천 시스템을 구현하였다. 사용자의 뉴스 선호도를 분석하여 추천 뉴스가 일괄적인 것이 아니라 로그인 한 회원마다 달리 나타나 사용자에게 맞춤형 경험을 제공할 수 있을 것으로 보인다. 사용자는 충족시킬 수 있어 콘텐츠 소비량이 증가할 것이다. 개인화 경험을 제공함으로써 사용자의 만족도를 높여 콘텐츠 소비량이 증가할 것으로 보인다.

# **7. 프로젝트 세부 추진 계획 및 일정**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **프로젝트 기간** | **2019.9.9~2019.12.9** | | | | | | | | | | | | |
| **개발 내용** | **프로젝트 기간** | | | | | | | | | | | | |
| **1주** | **2주** | **3주** | **4주** | **5주** | **6주** | **7주** | **8주** | **9주** | **10주** | **11주** | **12주** | **13주** |
| **아이디어 회의 및 계획** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **프로젝트 관련 자료 조사** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **역할 분담 및 핵심 기술 조사** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **사업 제안서 작성** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **요구 사항 정의서 작성** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **서버 및 데이터 베이스 구현** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **앱 구현** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **뉴스 크롤링** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **클러스터링 시스템구현** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **요약 시스템 구성(추상적)** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **요약 시스템 구성(추출적)** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **추천 시스템 구현** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **개별 테스트 및 보안** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **통합 테스트 및 보안** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **최종 발표 및 시연** |  |  |  |  |  |  |  |  |  |  |  |  |  |

# **8. 팀원 담당 업무**

스크린샷이(가) 표시된 사진

자동 생성된 설명