



최종 보고서

Machine Learning을 이용한 인공지능 뉴스 어플리케이션 (Today News)

Ver. 2.0

2019.12.21

한국외국어대학교

정보통신공학과

Team MUD

최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

문서 정보

구분	소속	성 명	날 짜
작성자	한국외국어대학교	박주영(T)	2019.12.21
	한국외국어대학교	김아연	2019.12.21
	한국외국어대학교	김혜원	2019.12.21
	한국외국어대학교	홍승환	2019.12.21
	한국외국어대학교	이산가 비두샤	2019.12.21
	한국외국어대학교	박주영(T)	2019.12.21
검토자			
사용자			
승인자	한국외국어대학교	홍진표	2019.12.23

머리말

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

최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

개정 이력

이력	작성자	개정일자	개정내역
1.0	김아연	2019.11.25	초안 작성
	검토자	박주영	
1.1	홍승환 김아연 검토자	2019.11.27	서비스 개요, 기대효과 작성 박주영
1.2	박주영	2019.11.30	시스템 설명, 기능 설명
	홍승환 김아연 김혜원 이산가비두샤		
	검토자		박주영,
1.3	박주영	2019.12.02	최종 보완 및 수정
	홍승환 김아연 김혜원		
	이산가비두샤		
	검토자		박주영
2.0	김아연	2019.12.21	2차 보완

목차

1.1 목적	8
1.2 참고 문서	8
2. 서비스 개요	9
2.1 서비스 기획 배경	9
2.2 서비스 기능 소개	10
3. 시스템 구성	11
3.1 전체 시스템 구성	11
3.2 Application Server	12
3.2.1 Django REST Framework & REST API	12
3.2.2 REST API 구성	12
3.3 Newsum Server	13
3.4 ML_Headline PC	13
3.5 Database 구성	15
3.5.1 Database 설계도(MySQL)	21
3.5.2 Database 구성 표	21
3.2.1 Database 저장 형태	21
4. 주요 기술 설명 및 적용 방법	16
4.1 Crawling	21
4.1.1 Newsum Crawler 구현	21
4.2 Lexrankr 을 이용한 뉴스 요약	21
4.2.1 News Summary 구현(1 차 요약)	21
4.2.2 News Multi-Summary 구현(2 차 요약)	21
4.3 K-means 를 이용한 News Clustering	21

4.4 Attention Mechanism 을 이용한 뉴스 헤드라인 재생성.....	25
4.4.1 Headline Summary Model 생성 구현.....	25
4.4.2 Headline ml_headline processing 구현.....	26
4.5 Hybrid Filtering 을 이용한 뉴스 추천.....	26
4.5.1 Collaborative Filtering – User based.....	26
4.5.2 Contents Based Filtering – Item based.....	27
4.5.3 Hybrid Filtering 추천 시스템 구현.....	28
4.5.4 추천 시스템 구현 시 고려사항.....	28
4.6 Scheduling 및 Multiprocessing.....	26
4.6.1 News Producing System 에 적용.....	26
4.6.2 News Providing System 에 적용.....	27
4.6.3 News Recommend System 에 적용.....	28
5. NewSum 기능 설명.....	31
5.1 인터페이스 및 기능 설명.....	31
5.1.1 로그인/로그아웃/회원가입 기능.....	31
5.1.2 북마크 기능.....	32
5.1.3 홈 메인 화면.....	32
5.1.4 카테고리 화면.....	33
5.1.5 군집 뉴스 보여주기.....	33
5.1.6 개별 뉴스 보여주기.....	33
5.1.7 평점 기능.....	33
5.1.8 TTS 기능.....	33
5.2 Sequence Diagram.....	36
5.2.1 News Provide System Sequence Diagram.....	34
5.2.2 News Cluster System Sequence Diagram.....	34
5.2.3 Headline & Multi-summary System Sequence Diagram.....	35
5.2.4 News Recommend System Sequence Diagram.....	35
5.3 APP flow chart.....	36

최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

6. 적용방안 및 기대효과	37
7. 프로젝트 세부 추진 계획 및 일정.....	38
8. 팀원 담당 업무.....	39

1. 개요

뉴스 어플리케이션 'Today news'는 인공지능 기술을 이용하여 오늘의 뉴스 요약본을 제공하고, 사용자 별로 뉴스를 추천하여 준다. 본 장에서는 해당 시스템에 대한 목적과 범위, 참고문서를 소개한다.

1.1 목적

본 프로젝트에서는 인공지능 뉴스 편집 기술을 구현하였다. 기존의 사람이 직접 편집하는 방식보다 정보를 신속하게 제공하고, 사용자의 관심사를 파악하여 원하는 뉴스를 개인별로 추천해주는 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 collaborative filtering 기술을 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 고려대학교 산업경영공학과/ 서울대학교 산업공학과	자가 주의 메커니즘을 활용한 seq-to-seq 기반 문서 생성 요약 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 봐야겠다 생각은 하나, 뉴스를 보지 않는 타겟에게 오늘 하루의 뉴스를 브리핑해주는 어플리케이션이다.



[Figure 1] 서비스 기획 배경

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 얻기를 바란다. 사람들은 뉴스를 통해 가장 많은 정보를 얻고 있는데 통계에 따르면 하루에 67개의 매체에서 평균 25,866개의 기사가 생성된다. 이러한 기사들을 전부 읽을 수는 없으며 간략화 하여 접근성을 낮출 필요가 있다고 생각하여 개인 맞춤 요약 뉴스를 제공하는 Today News 어플리케이션을 기획하고 구현하게 되었다.

2.2 서비스 기능 소개

본 장에서는 Newsum이 제공하는 주요 기능들을 간단하게 소개하고 세부 설명은 '5장 기능 설명'에서 이어 한다.

(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 피드와 카테고리 피드는 추천 알고리즘을 통해 사용자의 취향에 맞는 뉴스들로 채워지게 된다. 사용자가 뉴스에 평점을 매기면 그 점수를 기반으로 추천 알고리즘이 돌아간다.

(4) 북마크 기능

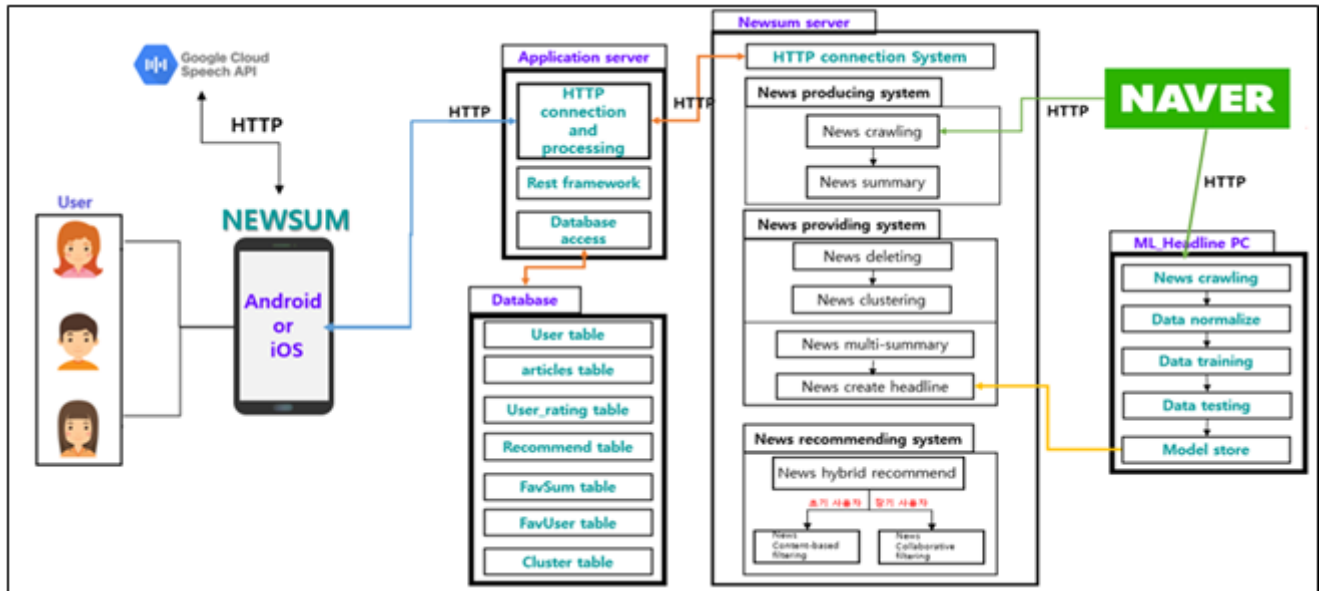
사용자는 나중에 다시 읽고 싶은 기사를 북마크(스크랩)기능을 통해 저장해 둘 수 있다. 스크랩한 뉴스는 프로필창에서 확인 할 수 있다.

(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 구성



[Figure 2] NewSum 시스템 구성도

NewSum System 구성요소				
user	App(android or iOS)	Application Server	Database	ML_Headline PC
user는 NewSum System Service를 받는 대상이다.	Application Server에서 운용되는 application을 사용할 수 있는 역할을 담당한다.	application server로 App과 Database와 Newsum server 사이에 존재하여 이들 사이에서 정보를 제공하고 제공받는다.	application server에서 운용되는 database로 Newsum Server와 App 사이에서 data language를 이용해 정보를 주고 받는다.	Naver에서 제공하는 뉴스들을 토대로 dataset을 만들고 train한 모델을 Newsum server로 headline을 보낸다.

Today News는 크게 3개의 서버로 구성된다. 클라이언트와 서버(Newsum server)간 중계역할을 하는 REST API 서버인 Application server, Newsum의 전체적인 서비스 및 기능을 제공하는 Newsum server, 딥 러닝 모델을 생성하는 ml_headline PC가 있다.

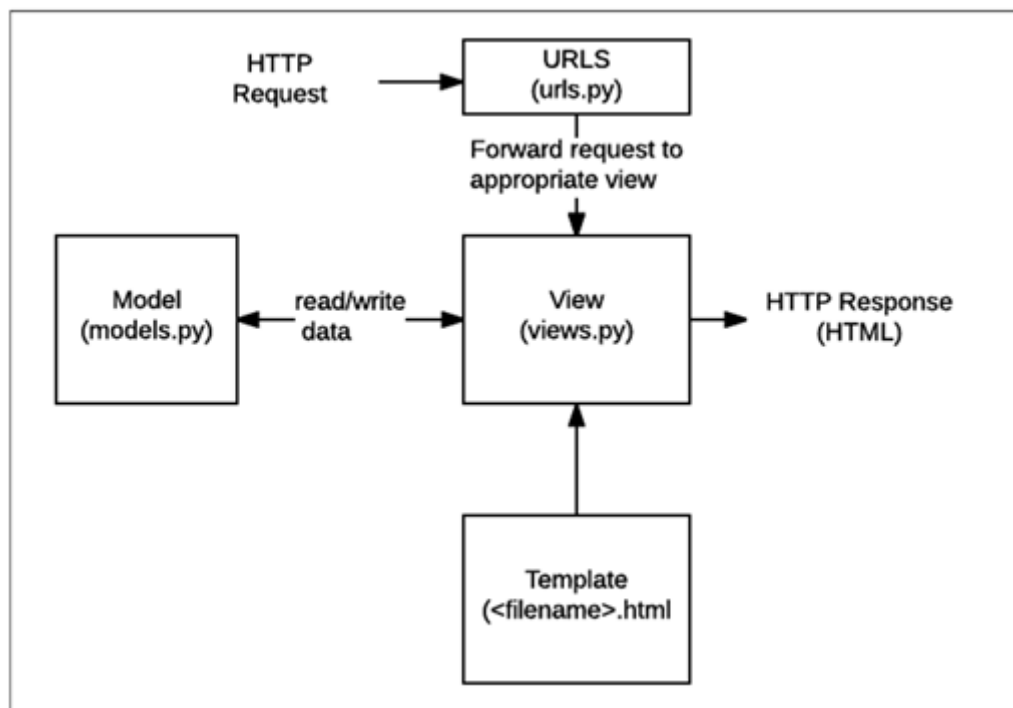
최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

Newsun APP은 정보를 요청 할 때, Django rest frame work를 사용하여 Application server와 통신한다. Newsum server는 News Producing System, News providing System(이 시스템은 News clustering System, Headline & Multi-summary System 2개의 서브시스템으로 구성된다.), News Recommend System 총 4개의 시스템으로 구성되어 있으며 각각의 System을 거쳐 생성된 데이터는 Application server를 거쳐 DB에 저장된다.

3.2 Application Server

3.2.1 Django REST Framework & REST API

Django는 python을 활용해 쉽게 웹서비스를 만들 수 있도록 하는 프레임 워크이다. Django Rest Framework는 Django를 활용하여 REST API를 구현하는데 필요한 다양한 기능들을 제공한다. REST는 웹의 장점과 HTTP의 우수성을 잘 활용하고 있는 아키텍처이며 Resource, Verb, Representation으로 구성된 아키텍처이다. HTTP URI를 통해 자원을 명시하고 HTTP Method(POST, GET, PUT, DELETE)를 통해 자원의 CRUD 연산을 적용한다. 따라서 HTTP 표준 프로토콜을 따르는 모든 플랫폼에서 사용가능하다. REST API를 사용하면 프론트엔드와 백엔드를 완전히 분리할 수 있으며, 코드의 재사용성을 높일 수 있다.



3.2.2 REST API 구성

Django 웹 어플리케이션은 위의 그림과 같이 분류된 파일들에 대해 일련의 단계를 수행하는 코드로 구성되어 있다. MVC아키텍처를 따르고 있다.

3.3 Newsum Server

Newsum server는 News Producing System, News providing System(News clustering System, Headline & Multi-summary System), News Recommend System 총 4개의 시스템으로 구성된다. 아래는 각 시스템들에 대한 간단한 설명을 표로 보였으며 시스템을 구현하는데 사용된 주요 기술과 그 적용 방법은 4장에서 기술한다.

System	설명
News Produce System	News Produce System은 실시간으로 뉴스를 크롤링 한 후 본문을 요약하여 DB에 저장하는 시스템이다. 즉, 이용자에게 제공할 날개 뉴스를 생성하는 시스템이다. News Crawling을 통해 네이버 뉴스 속보의 정치/경제/IT/사회란에서 실시간으로 뉴스를 크롤링 해온다. 크롤링 해 온 뉴스들은 News Summary를 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 linux server를 거쳐 DB에 저장된다.
News Cluster System	News Cluster System의 News clustering은 요약된 후 DB에 저장된 뉴스들을 군집화 한다. 군집이 완료된 News delete를 통해 지워진다. 클러스터링이 완료된 뉴스들은 DB의 Cluster table에 저장된다.
Headline & Multi-summary System	Headline & Multi-summary에서 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.
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 번갈아 사용하게 된다.

3.4 ML_Headline PC

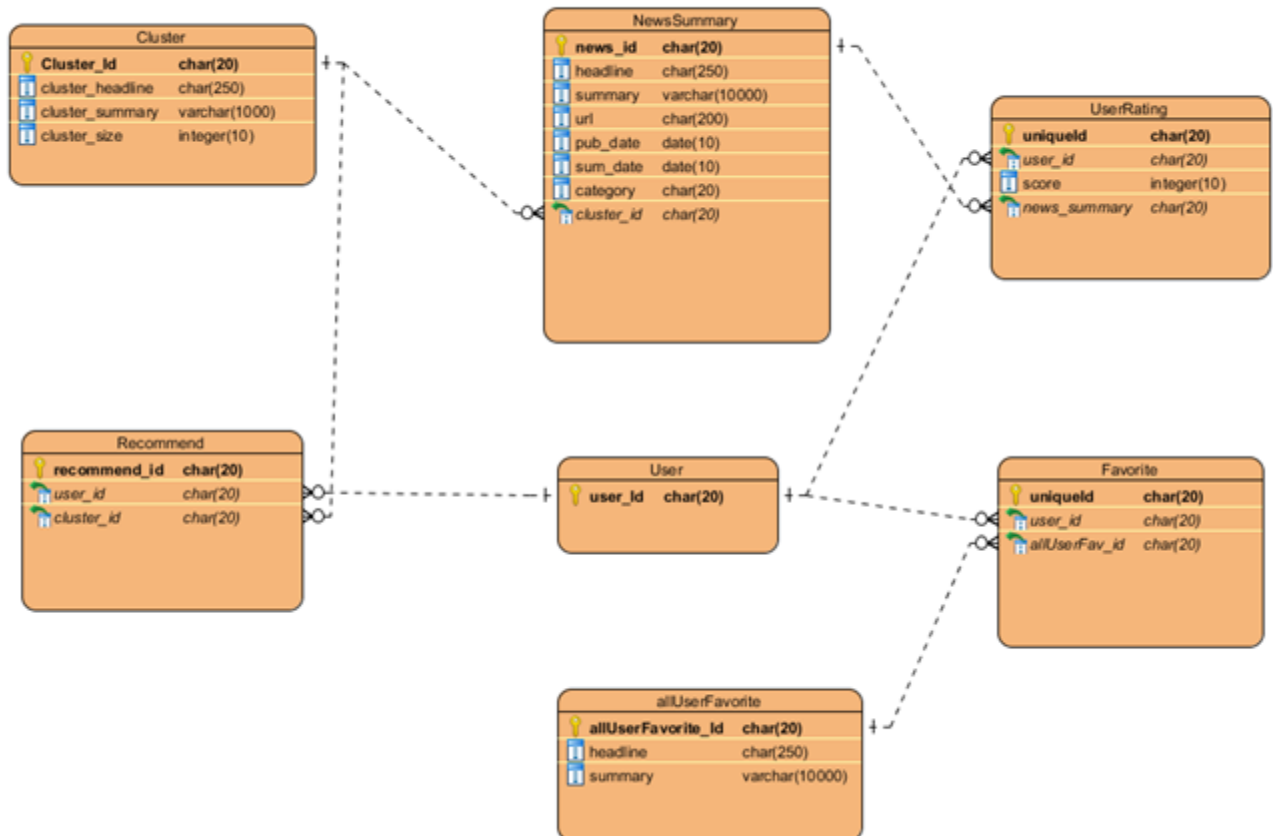
ML_Headline PC는 머신러닝 컴퓨터로 군집 뉴스의 Headline을 만드는 모델을 생성한다. 모델 학습을 실행시키고 모델을 자주 업그레이드 하기 위해 딥 러닝 모델을 처리할 정도의 GPU사양을 가진 컴퓨터를

최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

사용하였다. NewSum만의 딥러닝 시스템 ML_Headline PC를 두어 비용 효율성을 높였다.

3.5 Database 구성

3.3.1 Database 설계도(MySQL)



[Figure 12] 시스템 구성 요소 4.6: 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.

3.3.2 Database 구성 표

Table Name	Attribute	
Newssummary	News_id(PK)	Crawling한 뉴스 고유의 uuid4부여
	Headline	Crawling한 뉴스의 제목
	Summary	Crawling한 뉴스의 본문을 lexrankr 알고리즘을 이용하여 본문 요약 content
	url	Crawling한 뉴스의 URL주소
	Pub_date	Crawling한 뉴스의 생성 날짜[뉴스기사에 포함 되어있는 날짜를 의미함]
	Sum_date	Crawling한 뉴스의 본문 요약 완료된 날짜
	Category	Crawling한 뉴스의 카테고리 이름[economy, politics, IT/science, society]
	Cluster_id(FK)	Cluster table의 cluster_id값을 참조한다(FK) [clustering이 실행되는 시간은 정해져 있기 때문에 cluster되기전에 저장된 news들의 값은 default값으로 부여 받는다] [또한 cluster에 참여하지 못한 뉴스들도 default값으로 부여 받는다]
Cluster	Cluster_id(PK)	개별 뉴스들이 cluster(군집화)되어 군집화된 뉴스들에게 고유의 ID값(uuid4)를 부여한다
	Cluster_headline	Cluster 다중 문서 요약된 content를 통해 ml_headline_model에 input값으로 넣어 output으로 headline을 얻는다. 얻은 headline은 cluster_headline에 저장된다.
	Cluster_summary	Cluster된 뉴스들을 다중 문서요약에 특화된 lexrankr를 통해 다중 문서 요약 content를 cluster_summary에 저장한다

최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

Recommend	Recommend_id(PK)	Recommending system의 output으로 cluster의 id를 저장한다.
	User_id(FK)	User table의 user_id를 참조한다
	Cluster_id(FK)	Cluster의 cluster_id를 참조한다.
User	User_id	FirebaseDatabase에서 받은 user_id에 대한 고유의 id값(uuid4)이다
allUserFavorite	AllUserFavorite_id(PK)	모든 회원들이 스크랩한 뉴스들의 Newssummary의 news_id값을 참조하지 않고 복사하여 저장한다[이 AllUserFavorite_id _id의 값은 영원히 지워지지 않는 data이다]
	Headline	모든 회원들이 스크랩한 뉴스들의 Newssummary의 headline을 참조하지 않고 복사하여 저장한다[이 headline의 값은 영원히 지워지지 않는 data이다]
	summary	모든 회원들이 스크랩한 뉴스들의 Newssummary의 summary를 참조하지 않고 복사하여 저장한다[이 summary의 값은 영원히 지워지지 않는 data이다]
Favorite	uniqueId(PK)	User가 뉴스를 스크랩할 시 부여 받는 고유한 id이다.
	allUserFav_id(FK)	allUserFavorite table의 allUserFav_id를 참조한다.
	User_id(FK)	User table의 User_id값을 참조한다.
UserRating	uniqueId(PK)	User가 개별 각각 뉴스에 대해 평점을 매길 시 부여 받는 고유한 id이다
	User_id(FK)	User table의 usr_id를 참조하고 있다.
	Score	회원이 개별 각각 뉴스에 대해 1~5점사이의 점수를 부여하면, 그 값이 score에 저장된다[이 score는 recommending system에서 사용된다]

최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

	News_summary(FK)	Newssummary table의 news_id를 참조하고 있다
--	------------------	-------------------------------------

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

최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

3.3.3 Database 저장 형태

Table Name	저장 형태
newssummary	
Users	
Cluster	

최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

UserRating	<pre>{ "rating_id": "055c8196-9b47-4289-8592-a586a7625696", "score": 5, "user_id": "pFbDe7m550YtXhgcar18FgEFhYr1", "news_summary": "b47b1456-d816-4f30-8d5c-9d5846a79bb9" }, { "rating_id": "4f71ee52-e260-4382-8b30-5e9847765b6b", "score": 2, "user_id": "R7TdA4s1D105QwN71T3Llad6e2V2", "news_summary": "6fd0f744-5758-45c1-94c5-84a69b893429" }, { "rating_id": "6d501bd3-5078-4e36-ab70-401a2bf8e87f", "score": 4, "user_id": "Ghwo1CGPbOdxump04uDYXkHGL1H2", "news_summary": "6fd0f744-5758-45c1-94c5-84a69b893429" }, }</pre>
Recommend	<pre>{ "recommend_id": "862c007f-d49d-4265-8bce-814f7d2e18df", "user_id": "pFbDe7m550YtXhgcar18FgEFhYr1", "cluster_id": "58d0aa77-1664-4bed-9f45-0e28e8b11171" }, { "recommend_id": "b77947da-6355-48da-b2b1-7c0a7f729e10", "user_id": "pFbDe7m550YtXhgcar18FgEFhYr1", "cluster_id": "dbd18b3f-6ae2-4a2c-820e-5c815cd2662c" }, { "recommend_id": "e97eb3b4-5876-4bc1-8079-2897b25a0aac", "user_id": "pFbDe7m550YtXhgcar18FgEFhYr1", "cluster_id": "3e93d95e-543a-49e7-9a63-ae604fb535a8" } }</pre>

4. 주요 기술 설명 및 적용 방법

4.1 Crawling

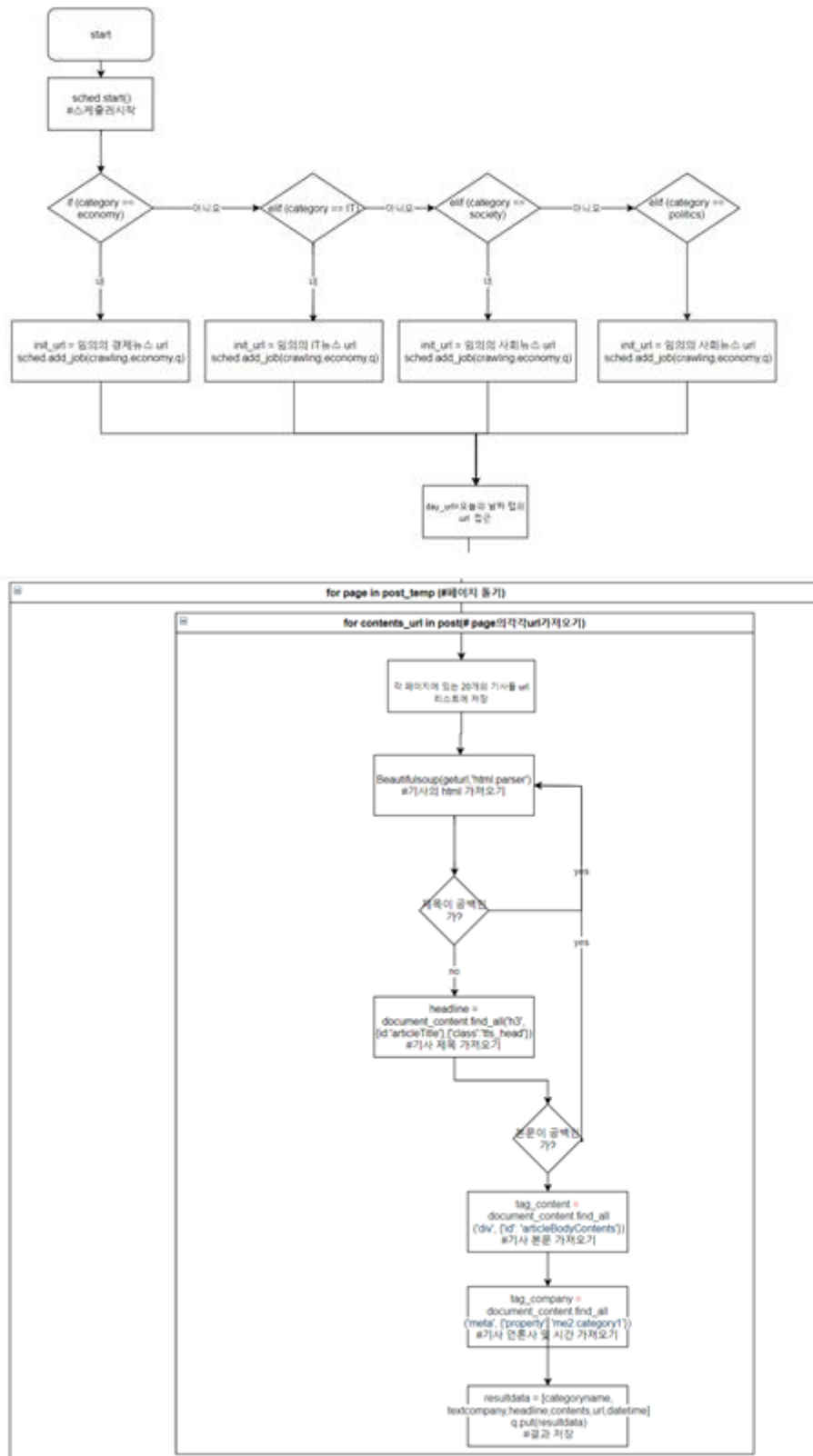
크롤링이란 컴퓨터 소프트웨어 기술로 웹 사이트에서 원하는 정보를 추출하는 것을 의미하며, 크롤러는 인터넷의 웹페이지를 방문해서 수많은 자료들을 수집하는 일을 하는 프로그램을 말한다. 사람이 모든 웹 사이트를 방문하며 데이터를 수집하는데 한계가 있기 때문에 크롤러는 사람이 일일이 하는 대신 이를 자동으로 수행한다.

Newsum은 실시간으로 뉴스 정보를 수집하는 자체 크롤러를 구현하며, 이는 News producing System의 클래스로 정의된다. 아래의 4.1.1장에서 Newsum Crawler 설계 방법에 대해 자세히 기술한다.

4.1.2 Newsum Crawler 구현

News Crawler 구현

최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)





최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

- ① 네이버 뉴스의 속보란에서 경제, 사회, 정치, IT 4개의 카테고리 내 오늘의 뉴스를 수집한다.
- ② 스케줄러를 이용하여 일정 시간 간격으로 계속하여 crawler함수를 실행시킨다. 경제: 3분, 사회: 3분, 정치: 7분, IT:15분 간격으로 크롤링을 실행한다.

```
##경제 :5분/ 사회:3분/ 정치:7분/IT:15분,
if "economy" in category:
    old.append(category["economy"])
    sched.add_job(Crawler.crawling, 'cron', minute="5", id="test_1",args=["economy", q]) # argssms 배열로 넣어주어야한다.
if "IT_science" in category:
    old.append(category["IT_science"])
    sched.add_job(Crawler.crawling, 'cron', minute="15", id="test_2", args=["IT_science", q]) # argssms 배열로 넣어주어야한다.
if "society" in category:
    old.append(category["society"],)
    sched.add_job(Crawler.crawling, 'cron', minute="3", id="test_3", args=["society", q]) # argssms 배열로 넣어주어야한다.
if "politics" in category:
    old.append(category["politics"])
    sched.add_job(Crawler.crawling, 'cron', minute="7", id="test_4", args=["politics", q]) # argssms 배열로 넣어주어야한다.
```

- 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.
- ⑤ main문의 category딕셔너리에는 초기값 url을 저장할 수 있으며, 크롤링의 시작점으로 정하고 싶은 기사의 url을 개발자가 직접 지정하여 준다.

```
old={}
category={"economy":'https://news.naver.com/main/read.nhn?mode=LSD&mid=sec&sid1=101&oid=005&aid=0001260441',"IT_science":'https://news.
,society':'https://news.naver.com/main/read.nhn?mode=LSD&mid=sec&sid1=102&oid=005&aid=0001260447',"politics":'https://news.n
```

4.2 Lexrank를 이용한 뉴스 요약

Lexrank는 Textrank기법에 그래프 클러스터링과 새로운 유사도 함수를 적용한 알고리즘으로 대용량 문서 요약과 다중 문서 요약에 적용 가능하다. tf-idf벡터 기반으로 미리 클러스터링 가능하지만 lexrank는 그래프를 구성 후 그래프 클러스터링 알고리즘을 적용한다. NewSum은 추출적 요약 기법인 Lexrank 알고리즘을 이용하여 기사 원문에서 3줄 요약문을 만들어 낸다. (설문조사에 의하면 모바일에서 이용자가 좋은 뉴스라고 생각하는 뉴스의 길이는 3줄-5줄이다.)

Newsum은 Lexrank를 이용하여 2번의 요약을 한다. 각각의 요약을 News Summary(1차 요약), News multi-Summary(2차 요약)이라 부른다. 1차 요약은 News Producing system에서 정의되며 크롤링 후 가공이 되지 않은 원본 뉴스들의 본문을 3줄로 요약하여 날개 뉴스의 요약본을 생성하는 것이다.

2차 요약은 News Providing System에서 정의되며 날개 뉴스들을 클러스터링 한 군집 뉴스들의 3줄 요약

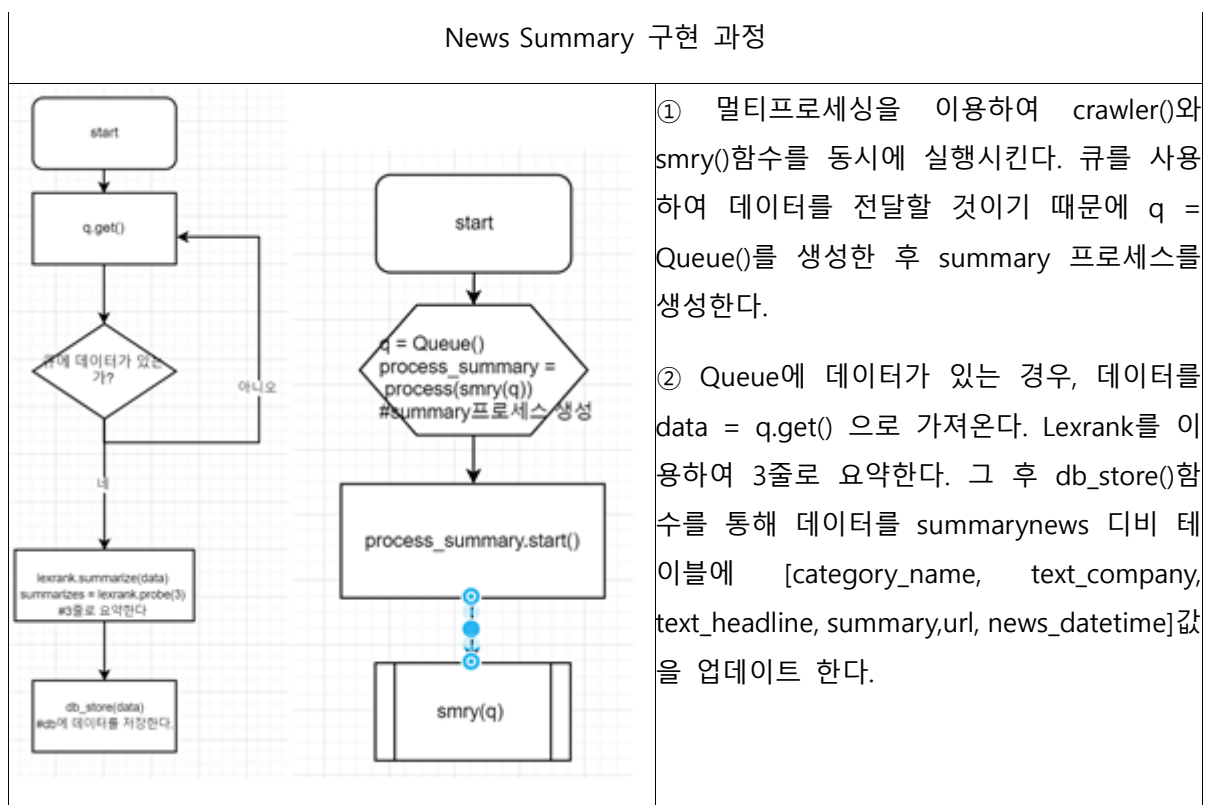
최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

본을 생성하는 것이다. 군집 내 날개 뉴스 4개를 랜덤으로 선택하여 요약하는데 이 때(다중 문서 요약을 할 때) Lexrank이 textrank나 그 외 다른 요약 알고리즘에 비해 성능이 뛰어난 것을 밑의 결과를 통해 알 수 있다.

방법 - 데이터셋	F1	ROUGE-1	ROUGE-2
기준 - 짧은 문서	50.4	61.1	51.5
제안 - 짧은 문서	59.2	75.8	67.0
기준 - 긴 문서	36.9	51.8	37.6
제안 - 긴 문서	47.6	68.0	56.8

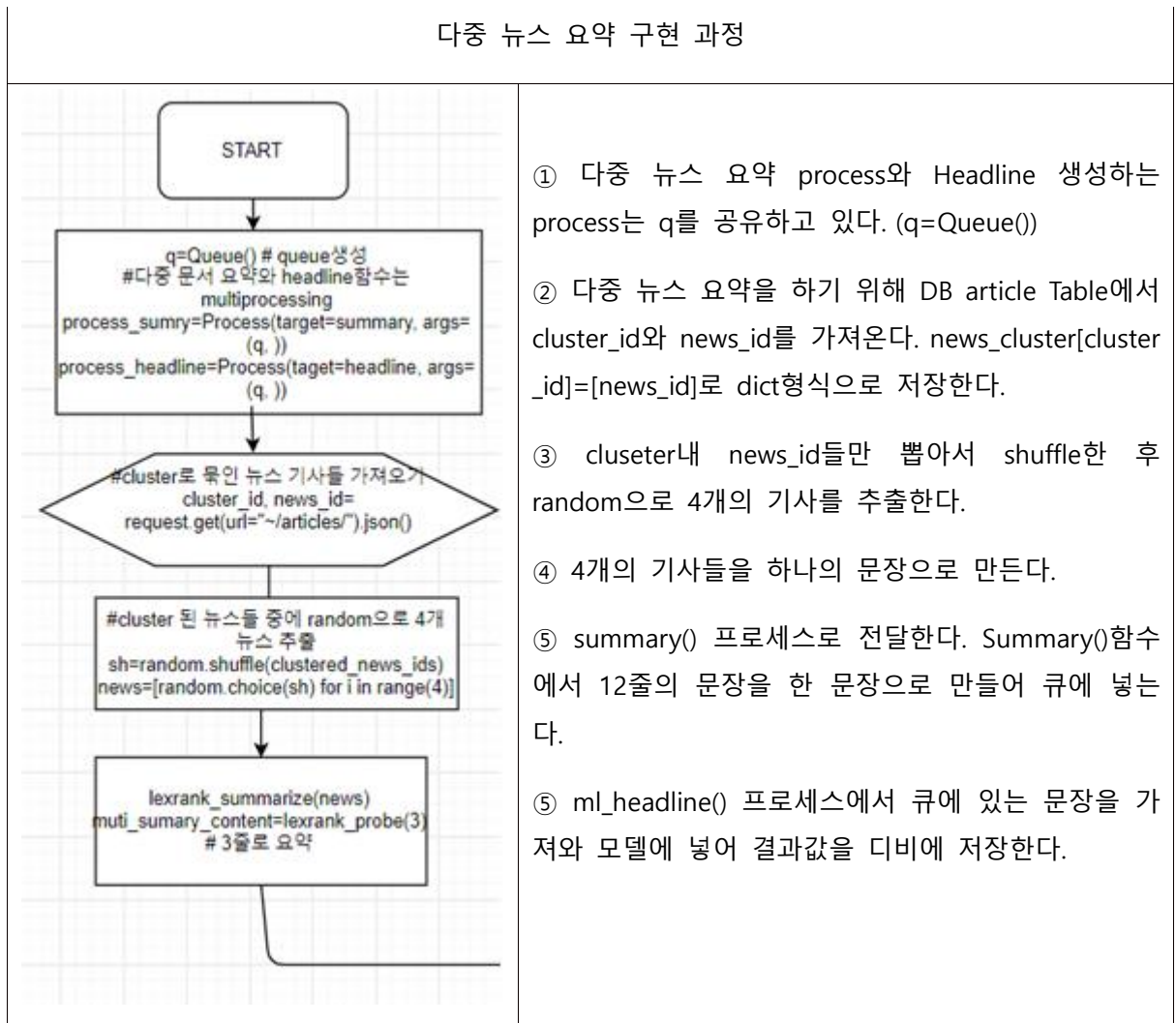
위 도표에서 기준은 기존에 존재하는 요약 알고리즘 textrank와 제안하는 lexrank 알고리즘 둘 중에서 어느 것이 더 한국어 다중문서 요약에 적합한지 보여지고 있다. 결과적으로 lexrank 알고리즘이 다중 문서요약에 적합하다는 것을 알 수 있었고 따라서 Newsum은 Lexrank 알고리즘을 통해 단일 문서 요약 뿐만 아니라 다중 문서요약도 진행하였다.

4.2.1 News Summary 구현 (1차 요약)



4.2.2 News Multi-Summary 구현 (2차 요약)

다중 뉴스 요약 구현 과정



4.3 K-means Clustering을 이용한 News Clustering

k-평균 알고리즘(K-means algorithm)은 주어진 데이터를 k개의 클러스터로 묶는 알고리즘으로, 각 클러스터와 거리 차이의 분산을 최소화하는 방식으로 동작한다. 이 알고리즘은 label이 없는 데이터의 입력을 받아 각 데이터에 label을 할당하여 군집을 수행한다. 개념이 매우 간단하며 실행속도가 빠르고 특정한 데이터에는 좋은 성능을 보인다. 하지만 k를 개발자가 직접 설정해주어야 하며 최적의 k를 구하는 데는 많은 시간이 걸린다는 단점이 있다. NewSum에서는 최적의 k를 구하는 시간을 줄여 성능을 높이는 법을 실험하고 적용해보았다.

4.3.1 Clustering 사용 과정

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. 묶는다는 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 K값 찾기

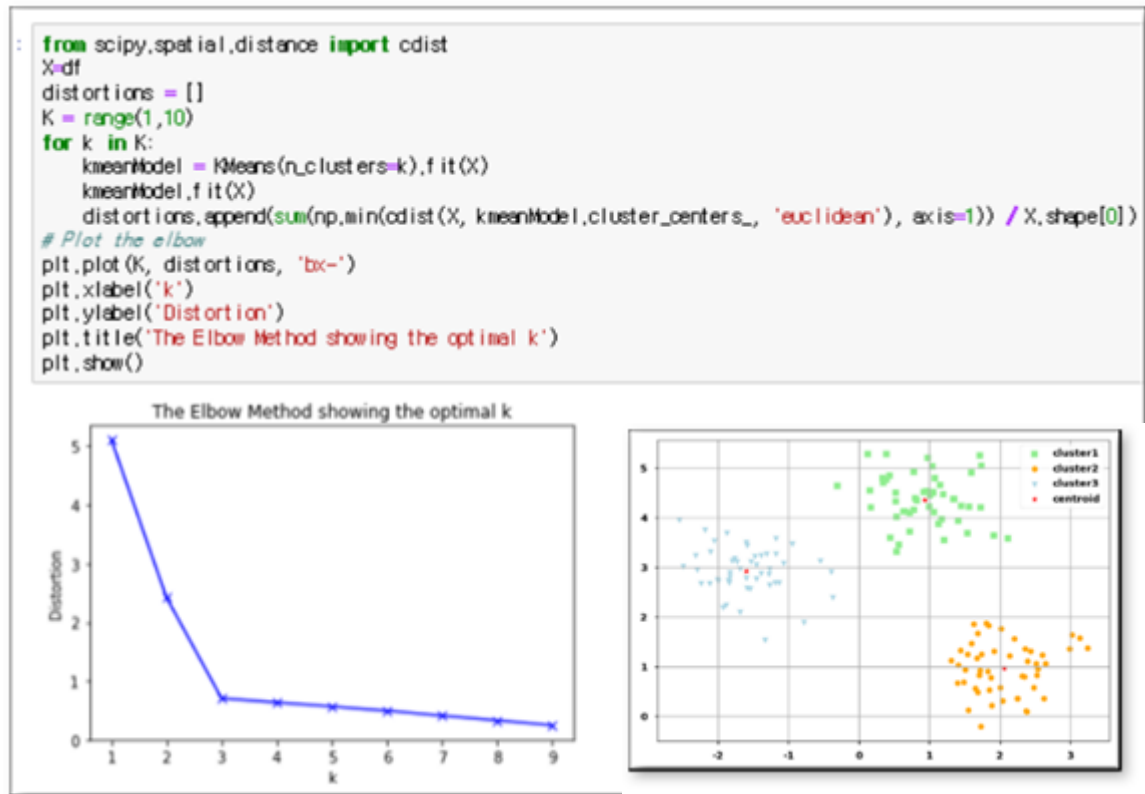
최적의 K값, 즉 cluster의 개수는 K-means 모듈을 사용하여 K-means를 선언 시에 꼭 넣어 줘야 하는 값이다. 이는 k-mean기법의 단점이기도 하다. 하지만 아래 그림과 같이 최적의 K값을 찾아 넣어 주면 해결 될 수 있는 단점이다.

그림에서 오른쪽 그래프는 data들을 벡터화 시켰을 때 x축, y축을 기준으로 어떻게 분포 되어 있는지를 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 선언 시, n_clusters

```
km = KMeans(n_clusters=3, init=init_centroid, random_state=0)
```

KMeans
에 대입



[Figure 7] 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.

```
km = KMeans(n_clusters=3, init=init_centroid, random_state=0)
```

▷ randomly select `init_centroid = 'random'` centroid

▷ randomly select `#init_centroid = 'k-means++'` centroid

: 임의의 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 data 를 이용했을 땐 random 과 k-means++은 차이가 없었다.

문서에서도 결과값의 성능에 차이가 없다면 random 을 쓸 예정이다.

4.3.4 전처리 과정



▷ 형태소 분석 처리

```
new_post = "이미징 데이터베이스는 저장한다."
new_post_tokens = ' '.join(pos_tagger.morphs(new_post))
```

▷ 불 용어 제거

```
##preprocessing(특수문자제거)

import re
#re.sub() 함수는 문자열에서 매치된 텍스트를 다른 텍스트로 치환할 때 사용한다.
def preprocessing(sentence):
    sentence = re.sub('2028', '***', sentence)
    return sentence
```

▷ 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 TfidfTransformer를 사용해 tf-idf로 변환)

```
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer

class StemmedCountVector(TfidfVectorizer):
    def build_analyzer(self):
        analyzer = super(StemmedCountVector, self).build_analyzer()
        return lambda doc: (english_stemmer.stem(w) for w in analyzer(doc))

vectorizer = StemmedCountVector(min_df = 10, max_df=0.5, stop_words='english', decode_error='ignore')
vectorized = vectorizer.fit_transform(train_data.data)
```

▶ CountVectorizer :

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

문서를 토큰 리스트로 변환한다.

각 문서에서 토큰의 출현 빈도를 센다.

각 문서를 BOW 인코딩 벡터로 변환한다.

```
# CountVectorizer로 토큰화
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(content)

# l2 정규화
X = normalize(X)
```

4.4.4 K_means 적용

KMeans module을 통해 아래와 같이 K개의 n_cluster로 분류할 수 있다.



최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

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

이후 앞서 걸러진 전처리된 dataset을 대입시켜 K 개의 군집화를 진행한다.

고려해야 할 점은 앞으로 새로운 뉴스 군집 설정할 때이다.

```
# 초기 군집 개수를 설정합니다.
num_clusters = 20

from sklearn.cluster import KMeans
km = KMeans(n_clusters = num_clusters, init='random', n_init=1, verbose=1)
km.fit(vectorized)

#새로운 뉴스 군집 설정

# 새로운 게시물
new_post = "Disk drive problems. Hi, I have a problem with my hard disk. ㄹ
After 1 year it is working only sporadically now. ㄹ
I tried to format it, but now it doesn't boot any more. ㄹ
Any ideas? Thanks."

# vectorization 시킨 후
new_post_vec = vectorizer.transform([new_post])

# KMeans의 predict 함수에 대입합니다.
new_post_label = km.predict(new_post_vec)[0]

similar_indices_array = (km.labels_ == new_post_label)
similar_indices = similar_indices_array.nonzero()[0]

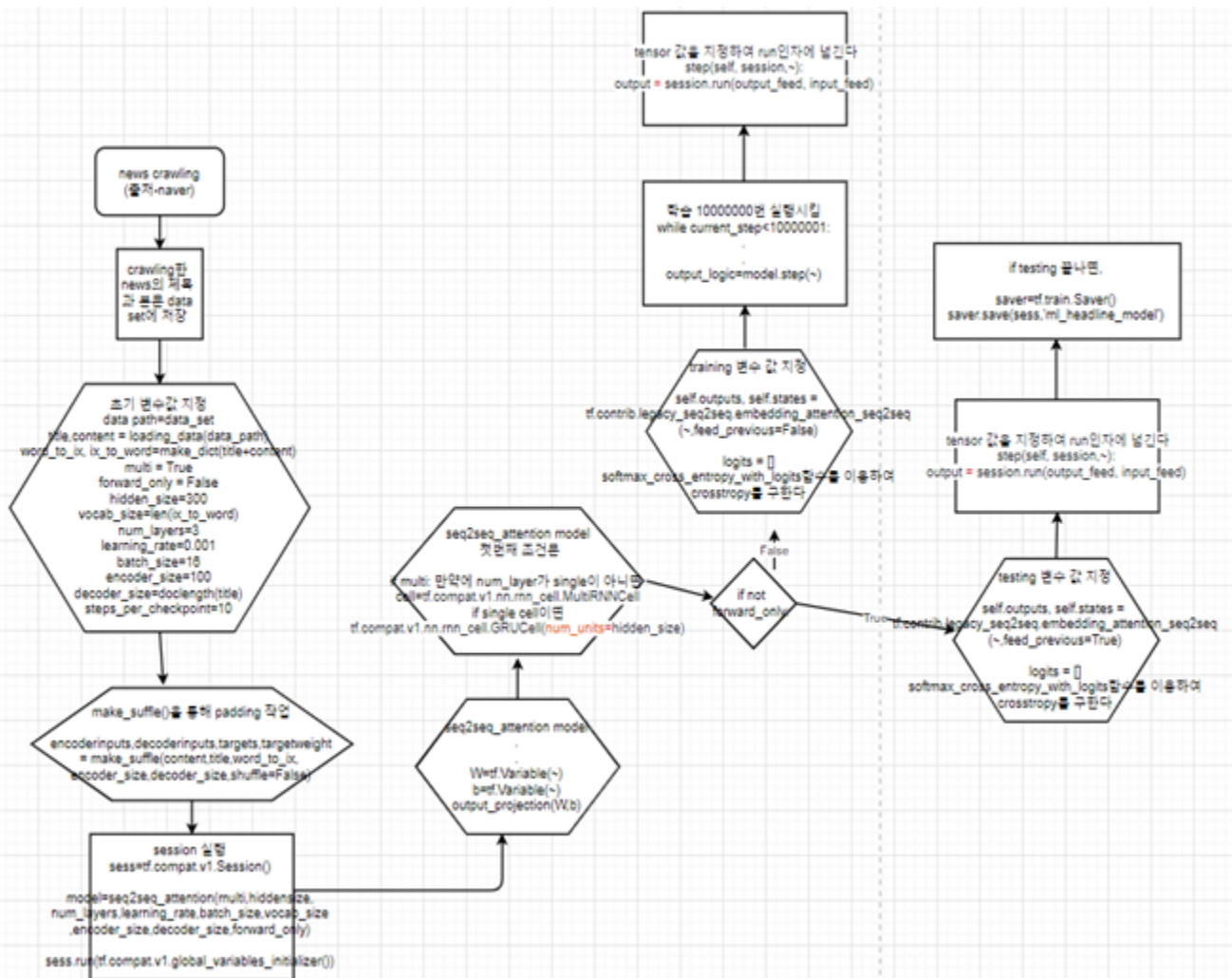
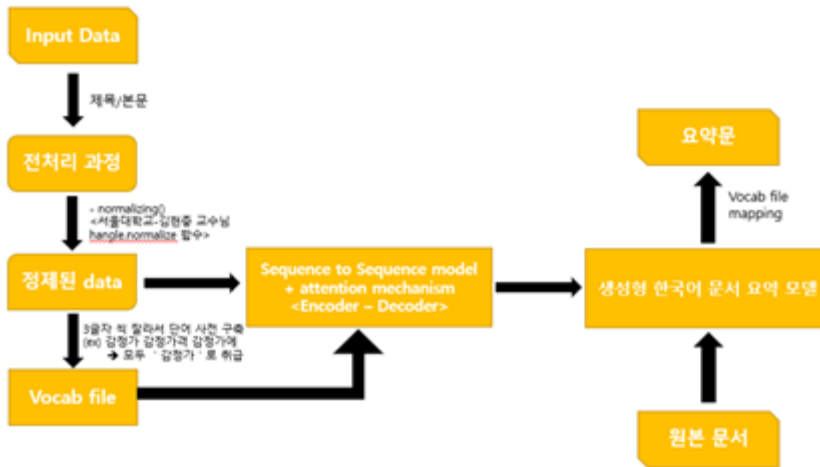
similar = []
for i in similar_indices:
    dist = np.linalg.norm(new_post_vec.toarray() - vectorized[i].toarray())
    similar.append((dist, train_data.data[i]))
# dist가 작은 순서대로 정렬됩니다.
similar = sorted(similar)
```

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.

최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

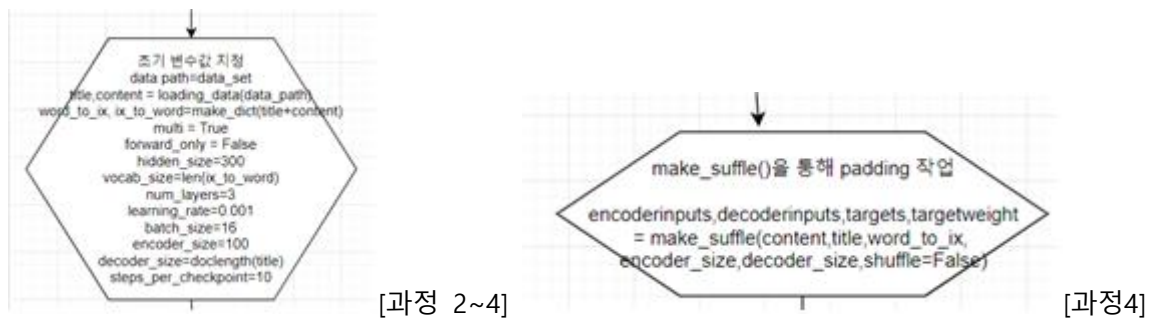
4.4 Attention Mechanism을 이용한 뉴스 헤드라인 재생성



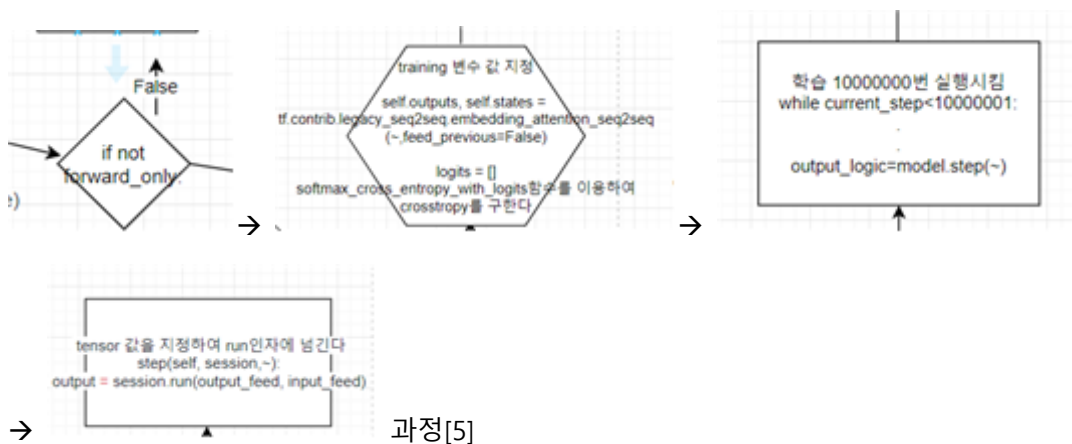
4.4.1 Headline summary Model 생성진행순서

최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

1. ml_headline server에서 news crawling을 통하여 dataset을 얻는다
 - news crawling의 기간은 한달로 늘려 dataset을 넉넉히 얻는다[얻는 정보는 title과 content이다]
 - 이때, 사용하는 news 기사 출처는 naver 뉴스이다[카테고리-정치,사회.IT/과학,경제]
2. 충분한 Dataset을 얻은 후, main.py에서 주어진 data_path에서 title과 content를 뽑아 normalize한다.
3. normalize된 title과 content를 make_dict()함수를 통해 정수 인코딩한다[return 값으로 word->index, index->word 딕셔너리 얻음]
4. seq2seq model을 train과 test하기위한 초기값 변수 설정 후 padding작업까지 완료한다.



5. session 시작 후, step<100000될 때 까지 training시킨다.

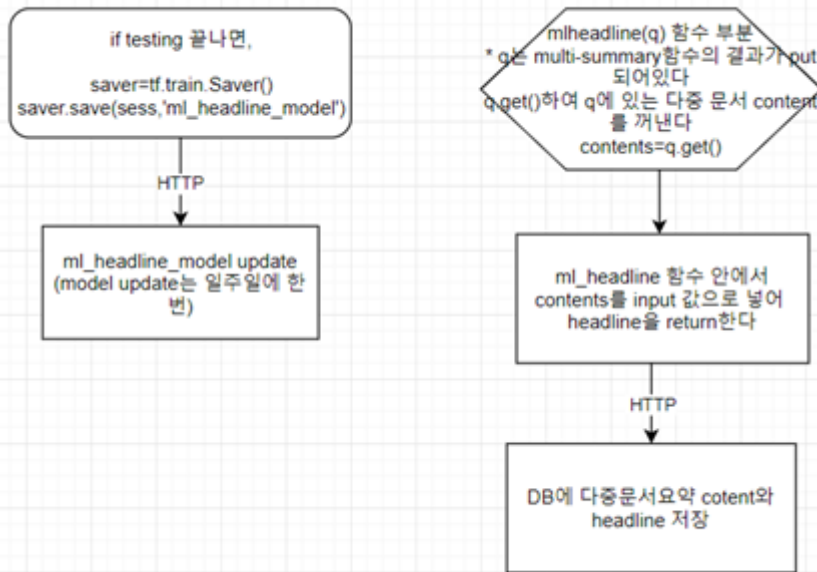


6. training 끝나면 testing 시작한다.



7. training까지 마치면 model 저장한다.

최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)



구현

진행 순서	
Model manual update	ml_headline server에서 model 생성되면 Today server와 HTTP통신 하여 Today server의 ml_headline 모델을 manual update한다.
Cluster의 headline 추출	<p>1. 다중 문서 요약된 내용이 queue에 q.put()하면, ml_headline은 queue에서 q.get()하여 다중 문서 요약 content를 가져와 ml_headline_model()에 input 값으로 넣어 output 값으로 headline을 도출한다.</p> <p>2. 다중 문서 요약된 content와 headline을 DB[Cluster table의 headline과 content를 저장한다]에 접근하여 json형태로 전송한다. 이때, DB에 접근하기 위해서 Today server는 Linux server에 HTTP통신한다.</p>

4.4.2 headline ml_headline_processing 구현



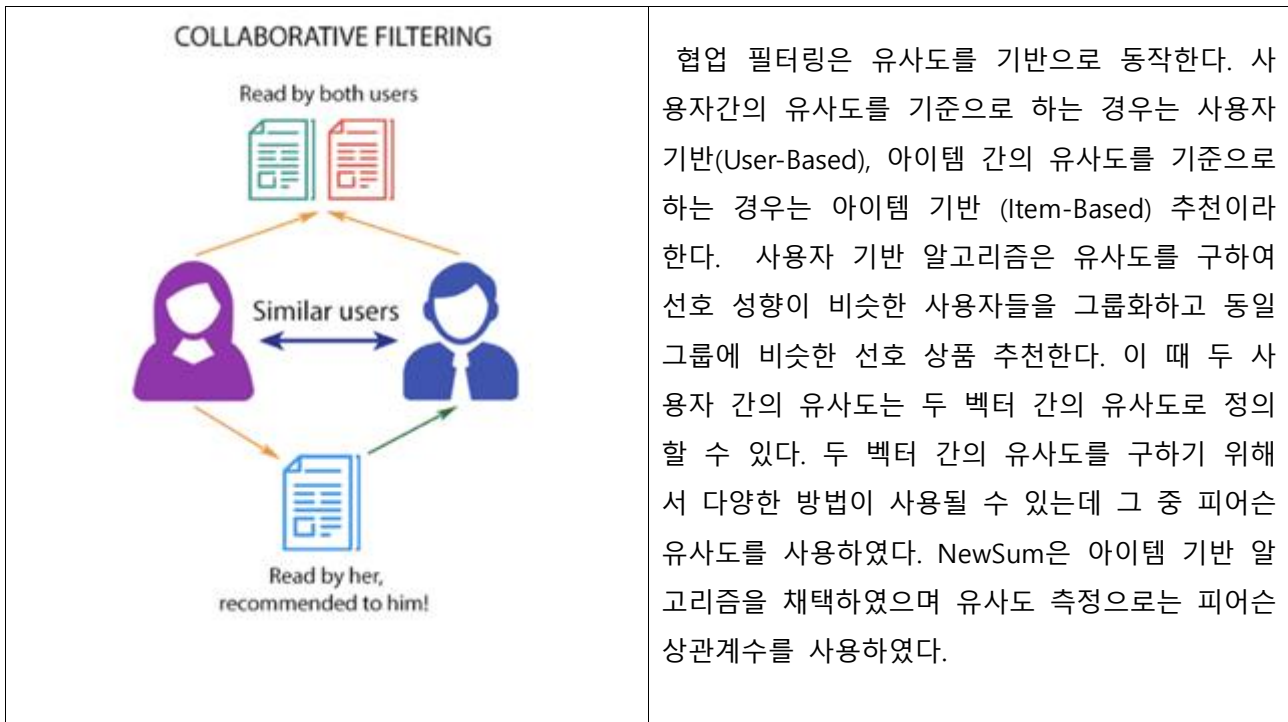
4.5 Hybrid Filtering을 이용한 뉴스 추천



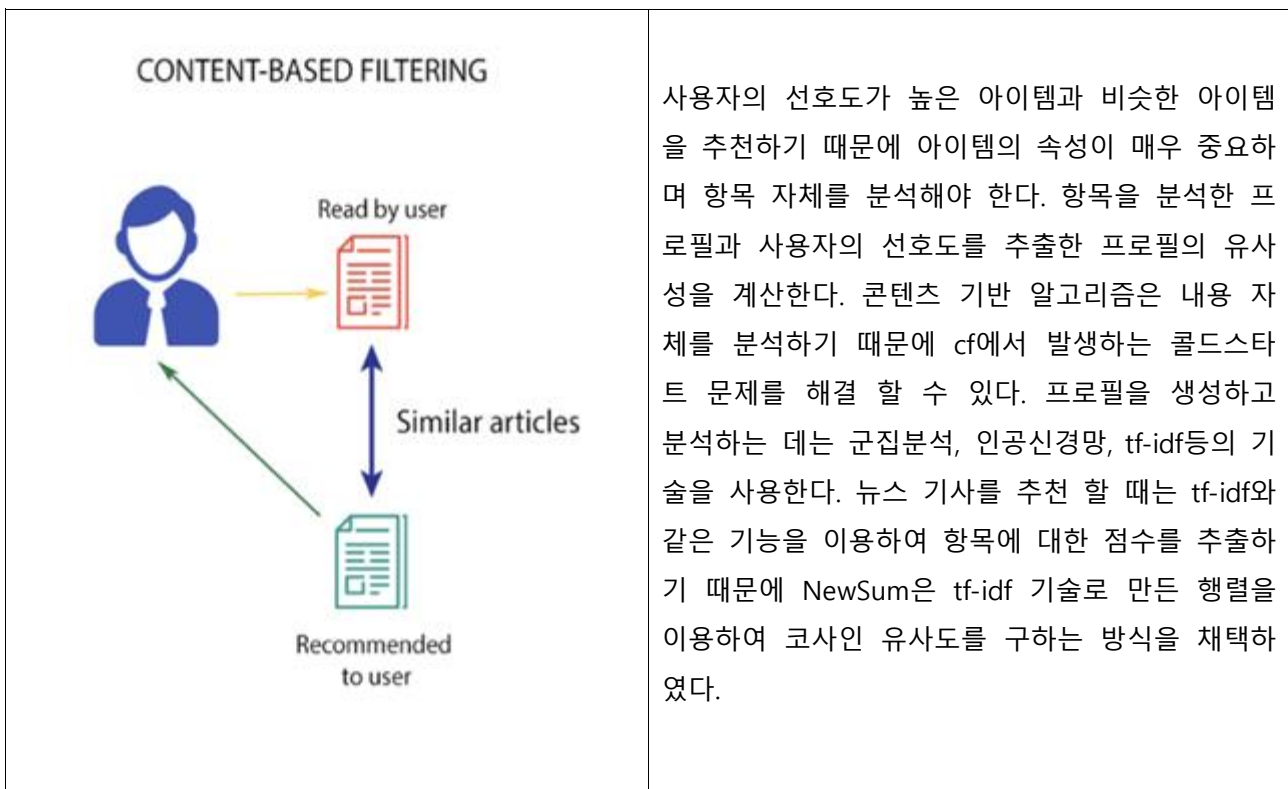
Figure 1. The categorization of recommender systems

추천이란 사용자가 아직 소비하지 않은 아이템 중 선호도가 높을 것으로 추정되는 아이템을 예측하는 것이다. NewSum에서는 사용자의 데이터를 기반으로 사용자가 아직 보지 않은 뉴스 중 선호도가 높을 만한 뉴스를 추천한다. 대표적인 추천 알고리즘으로는 Contents Based Filtering(콘텐츠 기반)과 Collaborative Filtering(협력 필터링)이 있으며 NewsSum은 각각의 장단점을 보완한 하이브리드 추천 시스템을 채택한다. 사용자 데이터가 부족한 초기에는 콘텐츠 기반 알고리즘을 사용하고 일정량의 사용자 데이터가 모이면 협업 필터링 알고리즘을 사용한다.

4.5.1 Collaborative Flitering – User based



4.5.2 Contents Based Flitering – Item based



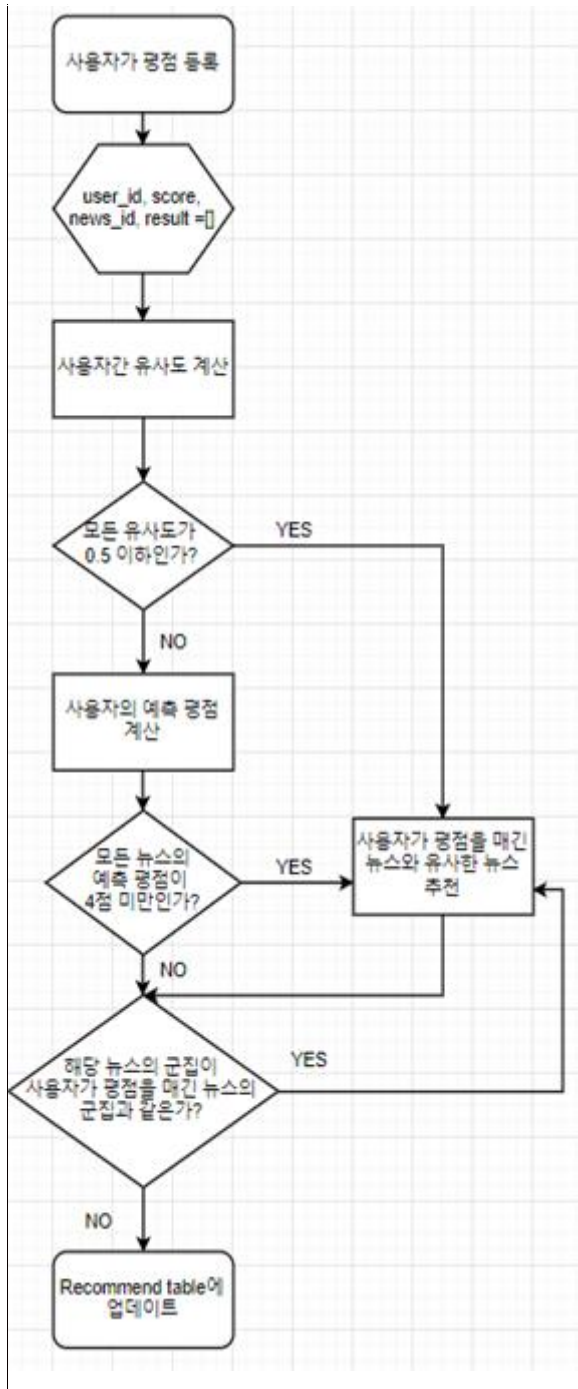
최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

4.5.3 Hybird Filtering 추천 시스템 구현

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.

사용자 데이터가 모인 이후, 협업 필터링 (Collaboration Filtering)중 사용자 기반 추천 알고리즘(User based)을 사용하며 개인 맞춤 뉴스 추천 기능을 제공한다. 초기에 평점에 대한 데이터가 전무한 사용자는 군집해서 묶인 개체가 많은 순으로 보여준다.

최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)



- ① 사용자에게 맞는 뉴스 추천을 하기 위해 DB Rating Table에서 user_id, score와 news_id를 가져온다.
- ② 가져온 데이터를 이용해 추천 받을 사용자와 타 사용자 간의 피어슨 상관계수를 구해 0.5 이상인 사용자만을 리스트에 [사용자, 상관계수] 형식으로 저장한다.
- ③ 추천 받을 사용자가 평점을 매긴 뉴스를 제외하고 리스트에 있는 사용자들이 평점을 매긴 뉴스들에 대한 평점과 유사도를 곱해 모두 더하여 딕셔너리 형태로 score_dic[news_id] = news_id: "점수" 저장하고 그 뉴스를 본 사용자의 유사도도 모두 더해 딕셔너리 형태로 sim_dic[news_id] = news_id: "유사도" 형태로 저장한 후 score_dic[news_id] / sim_dic[news_id]를 통해 해당 뉴스에 대한 추천 받을 사용자의 예측 평점 값을 구하고 그 값이 4점 이상인 것만 result_list에 저장한다.
- ④ result_list에 있는 뉴스들의 cluster_id가 DB Recommend Table에 있고 그에 해당하는 user_id가 추천 받을 사용자이면 result_list에서 삭제한다.
- ⑤ 만약 result_list가 비었다면 사용자가 본 뉴스 중 가장 높은 평점을 매긴 뉴스들을 이용해 그와 유사한 뉴스를 가져오고 result_list에 저장한 후 위와 같이 DB Recommend Table을 검사한다.
- ⑥ result_list에 남아있는 뉴스의 cluster_id를 DB Recommend Table에 user_id와 함께 저장한다.

4.5.4 추천 시스템 구현 시 고려 사항

I TF-IDF와 코사인 유사도

```
tf = TfidfVectorizer(analyzer='word', ngram_range=(1, 2), min_df=5, max_df=0.80, stop_words=stopword)
tfidf_matrix = tf.fit_transform(ds['content'])
```

[tf-idf]

```
cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)
for idx, row in ds.iterrows():
    similar_indices = cosine_similarities[idx].argsort()[:-100:-1]
    similar_items = [(cosine_similarities[idx][i], ds['id'][i]) for i in similar_indices]
    results[row['id']] = similar_items[1:]
```

[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.

I 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.

I 평점 예측

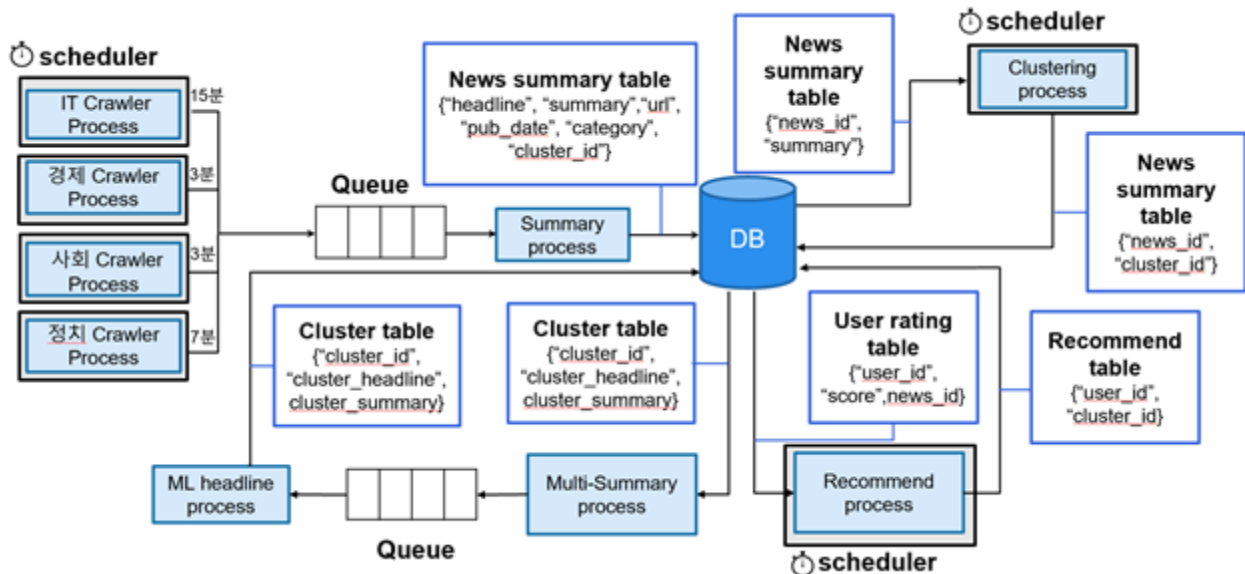
평점 예측은 피어슨 상관계수가 가장 높은 사람의 평점으로 계산 하는것보다 전체 사용자 중 유사도가 0.5 이상인 사람들의 평점들로 계산 하는것이 더 추천하는데 정확하다고 판단해 뉴스 추천 받을 사용자를 제외하고 그 사람과의 피어슨 상관계수가 0.5이상인 사람들의 뉴스평점과 유사도를 곱해 추측 평점을 구한 후 모두 더한 다음 유사도 총합을 나눠 나온 점수로 사용자의 평점을 예측한다.

I 뉴스 추천 주기

각 뉴스의 추천이 아닌 그 뉴스가 속한 클러스터를 추천해 주고 뉴스에 대한 평점 데이터가 일정 수준 쌓여야 하기 때문에 클러스터링 된 후 1 시간 뒤 실행하며 스케줄러 모듈을 이용하여 1 시, 7 시, 13 시, 16 시, 19 시, 22 시에 추천 시스템을 사용한다.



4.6 Scheduling과 Multiprocessing



NewSum은 사용자에게 뉴스를 실시간으로 제공하며 주기적으로 뉴스를 군집화하고 추천해 주기 위해 Scheduling 기능을 사용한다. 또한 News Summary와 News Crawling을 병렬처리, Headline Summary와 Contents Summary, News Clustering기능을 병렬처리 하기 위해 multiprocessing 기능을 사용한다.

스케줄러로는 APScheduler를 사용하며 이는 함수의 주기적 수행을 도와주는 라이브러리이다. Newsum은 일정 주기로 함수를 수행시키는 Interval 수행방식을 사용한다. 또한 스케줄 종류로는 다수 수행에 사용되는 BackgroundScheduler를 사용한다.

또한 멀티 프로세싱을 활용하여 시간이 오래 걸리는 크롤링등의 작업을 별도의 프로세스를 생성 한 후 병렬로 처리하여 성능을 높였다. Multiprocessing 패키지는 스레드 대신 서브 프로세스를 사용하여 파이썬의 GIL을 피할 수 있다.

4.6.1 News Producing System에 적용

- ① News Crawler와 News Summary 간의 객체 교환이 있기 때문에 멀티 프로세싱의 Queue를 생성한다.
- ② 스케줄러를 생성하여 crawler를 제어한다. 카테고리 별로 총 4개의 프로세스를 생성한다. 스케줄러를 이용하여 일정 시간 간격으로 계속하여 crawler함수를 실행시킨다. 경제: 3분, 사회: 3분, 정치: 7분, IT:15분

```

if "economy" in category:
    old.append(category["economy"])
    sched.add_job(Crawler.crawling, 'interval', seconds=180, id='test_1', args=["economy", q]) #
if "IT_science" in category:
    old.append(category["IT_science"])
    sched.add_job(Crawler.crawling, 'interval', seconds=900, id='test_2', args=["IT_science", q])
if "society" in category:
    old.append(category["society"],)
    sched.add_job(Crawler.crawling, 'interval', seconds=180, id='test_3', args=["society", q]) #
if "politics" in category:
    old.append(category["politics"])
    sched.add_job(Crawler.crawling, 'interval', seconds=420, id='test_4', args=["politics", q])
  
```



최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

간격으로 크롤링을 실행한다.

③ 크롤링 된 데이터는 큐에 넣어지고 Summary 프로세스가 그 데이터를 큐에서 꺼내와 처리한다. Crawler와 Summary프로세스는 병렬적으로 실행되어 처리속도가 빨라진다.

4.6.2 News Providing System에 적용

```
q1 = Queue()
q2 = Queue()
sched = BackgroundScheduler()

sched.start()
sched.add_job(random_four_news, 'cron', hour='7', id="s1", args=[df, cluster_id, q1])
sched.add_job(random_four_news, 'cron', hour='13', id="s2", args=[df, cluster_id, q1])
sched.add_job(random_four_news, 'cron', hour='16', id="s3", args=[df, cluster_id, q1])
sched.add_job(random_four_news, 'cron', hour='19', id="s4", args=[df, cluster_id, q1])
sched.add_job(random_four_news, 'cron', hour='22', id="s5", args=[df, cluster_id, q1])
sched.add_job(random_four_news, 'cron', hour='1', id="s6", args=[df, cluster_id, q1])

process_summary = Process(target=summary, args=(q1,q2,))
process_mlheadline = Process(target=ml_headline, args=(q2,"test",))
```

① 스케줄러를 생성하여 News Cluster를 제어한다. 카테고리 별로 총 4개의 프로세스를 생성한다. 스케줄러를 이용하여 일정 시간 간격으로 계속하여 Cluster함수를 실행시킨다.

② news clustering(군집화)가 완료된 후 1시간의 텀을 두고 스케줄러가 실행된다. 군집화된 뉴스들을 다중 요약(multi-summary)한다. 다중 요약과 ml_headline_processing을 병렬적으로 진행하기 위해서 둘 사이에 queue를 이용한다. 다중 요약된 contents들은 queue에 put하고 ml_headline_processing처리할 때 queue 에서 다중 요약된 contents들을 get하여 처리한다.

4.6.3 News Recommend System에 적용

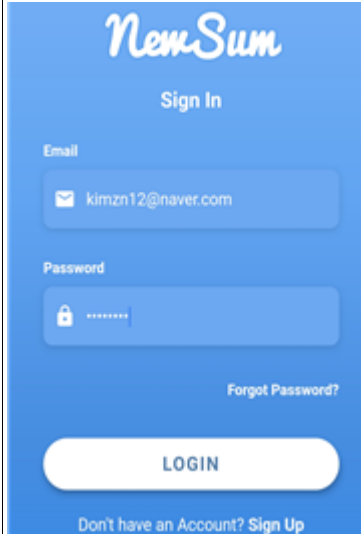
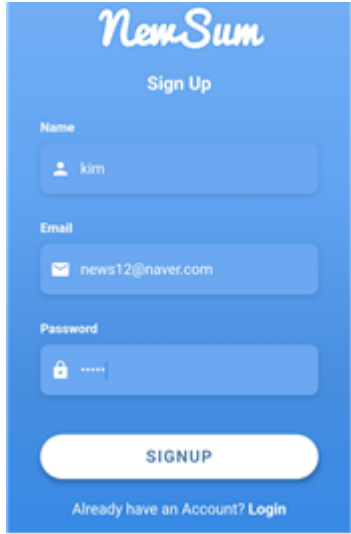
```
sched.start()
for i in user:
    sched.add_job(Recommender.getRecommendation, 'cron', hour='7', minute'30', id=None, args=[i])
    sched.add_job(Recommender.getRecommendation, 'cron', hour='13', minute'30', id=None, args=[i])
    sched.add_job(Recommender.getRecommendation, 'cron', hour='16', minute'30', id=None, args=[i])
    sched.add_job(Recommender.getRecommendation, 'cron', hour='19', minute'30', id=None, args=[i])
    sched.add_job(Recommender.getRecommendation, 'cron', hour='22', minute'30', id=None, args=[i])
    sched.add_job(Recommender.getRecommendation, 'cron', hour='1', minute'30', id=None, args=[i])
```

① 클러스터링된 이후 일정 시간이 지난 후 즉 1시간 반 이후로 정했다.

5. 기능 설명

5.1 인터페이스 및 기능 설명

5.1.1 로그인/회원가입 기능

로그인 창	회원가입 창
 <p>The login screen for NewSum. It features the 'NewSum' logo at the top, followed by 'Sign In'. There are input fields for 'Email' (containing 'kimzn12@naver.com') and 'Password' (masked with dots). A 'Forgot Password?' link is below the password field. At the bottom is a large 'LOGIN' button and a link 'Don't have an Account? Sign Up'.</p>	 <p>The sign-up screen for NewSum. It features the 'NewSum' logo at the top, followed by 'Sign Up'. There are input fields for 'Name' (containing 'kim'), 'Email' (containing 'news12@naver.com'), and 'Password' (masked with dots). At the bottom is a large 'SIGNUP' button and a link 'Already have an Account? Login'.</p>

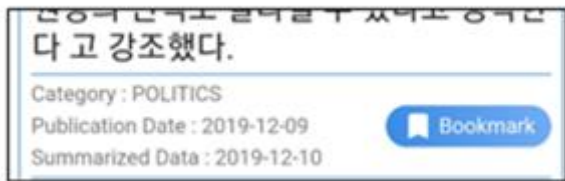
1. 회원가입 기능

- ① NewSum은 로그인을 한 사용자에게 한하여 서비스를 제공하기 때문에 사용자는 서비스를 이용하기 위해 먼저 회원가입을 해야한다. 첫 화면(로그인 창)의 Sign up버튼을 누르면 우측 그림과 같이 회원가입 창으로 넘어간다.
- ② 사용자가 Newsum APP에서 Email과 Password와 Name을 입력하고 Sign up 버튼을 누르면 data를 Firebase로 보낸다.
- ③ Firebase에서 유효ID를 Database ORM을 전송하여 DB를 만들고 데이터를 저장한다. 회원가입을 성공하면 다시 로그인 창을 보여준다.

(2) 로그인 기능(자동)

- ① 사용자가 Newsum APP에서 Email과 Password를 입력하고 로그인 누르면 사용자의 data는 Firebase로 보내진다. Firebase를 통해 유효한 Email과 password인지 검사한다. 유효한 정보라면 로그인 성공된다.
- ② 로그인에 성공하면 회원의 정보를 가져와 홈 메인화면을 보여준다.
- ③ 한 번 로그인에 성공 하면 정보는 local DB에 저장되므로 다음에 앱을 실행 시킬 시 자동 로그인 된다.

5.1.2 북마크 기능



▶ 사용자는 개별 뉴스의 Readmore 화면에서 Bookmark 표시를 하여 뉴스 스크랩을 할 수 있다.

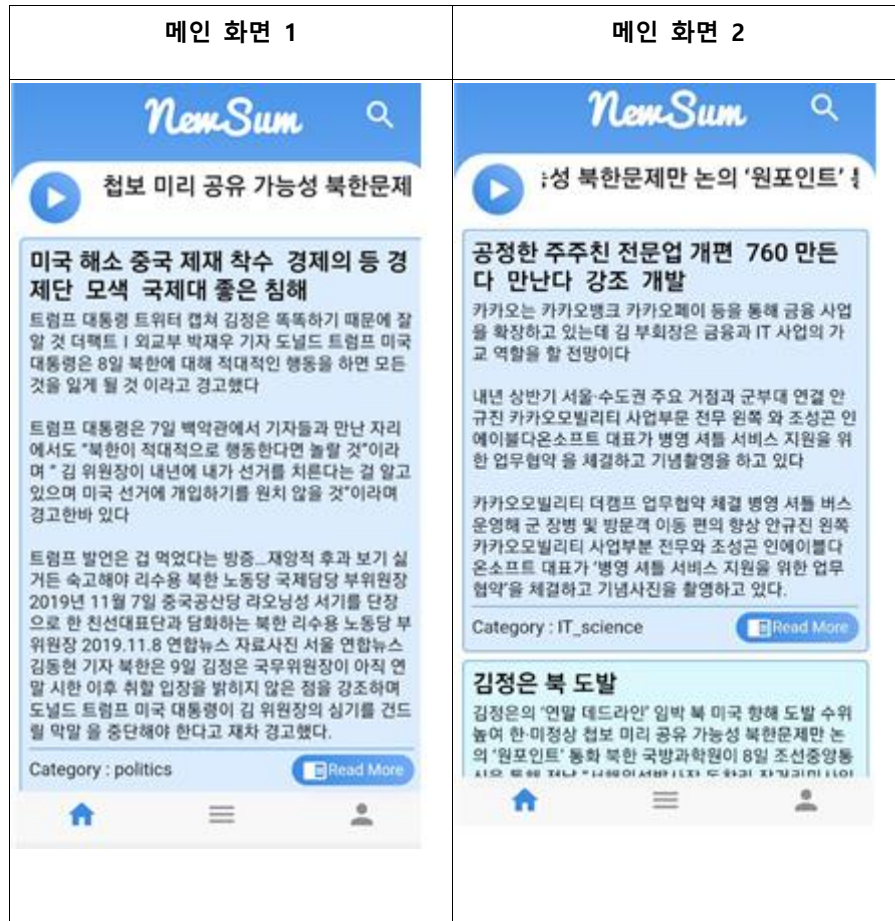


▶ 사용자는 과거 스크랩한 뉴스들을 Bookmark 화면에 가서 볼 수 있다. 이때 사용자는 스크랩한 뉴스들의 제목과 내용을 제공 받을 수 있다.

- ① 사용자가 개별 뉴스의 Readmore 화면에서 뉴스 북마크 표시를 하게 되면, 사용자 ID와 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.
- ② 사용자가 북마크 화면을 클릭하면, 사용자 ID를 보내어 해당 사용자의 과거 스크랩 뉴스정보들을 FavUser table과 FavSum table에서 전달받아 화면으로 제공받는다.

최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

5.1.3 홈 메인 화면




(사용자별로 홈 메인 화면에 보이는 기사가 다른 것을 확인할 수 있다.)

- ① 홈 화면에서는 카테고리에 상관없이 종합적으로 뉴스가 보여진다. 뉴스들은 주제별로 묶여있으며, 선택 시 3줄로 요약된 뉴스들을 볼 수 있다. 상단의 스피커 버튼을 누르면 뉴스 브리핑을 들을 수 있다.
- ② 홈화면을 클릭하면 사용자는 먼저 사용자 개별 추천순, 군집 뉴스(묶음 뉴스)의 크기순으로 뉴스를 제공받을 수 있다. 이때 개별 추천순은 사용자 별 추천 뉴스를 의미하고, 군집 뉴스의 크기는 현재 이슈되고 있는 정도를 측정하여 이슈의 랭킹을 의미한다.
- ③ 사용자가 APP에서 홈화면을 클릭하면 Application server로 User_id와 홈화면 상태를 전송하고 Application server가 이를 Database로 전달한다.
- ④ Database는 정보를 전달받은 후, Recommend table과 Cluster table의 cluster_size를 통해 뉴스의 순서와 해당 뉴스들의 news_id를 통해 articles table에서 정보를 가져와 Application server에 전송하여 APP 화면에 사용자의 추천 뉴스와 군집 뉴스가 보여진다.



최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

5.1.4 카테고리 화면

카테고리 창	카테고리 별 뉴스
	

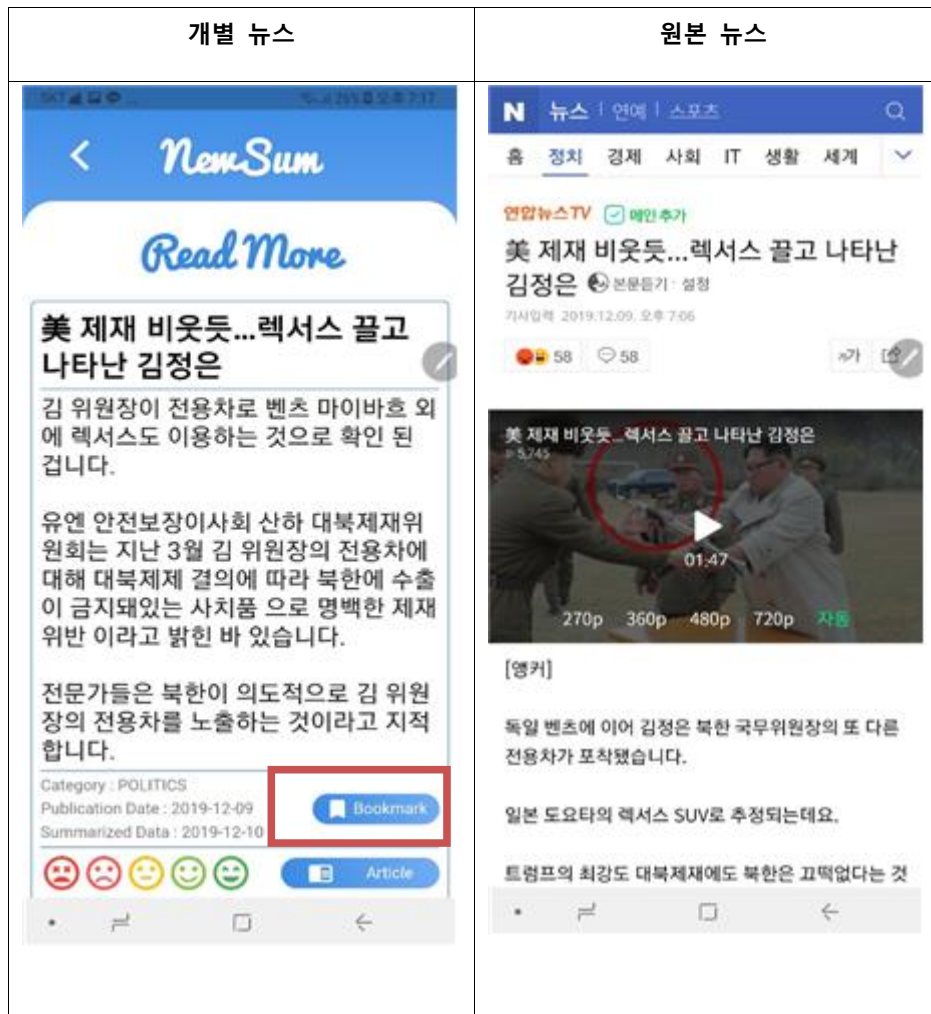
- ① 총 4가지 카테고리(Society, Economy, IT, Politics)중 선택 가능하며 각 카테고리 별로 뉴스를 보여준다. 뉴스는 일괄적인 것이 아닌 사용자마다 추천 뉴스를 보여준다.
- ② 카테고리 선택화면에서 특정 카테고리를 클릭하면 사용자는 해당 카테고리에 대한 군집 뉴스와 개별 뉴스를 제공받을 수 있다.
- ③ 사용자가 APP의 카테고리 선택 화면에서 Society, Economy, IT & Science, Politics 카테고리 중 원하는 카테고리를 선택하면 Application server로 해당 카테고리 상태가 전달된다.
- ④ Application server는 Database로 정보를 전달하고 Database는 Cluster table 내에서 해당 카테고리에 대한 군집뉴스 정보와 개별 뉴스 정보를 Application server에게 전송한다.
- ⑤ Application server는 받은 군집 뉴스 정보를 APP화면에 보여준다.

5.1.5 군집 뉴스 보여주기

군집 뉴스	군집 뉴스의 Read More 화면
	

- ① 군집 뉴스란 Newsum 시스템을 통해 새롭게 가공된 뉴스로, 유사한 내용을 가진 여러개의 개별 뉴스를 묶어 새로 생성한 뉴스이다. Database의 Cluster table의 cluster_headline과 cluster_summary가 군집 뉴스의 제목과 내용이 된다.
- ② 각 군집 뉴스 마다 'Read more'버튼을 가지며 Read more 버튼을 클릭하면 클러스터 내부 화면에서 해당 cluster의 개별 뉴스들을 볼 수 있다. 이때 이 개별 뉴스들이 보여지는 순서를 cluster_size가 결정한다.
- ③ Application server가 사용자의 요청에 따라 해당되는 상태를 Database로 전달 해 주면 Database의 Cluster table에서 cluster size를 통해 군집 뉴스가 나열되는 순서에 대한 data를, Cluster table에서 FK인 news_id를 통해 해당 cluster 내 개별 뉴스들의 뉴스 정보 data를 news_id가 PK로 있는articles table에서 불러와 Application server에게 전송한다.
- ④ Application server에서 전달 받은 data를 통해 군집 뉴스를 articles table의 뉴스 정보로, 특정 군집 뉴스의 'Read more'버튼을 클릭한 클러스터 내부화면에서 cluster_size 순서로 개별 뉴스를 APP 화면에 보여 준다.

5.1.6 개별 뉴스 보여주기



① 개별 뉴스를 보여주는 상태는 카테고리 화면에서 보여주는 개별 뉴스(첫번째 그림)와 군집 뉴스의 'Read more' 클릭하여 클러스터 내부 화면(두번째 그림)에서 보여주는 개별 뉴스가 있다. 하지만 흔히 말하는 개별 뉴스는 카테고리 화면에서 보여주는 개별 뉴스를 의미한다.

② Application server가 사용자의 요청에 따라 해당되는 카테고리 상태 혹은 클러스터 내부 화면 상태를 Database로 전달 해 주면, Database의 articles table에서 FK인 cluster_id가 default인 뉴스들의 headline, summary, category, pub_date, sum_date, url를 Application server에게 전송한다.

③ Application server에서 전달 받은 data를 통해 카테고리 화면 상태에서는 pub_date, headline, summary의 일부를, 클러스터 내부 화면에서는 pub_date, headline만 APP 화면에 보여준다.

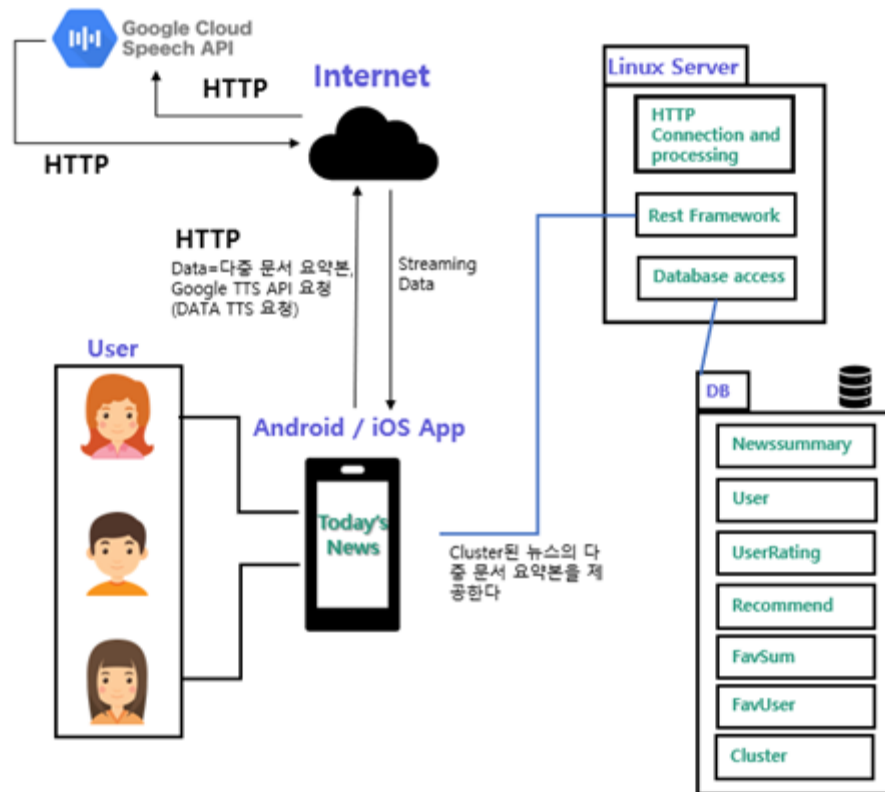
④ 이때 개별 뉴스 보기에서도 'Read More'버튼이 있다.

버튼 클릭 시 ReadMore화면이 보여지고 여기서는 해당 개별 뉴스의 headline, summary, category, pub_date, sum_date를 보여주고 북마크 버튼, 평점 버튼, Article 버튼이 존재한다. 이때 Article 버튼을 클릭하면 해당 개별 뉴스의 url 페이지가 보여지게 된다.

5.1.7 평점 기능

- ① 사용자는 개별 뉴스마다 평점을 부여할 수 있다. 1-5점까지의 평점을 매길 수 있다.
- ② Application server가 사용자가 부여한 평점 데이터를 Database로 전달 해 주면, Database의 UserRating table의 score에 평점이 저장된다. 이 정보는 후에 추천 시스템에 사용된다.

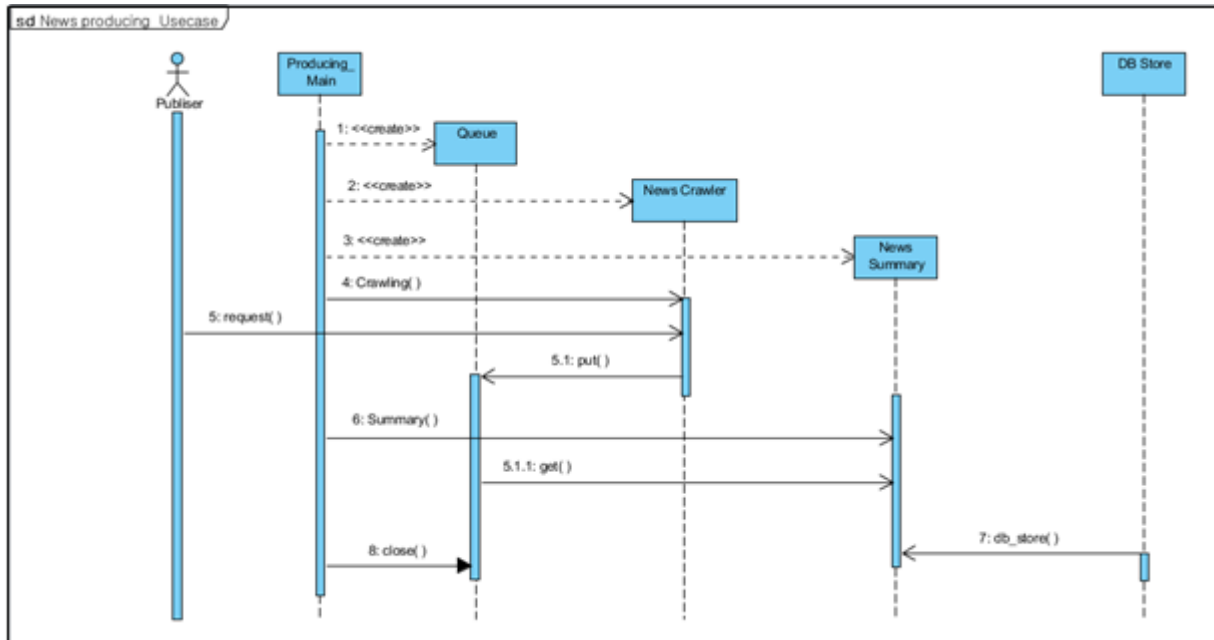
5.1.8 TTS 기능



- ① 사용자가 APP의 홈화면에서 플레이 버튼을 클릭하면 Google TTS API로 Data[홈화면 내의 추천 뉴스]를 전송한다.
- ② Google TTS API는 Streaming 방식으로 APP으로 Data의 음성을 전달하여 사용자는 추천 뉴스를 라디오처럼 들을 수 있다.

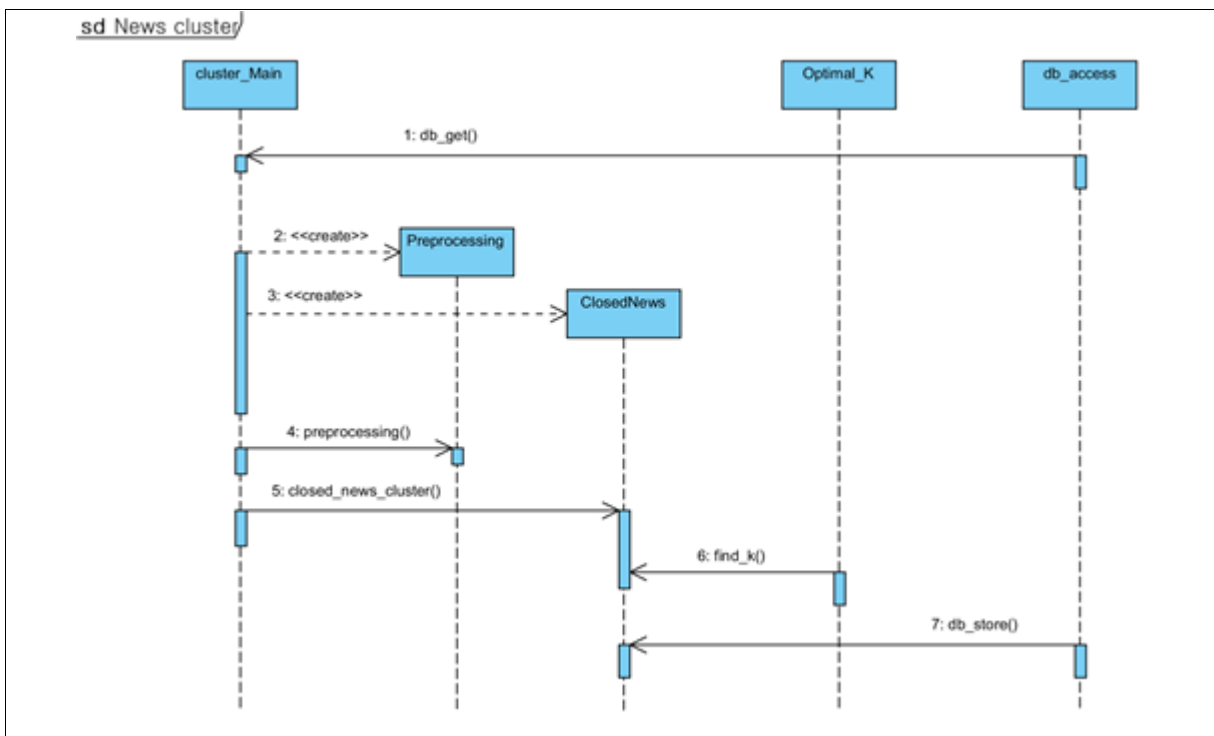
5.2 Sequence Diagram

5.2.1 News Provide System Sequence Diagram



[Figure 17] 뉴스제공 sequence diagram

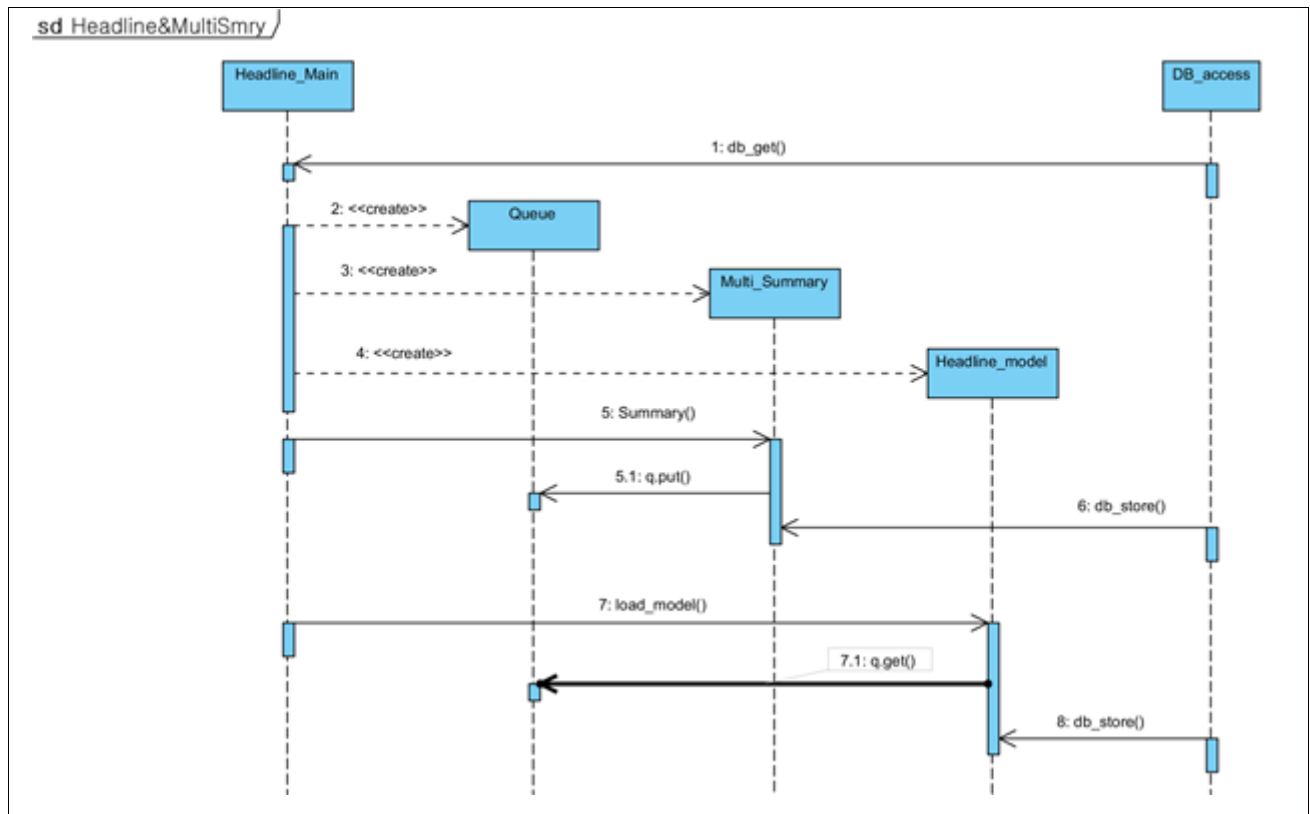
5.2.2 News Cluster System Sequence Diagram



[Figure 18] News Cluster sequence diagram

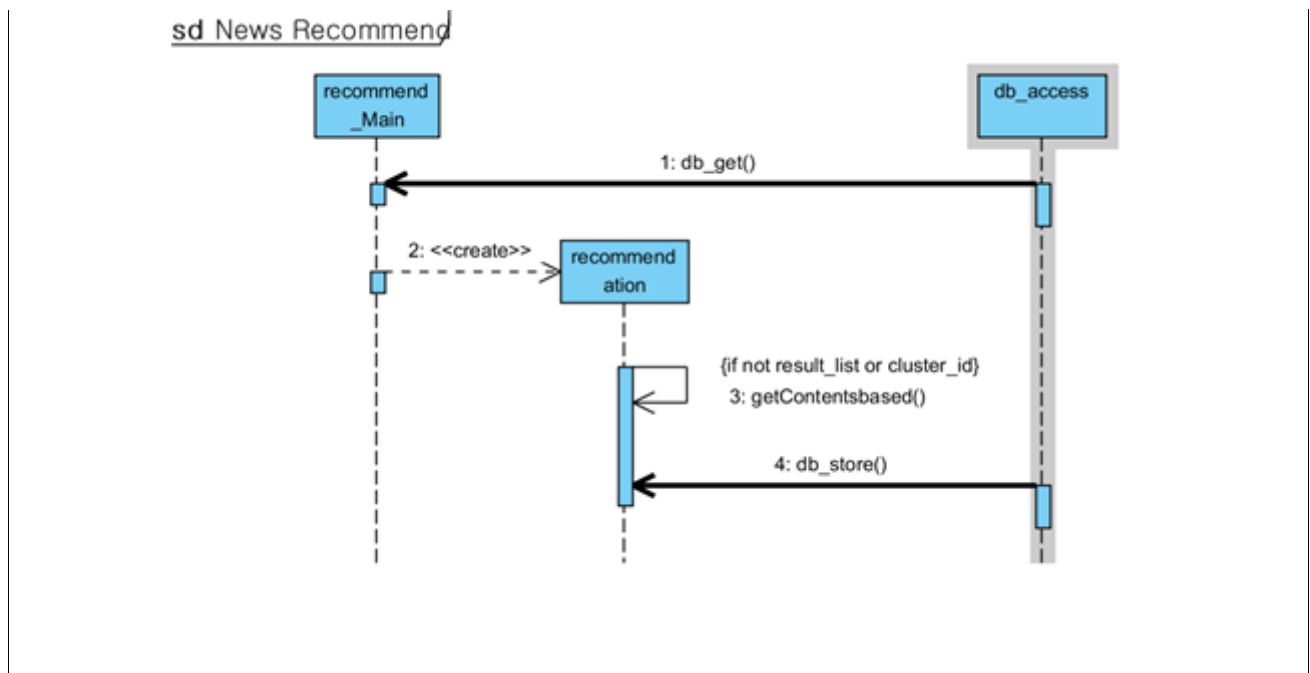


5.2.3 Headline & Multi-summary System Sequence Diagram



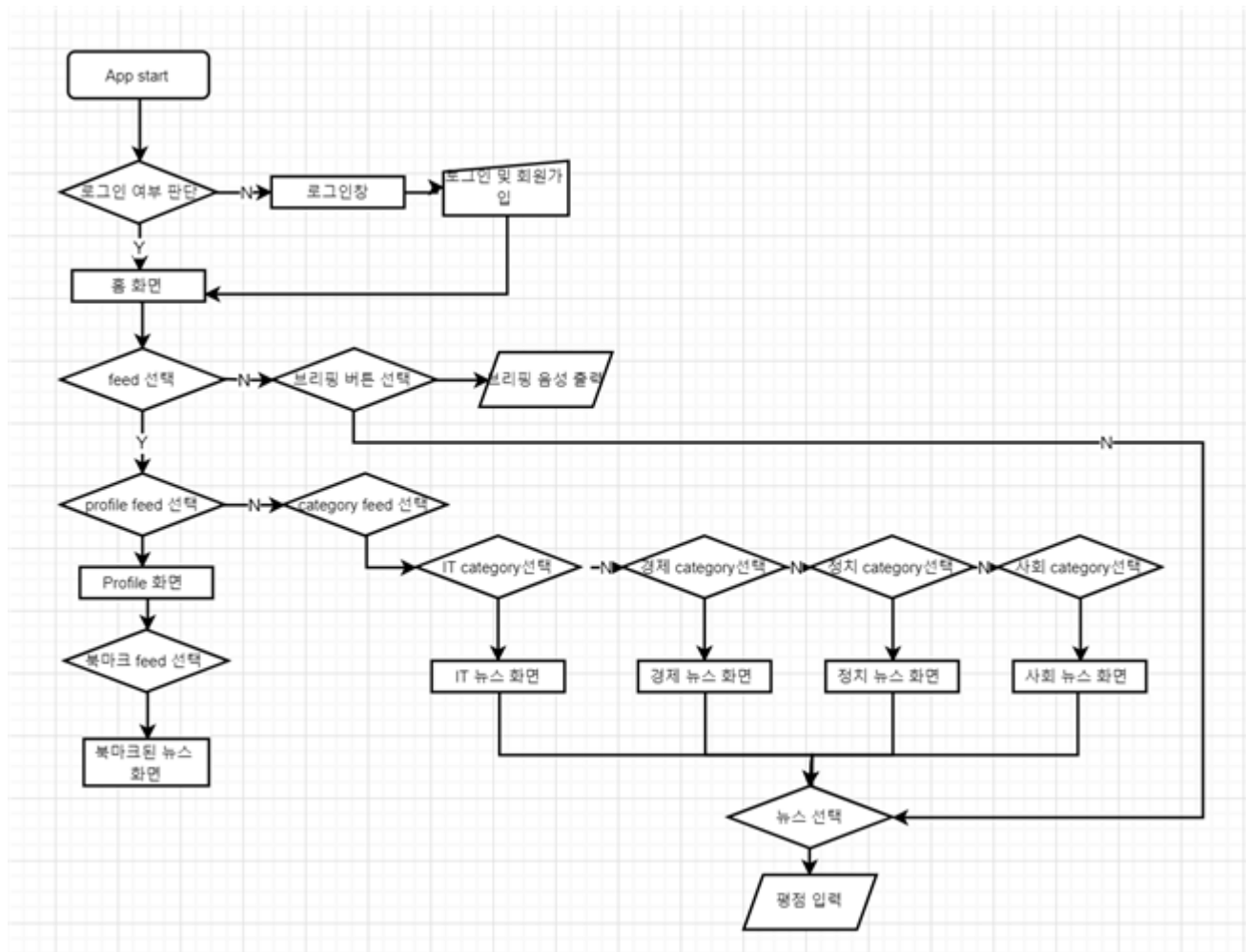
[Figure 19] Headline & Multi-summary sequence diagram

5.2.4 News Recommend System Sequence Diagram



[Figure 20] News Recommend sequence diagram

5.3 APP flow chart



[Figure 21] 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 News 는 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 News 는 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.

최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

7. 프로젝트 세부 추진 계획 및 일정

프로젝트 기간 개발 내용	2019.9.9~2019.12.9												
	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 정의 서 작성 서버 및 데이터 베이스 구현 앱 구현 뉴스 크롤링													
클러스터링 시스템 구현													
요약 시스템 구성(추상적) 요약 시스템 구성(추출적) 추천 시스템 구현													
개별 테스트 및 보안													
통합 테스트 및 보안													
최종 발표 및 시연													

최종보고서: Machine Learning을 이용한 뉴스 요약 및 추천 어플리케이션(Today News)

8. 팀원 담당 업무

이름	정	부	비고
박주영	Cluster의 headline 생성	News crawling, multiprocessing	팀장
김아연	Multiprocessing, collaborative recommendation	News crawling, News clustering	팀원
김혜원	News clustering	Multiprocessing	팀원
홍승환	Hybrid recommendation	Summarization of cluster	팀원
이산가	Web server, Database	App UI	팀원