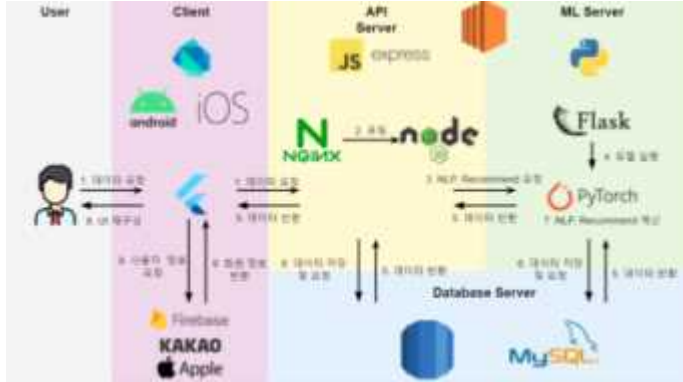


2021 한이음 공모전 개 발 보 고 서

2021. 10. 18

프로젝트명	국문	빅데이터와 AI를 활용한 융합형 콘텐츠 마인드맵 생성 플랫폼 구축
	영문	Mindmap Search and Scrap Platform Using Big Data and AI
작 품 명	웹서핑의 지도, Findmap	

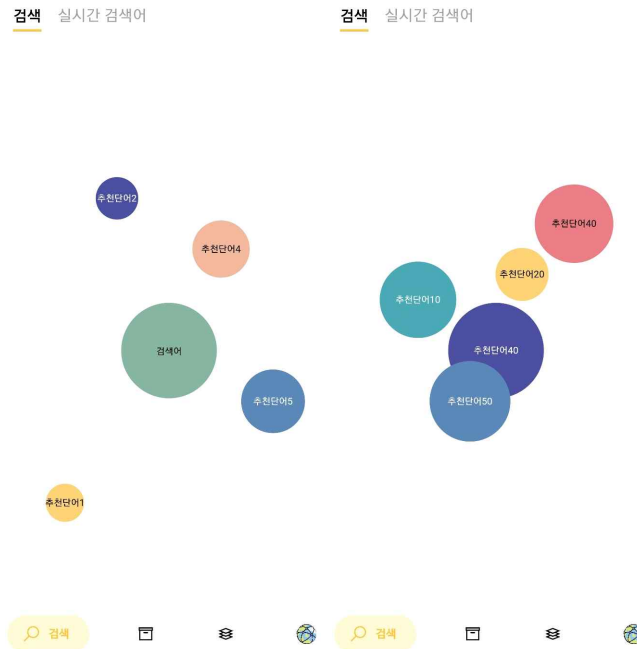
요 약 본

작품 정보		
프로젝트명	국문	빅데이터와 AI를 활용한 융합형 콘텐츠 마인드맵 생성 플랫폼 구축
	영문	Mindmap Search and Scrap Platform Using Big Data and AI
작품명	웹서핑의 지도, Findmap	
작품 소개	<p>‘Findmap’은 Find와 Mindmap의 map을 합친 단어로 사용자들이 웹서핑할 때 헤매지 않도록 지도(맞춤 검색 결과, 맞춤 게시물 추천) 또는 마인드맵(맞춤 검색어)을 그려주어 방향성을 정해주고자 하는 이중적인 의미를 담아 만든 애플리케이션이다.</p> <p>‘Findmap’은 검색 시 사용자의 데이터를 기반으로 맞춤 검색 결과를 보여주고, 사용자가 검색할 것 같은 단어를 예측해서 추천 검색어를 제시한다. 또한, 검색 중 저장하고 싶은 게시글이 있을 때 나만의 아카이브에 스크랩하고 관리할 수 있다. 마지막으로 SNS 기능을 추가해서 다른 사람들과 게시글을 공유하며 소통할 수 있다.</p>	
작품 구성도	 <p>The diagram illustrates the system architecture. It is divided into five main sections: User, Client, API Server, ML Server, and Database Server. The User interacts with the Client (iOS/Android) via 1. Search and 2. Save. The Client sends requests to the API Server (Node.js, Express, NGINX) via 3. Search and 4. Save. The API Server interacts with the ML Server (Flask, PyTorch) via 5. API Recommendation and 6. API Recommendation. The ML Server interacts with the Database Server (MySQL) via 7. API Recommendation and 8. API Recommendation. The Database Server interacts with the Client via 9. Search and 10. Save. The Client also interacts with Firebase/KAKAO/Apple via 11. Search and 12. Save.</p>	
작품의 개발배경 및 필요성	<ul style="list-style-type: none"> • 검색을 했을 때 원하는 검색 결과가 나오지 않아서 검색에 소비하는 시간이 많음 • 실시간 검색어가 없어서 트렌드를 파악하기 어려움 • 유용한 글은 저장해서 여러 번 보고 싶음 • 유용한 글은 다른 이들과 함께 공유하고 싶음 	
작품의 특징점	<ul style="list-style-type: none"> • 사용자 기반 검색 및 검색어 추천 • AI를 이용한 게시물 추천 • 나만의 아카이브에 스크랩 • SNS를 이용한 게시물 공유 	
작품 기능	<ol style="list-style-type: none"> 1. 맞춤형 검색 결과 2. 검색어 추천 3. 맞춤형 게시물 추천 4. 실시간 검색어 5. 스크랩 기능 6. SNS 기능 	
작품의 기대효과 및 활용분야	<ol style="list-style-type: none"> 1. 나에게 알맞은 검색 결과로 검색에 필요한 노력과 시간을 단축할 수 있다. 2. 맞춤형 검색어, 맞춤형 게시물, 실시간 검색어를 통해 웹서핑을 수월하게 할 수 있다. 3. 나만의 게시물들을 한 번에 관리하면서 언제든지 열어볼 수 있다. 4. SNS로 사람들과 게시글을 공유하고 소통할 수 있다. 5. 기존에 없던 새로운 검색 및 스크랩 애플리케이션 서비스로 자리 잡을 수 있다. 	

본 문

I. 작품 개요

1. 작품 소개



[그림 1] 키워드 검색의 예시

1) 정의

- (1) ‘Findmap’ 은 Find와 Mindmap의 map을 합친 단어
- (2) 웹서핑할 때 지도의 역할(맞춤 검색 결과, 맞춤 게시글)을 하고자 함
- (3) 또한, 마인드맵(맞춤 검색어)을 그려주어 방향성을 제시함

2) 기획 의도

- (1) 검색에 소비되는 시간이 많으므로 AI를 통해 이를 해결하고자 함
- (2) 실시간 검색어를 부활시켜 사용자가 쉽게 트렌드를 읽을 수 있게 하고자 함
- (3) 스크랩과 SNS 기능을 통해 좋은 글을 타인과 공유하게 하고자 함

3) 작품 내용

- (1) 맞춤형 검색 결과를 통해 사용자가 원하는 글을 상단에 위치시켜 검색을 더욱 쉽게 함
- (2) 검색어 추천을 통해 웹서핑을 더욱 수월하게 도와줌
- (3) 맞춤 게시글을 통해 사용자는 검색하지 않아도 흥미로운 글을 접할 수 있음

- (4) 실시간 검색어로 최신 트렌드를 알 수 있음
- (5) 스크랩 기능으로 원하는 글을 나만의 아카이브에 저장할 수 있음
- (6) SNS 기능을 통해 타인과 게시글을 공유할 수 있음

2. 작품의 개발 배경 및 필요성

1) 개발 배경

- (1) 검색에 드는 시간이 많음
- (2) 전문 용어는 정확한 용어를 검색하지 않으면 해당 결과를 찾기 어려움
- (3) 좋은 글은 타인과 공유하여 지식 습득 혹은 문제 해결 등에 도움을 주는 등 소통과 협력을 하고자 함
- (4) 마인드맵 형식에서 착안한 추천 검색어와 맞춤형 추천 게시글로 검색 및 취사 선택을 도와주는 서비스를 제공하고자 함
- (5) 스크랩과 SNS 기능을 통해 글을 공유하고 소통하는 서비스를 제공하고자 함

2) 필요성

- (1) 정보화 시대에 원하는 정보 글을 찾을 수 있는 검색 및 취사 선택 능력이 필요함
- (2) 검색에 드는 비용을 줄이는 일은 시간 관리에 필수적임
- (3) 유용한 정보 글은 반복적으로 읽기 때문에 해당 글의 효과적인 저장, 접근 및 관리 기능이 필요

3. 작품의 특징 및 장점

1) 특징 및 장점

- (1) 검색어 추천: Findmap AI로 적절한 검색어를 제공하여 원활한 웹서핑 가능
- (2) 실시간 검색어: 화젯거리인 검색어를 통해 최신 트렌드 예측 가능
- (3) 맞춤형 검색 결과: Findmap AI로 검색 결과를 사용자 취향에 맞게 정렬하여 검색 시간 최소화
- (4) 맞춤형 게시글 추천: 사용자의 취향을 바탕으로 관심 있어 할 만한 글 제공
- (5) 스크랩: 원하는 글을 나만의 아카이브에 바로 저장
- (6) SNS: 팔로우, 팔로잉 기능과 공유 기능을 통해 사용자 간 상호작용 가능

2) 유사 상용 서비스인 Pocket과의 비교를 통한 차별성

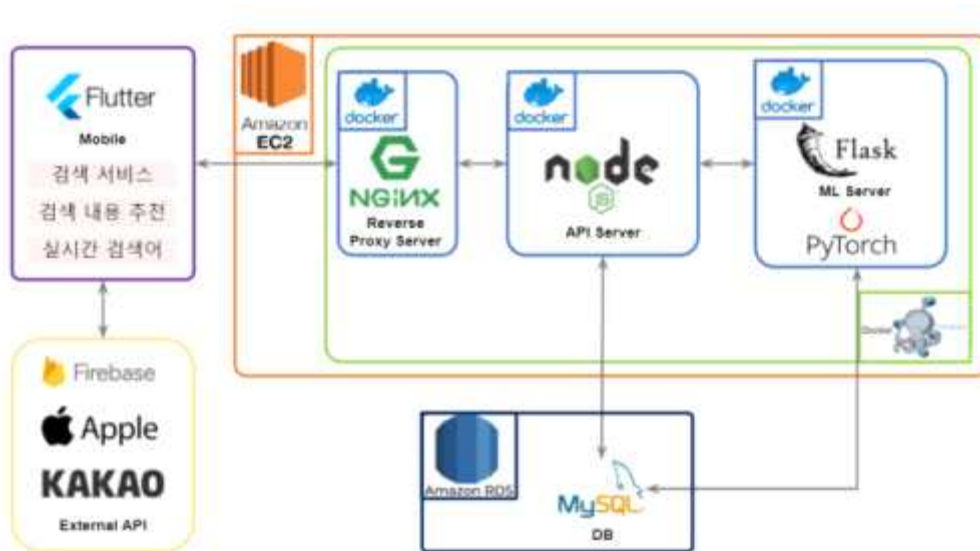
- (1) Pocket은 해외에서 유명한 스크랩 애플리케이션으로, 사용자가 원하는 웹페이지 글을 스크랩하고, 이를 애플리케이션 내에서 읽을 수 있도록 기능을 제공
- (2) Findmap은 마인드맵 형식을 통한 검색 기능 제공을 통해 사용자가 원하는 정보를 쉽게 찾을 수 있도록 돕는 기능을 추가
- (3) 위 기능은 사용자의 검색 시간을 단축해줄 뿐만 아니라, 검색 능력 향상에도 도움을 줄 것으로 기대됨

II. 작품 내용

※ 평가항목 : 기술력 (기능구체성, 난이도, 완성도)

1. 작품 구성도

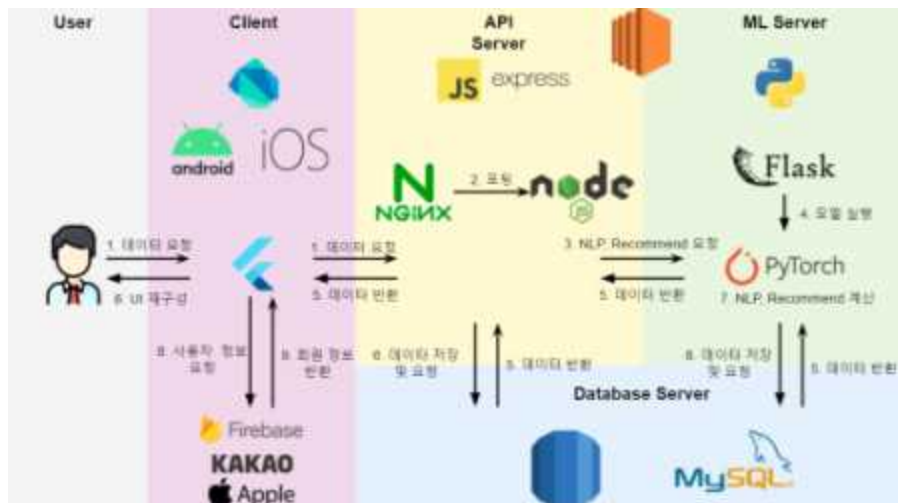
1) S/W 구성도



[그림 2] 시스템 구조도

1. 백엔드의 각각의 인스턴스는 docker container 위에서 동작하며 docker-compose 로 구현
2. 백엔드에서 DB는 RDS를 이용해 MySQL로 구축
3. 프론트엔드는 Flutter를 이용해 Android와 iOS를 동시에 개발
4. 프론트엔드에서 External API로 사용자 authenticate 함

2) 서비스 흐름도



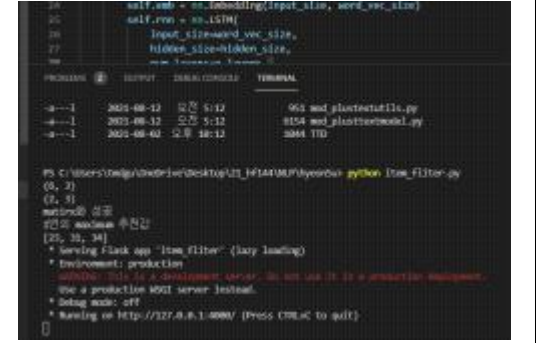
[그림 3] 서비스 흐름도

2. 작품 기능

1) 전체 기능 목록

구분	기능	설명	현재 진척도
S/W	회원가입 및 로그인	- 이메일 및 SNS를 이용한 회원가입, 로그인 서비스 제공	100%
	검색	- 실시간 검색어 순위 - 범주별 실시간 검색어 순위 - 사용자 맞춤형 검색 결과 정렬	100%
	추천 검색어	- 사용자 맞춤형 추천 검색어 - 추천 모델 설계 완료	100%
	스크랩	- 인터넷에서 공유하기 버튼을 눌러서 스크랩 - 앱 내의 검색 기능을 통해 게시글을 보고 스크랩 - 피드를 보면서 마음에 드는 게시글 스크랩 - 스크랩한 게시글을 폴더 형태로 관리	100%
	피드	- 자신이 스크랩한 게시글 피드로 업로드 가능 - 다른 피드를 조회, 하트, 저장	100%
	추천 피드	- 추천 모델을 통해 관심 있어 할 만한 콘텐츠를 피드 형태로 제공	100%
	팔로우	- 다른 사람의 피드와 프로필을 둘러보고 팔로우	100%
	팔로우 피드	- 팔로우한 사람의 새로운 공유 콘텐츠를 피드로 모아봄	100%
	마이페이지	- 사용자 상태 및 자신의 피드 표현 - 설정 변경	100%
H/W	해당 없음		

2) S/W 주요 기능

기능	설명	작품 실물사진
연관 콘텐츠 분류(AI)	사용자가 검색어를 입력하면 검색어와 연관된 영화, 책, 기사, 음악, 예술 등의 카테고리로 분류. 이를 위해 각 콘텐츠가 어떤 카테고리에 속하는지 분류할 수 있어야 하는데, 이를 수작업으로 분류하지 않고 시를 이용해 콘텐츠의 카테고리를 분류	
사용자 기반 결과 정렬(AI)	검색 결과를 사용자의 평소 관심사에 맞게 재정렬하여 검색에 드는 비용을 감소 좌: 스타/연예인에 관심이 많은 사람 우: 요리/레시피에 관심이 많은 사람	
사용자 기반 피드 추천(AI)	User based recommendation을 통하여 사용자의 나이, 성별, 직업에 맞는 적절한 콘텐츠를 추천 사진은 DB에 사용자 테이블을 임의로 집어넣고 카테고리(32개를 IDX로 표현)에 대한 MATRIX 추출을 하여 결측치 값을 평균으로 하여 MAXIMUM 값이 추천 CONTENT가 되는 모습	

<p>마인드맵 형태의 검색어 추천(AI)</p>	<p>작은 화면에 알맞은 마인드맵 형태를 통해 검색어와 연관성 있는 타 검색어 를 추천</p>	<div> <div>검색 실시간 검색어</div> <div> </div> </div>
<p>실시간 검색어 순위</p>	<p>서비스를 통해서 검색한 키워드의 카운 트가 높은 순으로 순위를 볼 수 있으 며, 실시간 검색어 순위의 카테고리는 사용 자가 지정한 관심 카테고리가 반영됨</p>	<div> <div> <div>검색 실시간 검색어</div> <div> <div>종합</div> <div>공연 전시</div> <div>문화 예니</div> <div>IT 컴퓨터</div> <div>자동차</div> </div> <div> <ol style="list-style-type: none"> 1 맥북이 2 My Universe 3 전나비 4 태연 5 방탄소년단 6 널 7 색안경 8 알기 맥북이 9 존맛 부대찌개 10 Coldplay 11 치즈 케탄밥이 12 aespa </div> </div> <div> <div>맞춤 서비스</div> <div> <div>공식</div> <div>생활</div> <div>노하우</div> <div>쇼핑</div> <div>9</div> </div> <div> <div>관심</div> <div>일상</div> <div>생각</div> <div>육아</div> <div>결혼</div> <div>해당</div> <div>번역</div> <div>등용</div> </div> <div> <div>팔로</div> <div>좋은</div> <div>공</div> <div>이미지</div> <div>재</div> <div>선</div> <div>미용</div> </div> <div> <div>약간</div> <div>인</div> <div>태</div> <div>리</div> <div>어</div> <div>DIY</div> <div>요리</div> <div>레시피</div> <div>상</div> <div>품</div> <div>리</div> <div>뷰</div> </div> <div> <div>오전</div> <div>원</div> <div>에</div> <div>제</div> <div>애</div> </div> </div> <div> <div>비밀</div> <div>엔터</div> <div>테</div> <div>인</div> <div>트</div> <div>예술</div> <div>9</div> </div> <div> <div>문화</div> <div>학</div> <div>적</div> <div>영</div> <div>화</div> <div>마</div> <div>술</div> <div>디</div> <div>자</div> <div>인</div> </div> <div> <div>회원</div> <div>공</div> <div>연</div> <div>전</div> <div>시</div> <div>음악</div> <div>드</div> <div>라</div> <div>마</div> </div> <div> <div>로</div> <div>스</div> <div>스타</div> <div>연</div> <div>애</div> <div>인</div> <div>안</div> <div>화</div> <div>매</div> <div>니</div> <div>발</div> <div>송</div> </div> </div> <div> <div>AJK</div> <div>지</div> <div>식</div> <div>동</div> <div>향</div> <div>6</div> </div> <div> <div>IT</div> <div>컴</div> <div>퓨</div> <div>터</div> <div>시</div> <div>화</div> <div>장</div> <div>차</div> <div>건</div> <div>강</div> <div>의학</div> </div> <div> <div>비</div> <div>즈</div> <div>니</div> <div>스</div> <div>경</div> <div>제</div> <div>위</div> <div>학</div> <div>외</div> <div>국</div> <div>어</div> <div>교</div> <div>육</div> <div>학</div> <div>학</div> <div>문</div> </div>

위

미

여

가

여

학

8

계

열

스

포

츠

사

관

자

동

차

<p>피드</p>	<p>자신이 스크랩한 게시글을 피드에 업로드 가능</p> <p>타인의 피드를 조회, 하트, 저장 가능, 위 정보를 바탕으로 관련 category에 대해서 가중치를 부여함으로써 추천 피드에서 관련 내용을 추천받을 수 있음</p> <p>팔로우한 사람의 피드를 모아서 열람할 수 있음</p>	
<p>팔로우 (프로필)</p>	<p>프로필에서 피드에 저장한 글의 스크랩 수와 팔로워 수, 하트 수를 알 수 있음. 자신을 나타내는 설명과 이미지 등록 가능. 자신의 피드 노출 가능</p> <p>다른 사람의 프로필을 보고 버튼을 눌러 팔로우 가능</p>	
<p>회원가입 및 로그인</p>	<p>사용자는 자신의 성별, 나이, 직업 등의 특성에 따라 적절한 추천 서비스를 받음. 이를 위해서는 사용자를 구분할 수 있는 로그인 시스템이 필요</p>	

3) H/W 주요 기능

기능/부품	설명	작품 실물사진
해당 사항 없음		

3. 주요 적용 기술

구분	기술	주요 기능	적용범위
Front	Flutter	Mobile	애플리케이션의 기본화면 구성에 사용되었으며 android IOS 에서 모두 구동할 수 있게 사용되었다.
API Server	Node.js, Express	API 서버	애플리케이션의 REST API를 구축하는데 사용되었고 클라이언트의 정보를 다루는데 사용되었다.
	NginX	리버스 프록시 서버	클라이언트의 요청을 분산시켜서 처리하며, 서버 환경 설정을 더욱 수월하게 다룰 수 있도록 사용되었다.
	open API	사용자 정보 수집	구글, 카카오, 애플에서 제공하는 로그인 API를 이용해서 사용자의 정보를 가져온 후 데이터베이스에 저장하고 클라이언트에 넘겨주는 데 사용되었다.
ML Server	크롤링	데이터 수집	BeautifulSoup와 Selenium을 이용해 커뮤니티 사이트를 크롤링하여 데이터를 수집하였다. 해당 데이터는 콘텐츠 범주화 모델 학습에 사용되었다.
	NLP	사용자 맞춤형 추천	Collaborative Filtering은 DB의 사용자 정보에 따라 검색어와 카테고리 간의 연관성을 추출하고 사용자별 선호도를 파악하는 데 사용되었다.
			Content based filtering은 검색어의 유사단어를 통한 비슷한 맥락의 추천 검색어를 제공하는데 사용되었다.
		게시글 자동 카테고리	범주화 라이브러리로 채택한 FastText는 Text Classification에 사용되었고 이는 게시글 분류에 사용되었다.
	Flask	API 서버	ML server와 API server 간에 REST API 통신이 가능하도록 Flask가 사용되었다.
	open API	검색 결과 수집	네이버와 다음에서 제공하는 검색 API값의 데이터를 사용자의 취향에 맞게 재정렬하여 검색 결과로 보여주는데 사용되었다.
Infra	GitLab Runner	CI/CD	CI/CD는 자동 테스트 및 배포 환경구성 및 개발하는 공간과 구동하는 공간을 독립시키는데 사용되었다.
	Docker	컨테이너 구성	서버를 물리적으로 분리하지 않고 container 단위로 만들어

			다중 container를 다루는데 사용되었다.
	Docker-compose	다수 컨테이너 실행 및 관리	내부 정보 방화벽을 한층 더 강화하고, 쉘로 각 container를 연결하여 다수의 container를 관리하는데 사용되었다.
Cloud	AWS Server (Ubuntu 18.04)	물리적 서버	instance를 통해 환경 구성을 복사하거나 외부와 연결을 통해 개발용 서버와 배포용 서버를 만드는데 사용되었다.
	AWS S3	이미지 서버	사용자의 프로필 이미지를 저장할 때 사용되었다.
	Cloud DB (AWS RDS)	DataBase, MySQL	사용자 데이터, 검색 카테고리 등의 정보를 관리하는데 사용되었다.

1) 주요 적용 기술 상세 내용

(1) **Flutter**: 최근 화젯거리인 기술인 dart 언어를 이용한 프론트엔트 개발 프레임워크다. 한 번의 개발로 Android, iOS, Web에서 모두 작동하게 구현할 수 있는 장점이 있다.

(2) **Node.JS, Express**: Single thread의 non blocking 비동기 방식의 node.js와 express를 이용하여 구축하였으며 본 서비스는 AI를 기반으로 데이터를 처리하기 때문에 non blocking 비동기 방식이 필수적이고 이에 따른 REST API를 구축하기 위해 express를 사용하였다.

(3) **NginX**: 웹서버를 띄워서 리버스 프록시(reverse proxy) 역할과 동시접속 처리를 한다. 리버스 프록시는, 클라이언트가 가짜 서버에게 요청(request)하면, 프록시 서버가 배후 서버 (reverse server)로부터 데이터를 가져오는 역할을 한다. 여기서 프록시 서버가 Nginx, 리버스 서버가 응용프로그램 서버를 의미한다. 웹 응용프로그램 서버에 리버스 프록시를 두는 이유는 요청(request)에 대한 버퍼링이 있기 때문이다. 클라이언트가 직접 서버에게 요청하는 경우, 프로세스 1개가 응답 대기 상태가 되어야만 한다. 따라서 프록시 서버를 둬으로써 요청을 배분하는 역할을 한다.

(4) **Collaborative filtering**: Collaborative Filtering은 사용자들의 아이템 선호에 따른 각각의 선호 list와 평점이 존재할 때, 즉 사용자별 정보에 따른 평점과 그에 대한 점수를 matrix화 하였을 때 결측치에 대한 값들을 다른 사용자들 간의 유사도를 계산하여 평점을 예측하는 방법이다. Findmap은 우선 직접 구현한 user based item filtering model로 DB의 사용자 정보에 따라 검색 category의 matrix 화를 진행한다. 그다음 matrix 간의 cosine 유사도를 통해서 검색 카테고리 간의 연관성을 추출한다.

(5) **Content based filtering**: db에 충분한 정보가 없을 경우인 cold start 문제를 대처하기 위해 처음 추천을 하는 데 사용되었다. 이는 FastText skip-gram을 사용하였다. Skip-gram 알고리즘은 단어를 tokenize 시킨 후 이를 각각 one hot vector로 나타낸다. 각각의 주변 단어를 통해 특정 단어를 타겟팅하는 c-bow 방식과는 달리 그 단어 자체의 주변 단어를 통해서 한 단어에서 여러 파생단어를 예측하여 유사단어를 추출하는 데 적합하므로 skip-gram을 사용하였다. 따라서

사용자가 처음 검색을 하였을 시 비슷한 유형의 유사 추천 검색어를 받아 볼 수 있다. 또한, categorize를 통해 db에 사용자 선호도를 업데이트하면서 차후 db에 특정 수 이상의 정보가 누적될 시 collaborative filtering을 통하여 추천할 수 있는 기반을 만든다.

(6) 게시글 자동 카테고리(Fasttext): 범주화 라이브러리로 채택한 FastText는 Word2Vec과는 달리, n-gram을 사용하여 하나의 단어 안에도 여러 단어가 존재하는 것으로 간주한다. 이로 인해 모르는 단어나 등장 빈도가 수가 적은 단어에 대해서도 임베딩의 정확도가 높다는 장점이 있다. 범주화에서는 FastText의 Text Classification 모델을 사용한다. FastText의 Text Classification 모델은 CBOW 모델의 구조와 유사하지만, 문장을 bag of words 대신 bag of n-gram으로 나누어 효과적으로 학습한다. 마지막에는 hierarchical softmax를 사용하여 predefined classes에서 probability를 계산하고, 결과를 도출한다. Text representation의 dimension이 h , class의 수가 k 라고 했을 때 기존의 softmax의 경우 연산 복잡도가 $O(kh)$ 이지만, hierarchical softmax를 사용하여 연산 복잡도를 $O(h\log_2(k))$ 까지 줄였다. 위의 특징으로 인해 FastText의 Text Classification model은 기존보다 빠른 속도로 모델을 학습 및 결과를 도출하며, n-gram으로 word embedding을 하기에 Fasttext 모델을 선정하였다.

(7) Flask: Python 환경에서 서버를 구축할 수 있는 프레임워크다. 추천 기능과 여러 사용자 편의 기능을 제공하기 위해 AI 모델을 학습하는데 pytorch를 이용하였기 때문에 이를 실행할 수 있는 환경이 필요하였고 Flask를 사용하게 되었다.

(8) GitLab Runner: CI/CD는 자동 테스트 및 배포 환경을 만드는 것을 의미한다. 이를 위해 dev branch에 commit 할 때마다 test 코드를 실행할 수 있도록 gitlab에서 제공하는 CI/CD tool을 사용하였다.

(9) Docker: 의존성 충돌을 막고 개발의 편의성을 위해서 ML Server, API Server, Proxy Server를 독립적으로 만든다. container 단위로 만들어 환경을 구성하였고 이후에 docker-compose를 이용해 다중 container를 다루기 더욱 유리하기 때문에 모든 개발 환경을 docker 위에서 구성하였다.

(10) Docker-compose: 여러 개의 docker container를 한 번에 관리하도록 도와주는 툴로 하나의 커맨드로 모든 실행 컨테이너를 여닫을 수 있으며, 로컬로 각 container를 연결할 수 있다. 이를 통해 Proxy Server만 외부 포트와 연결하고 나머지 container는 모두 접근을 막아서 보안에 강점을 보인다.

(11) AWS Server: AWS EC2는 사용자가 서버를 쉽게 만들 수 있도록 도와준다. GPU 서버를 사용하기 위해 오하이오주의 GPU 인스턴스를 사용하였고 모든 서버는 사용자를 추가로 생성해 각 사용자가 ssh로 접속하여 개발할 수 있도록 하였다.

(12) Cloud DB(AWS RDS): AWS RDS는 관계형 데이터베이스를 더 쉽게 설치, 운영 및 확장할 수 있는 웹 서비스이다. 서버와 데이터베이스를 분리해서 데이터를 쉽게 잃어버리지 않도록 했다.

4. 작품 개발 환경

구분		상세내용
Server 개발환경	OS	macOS, Ubuntu 18.04
	개발환경(IDE)	IntelliJ, DataGrip, MySQL workbench, VSCode
	개발도구	NodeJS, NginX, Flask
	개발언어	JavaScript, Python3, MySQL
	기타사항	세 개의 독립적인 서버 환경을 구축하고 Docker-compose로 연결
AI 개발환경	OS	Windows, Ubuntu 18.04
	개발환경(IDE)	Google Colab, Jupyter Notebook, Pycharm, VSCode
	라이브러리	Pytorch, FastText, Pandas, Pororo
	개발언어	Python3
	기타사항	외부 라이브러리 모델과 자체 개발한 모델을 혼용하여 사용
Mobile 개발환경	OS	Windows, macOS
	개발환경(IDE)	Android studio
	개발언어	Dart, Kotlin, Swift
	기타사항	-
프로젝트 관리환경	형상관리	한이름 Gitlab
	의사소통관리	Coda, Google Meets, Zoom, Jira
	기타사항	학기 중에는 주1회, 방학에는 주2회 온라인 미팅을 진행함

5. 기타 사항 [본문에서 표현되지 못한 작품의 가치(Value)] 및 제작 노력

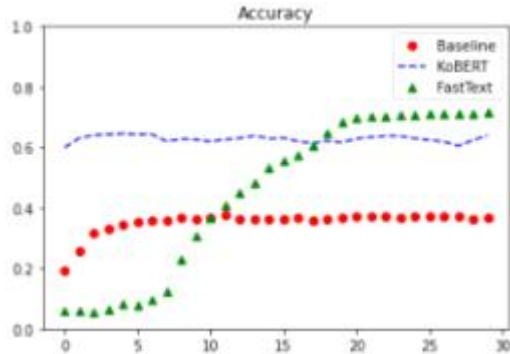
1) 추천 시스템 모델 개발

애플리케이션에 필요한 추천 시스템을 **직접 개발**함으로써, 사용자와 카테고리에 대한 feature를 matrix화 하고 cold start문제에 대응하기 위해 사용자와 카테고리에 대한 유사도를 cosine matrix로 상관관계를 추출하여 사용자의 취향에 맞게 검색결과를 정렬해주는 추천 시스템 모델을 개발하는 데 성공하였다. 아래 표는 사용자의 관심분야에 따른 검색어 ‘레드벨벳’의 Top-3 검색결과를 보여주고 있다.

관심 분야	검색 결과 (Top-3)
공연/전시, 방송	1. 내 곁에 미술관 HAPPINESS 후기 레드벨벳 슬기와 함께하는... 2. 내 곁에 미술관 SLEEP 후기 레드벨벳 슬기 미술책 추천해요 3. 온앤오프-레드벨벳-로꼬... MZ세대 겨냥한 DJ들의...
스타/연예인	1. 연예인 평판 1위는 방탄소년단... 2위 임영웅 3위 블랙핑크 2. [단독] 레드벨벳 조이♥크러쉬, 핑크빛 열애... ‘고막커피... 3. 이무진의 '신호등', 멜론 주간 국내종합 1위... 아이유 8곡, 임영웅 곡
요리/레시피, 상품리뷰	1. 대전 관저동 :: 윤달/컵케익(레드벨벳, 딸기) 2. [광주 운암동] 레드벨벳 컵케이크 맛집 아담한 카페 버터플레이버 3. 큐브, 레드벨벳 크림 치즈 케이크, 부드러운 생크림 카스텔라...
사진, 패션/미용	1. 레드벨벳 예리 인스타그램 여성 원피스 코디, 연예인 옷... 2. 레드벨벳 웬디 귀걸이 추천 에끌라스튜디오로 여자 가을 코디에... 3. 레드벨벳 쿤덤 슬기 솔더백 가방 패션 @마지셔우드

2) 다양한 NLP 라이브러리 비교

Categorize를 위한 NLP process에서 다양한 라이브러리를 사용해보고 가장 최적화된 것을 사용하였다. Kakao brain에서 만든 Pororo, KoELECTRA, FastText를 사용해보며 정확도 및 작동시간을 비교하여 우리의 서비스에 알맞은 라이브러리를 선정하였다. Baseline은 LSTM으로 만든 간단한 모델이다.



모델별 epoch에 따른 정확도 비교

모델	Epoch에 따른 정확도(%)		
	10 epoch	20 epoch	30 epoch
Baseline	36.33	36.63	36.49
FastText	30.74	64.83	71.51
KoBERT	62.91	62.86	68.37

또한, FastText를 사용하는 데 있어서 직접 32개의 category에 해당하는 65000개 이상의 데이터를 수집 및 라벨링하고, 지도학습으로 모델을 만들어 사용하였다.

3) 보안

(1) JWT 사용

JWT는 Json Web Token의 약자로 클라이언트가 필요한 정보를 토큰 body에 저장해서 클라이언트가 가지고 이를 증명서처럼 사용할 수 있다. JWT를 사용함으로써 사용자는 쿠키를 전달하지 않아도 되기 때문에 쿠키로 인해 발생하는 보안적 취약점을 보완할 수 있었다.

(2) Token 처리

WT는 secret key가 노출된다면 토큰을 뺏길 수 있고, 발급된 토큰에 대한 제어가 불가능하다는 새로운 보안 문제가 있었다. 그래서 서버에서는 토큰 안에 중요한 사용자 정보는 담지 않도록 한 후 사용자 정보를 서버 내에서 한 번 더 검증하는 방식을 이용했다. 또한, 토큰의 유효기간을 지정한 후 유효기간이 긴 토큰과 짧은 토큰을 발급해서 사용하며 사용자의 불편함을 줄이면서 보안성을 높이기 위해 노력했다.

(3) 암호화

사용자가 로그인할 때의 비밀번호는 해시 방식을 통해 암호화했고, 사용자가 입력한 비밀번호는 DB에 저장되지 않도록 했다. 프로젝트에서의 중요한 JWT secret key, API 접근 key 값, 서비스 메일 비밀번호 등의 중요한 값들은 외부에 노출되지 않도록 코드를 통해 암호화 처리하였다.

4) 사용성

사용자 친화적인 UI/UX를 개발하고 처음 애플리케이션을 사용하는 사용자도 어렵지 않게 사용할 수 있도록 고려하여 디자인하였다. 검색 버튼을 누르면 검색창이 생김과 동시에 자판이 올라오거나, 아이디, 비밀번호 입력에는 바로 영문 자판이 뜨고, 숫자를 입력해야 할 때는 숫자 자판이 먼저 활성화되도록 하는 등 세세한 곳에 신경을 기울였다.

5) 유지관리

CI/CD를 적용할 수 있도록 Gitlab-runner를 만들어놓고 test code를 작성할 시 자동으로 test 될 수 있는 환경을 구축해놓았다.

III. 프로젝트 수행 내용

※ 평가항목 : 수행능력 (문제해결능력, 수행충실성)

1. 프로젝트 수행 일정

프로젝트 기간 (ICT멘토링 사이트 기준)		2021.03.30. ~ 2021.11.30.										
구분	추진내용	프로젝트 기간										
		3월	4월	5월	6월	7월	8월	9월	10월	11월	12월	
계획	주제 선정, 기술 설정											
분석	타 서비스 분석, 시장 분석, 유사기술 조사, 사용자 스터디											
설계	사용자 시나리오 설정, Affinity diagram 설계, 시스템구조 정의											
	프로토타입 제작, 사용 데이터 종류 작성, DB schema 작성											
개발	로그인 환경, 스크랩 기능 구현 DB 구축, 데이터셋 구축(수집 & 전처리)											
	AI 모델 훈련 및 적용 피드, 검색, 마이페이지 기능 구현											
	Code refactoring, 최적화											
테스트	기능 테스트											
종료	애플리케이션 출시											

2. 프로젝트 추진 과정에서의 문제점 및 해결방안

1) 프로젝트 관리 측면

(1) 비대면 회의

코로나 19로 인하여 대면 회의가 불가능해지면서 원활한 소통이 어려웠다. 효율적인 프로젝트 진행을 위해 애자일 소프트웨어 개발 방법론을 택했는데, 효과적으로 스크럼을 진행하기 위해서는 daily scrum meeting을 진행해야 했다. 하지만 비대면 회의로 진행하면서 대면보다는 소통이 원활하지 않다는 생각이 들었고, 팀원들과 회의 시간을 맞추는 것이 어려웠다는 것 또한 문제였다.

그래서 일주일에 한 번씩은 미팅을 고정적으로 진행하도록 규칙을 정했고, 방학 기간에는 일주일에 회의를 두 번씩 진행하며 소통을 원활히 할 수 있도록 했다. 또한, 회의 진행 도구로 Notion, Jira, Coda를 활용하며 회의한 내용을 바로 정리하고 공유하며 비대면 회의의 강점을 최대한 끌어올렸다. 문제 상황이 생겼을 때는 단독방에서 바로바로 공유하며 팀원들과 함께 해결할 수 있도록 했다. 추가로, 회의를 진행할 때마다 gitlab에서 pull request를 통해 작업한 코드를 공유하고 피드

백하는 시간을 가졌다.

2) 작품 개발 측면

(1) 추천 시스템 데이터 구축

프로젝트에서는 사용자 맞춤 콘텐츠를 추천해주기 위하여 추천 시스템 기술을 사용한다. 해당 기술은 collaborative filtering과 content-based filtering을 사용한다. Collaborative filtering, content-based filtering 둘 다 사용자가 좋아할 것들을 추천해주기 위해서는 타 사용자의 선호도나, 아이템 간의 유사도 등 사전 데이터가 필요하다. 그러나 인터넷에 공개된 한국어 데이터셋만으로는 우리가 원하는 추천 시스템 모델을 학습시키는데 필요한 데이터의 양에 미치지 못한다. 또한, 애플리케이션 서비스 초반에는 사용자 데이터가 충분히 모이지 못한 상태이기 때문에 cold start 문제가 발생할 수 있다. 따라서 데이터가 충분하지 못한 초기 상황에는 간단한 방식을 이용해 사용자가 선호하는 카테고리를 선정하고 해당 카테고리에 해당하는 아이템을 추천하고 일정 데이터 이상이 모이면 추천 시스템을 사용하는 등 **hybrid 추천 시스템 방식**을 사용하였다.

(2) NLP 데이터 구축

콘텐츠의 분류를 AI화 하기 위해서는 콘텐츠와 해당 콘텐츠가 어떤 카테고리에 속하는지 라벨이 있는 데이터가 필요하다. 하지만 아직 이러한 **조건에 걸맞은 데이터셋은 존재하지 않기 때문에 직접 크롤링을 통해 구축**하였다. 네이버와 다음의 블로그들을 크롤링하며 해당 블로그의 카테고리를 함께 크롤링하였다. 하지만 대부분 글은 적절한 카테고리가 선정되어 있지 않아 해당 데이터를 그대로 사용하면 학습이 제대로 되지 않았다. 따라서 **크롤링한 데이터의 라벨을 손수 수정**하는 과정을 거쳤다.

(3) 연산 속도

모델	속도(초)
Baseline	2.321
FastText	0.183
KoBERT	9.832
Pororo	1920.826

프로젝트 진행을 위해 카테고리를 분류하는 모델이 필요하였다. Kakao Brain에서 제공하는 오픈소스인 Pororo 중 zero-shot classification을 활용하여 카테고리 분류를 시도하였으나, test 파일을 돌렸을 때 1. 정확성이 매우 낮고 (약 15%) 2. 연산 속도가 매우 느리다는 문제점이 있었다(3200개 테스트셋 기준 약 1920초). 또한 동일 조건상에서 각각 LSTM, KoBert, Fasttext를 사용하여서 text classification 과정을 진행하였다. 그 결과 Fasttext의 속도가 가장 빠른 속도를 보여주고 이에 따라 Fasttext를 주 모델로 사용하게 되었다.

(4) 정확도

모델	정확도(%)
Baseline	36.49
FastText	71.51
KoBERT	68.37
Pororo	15

정확도 또한 Pororo를 사용했을 때보다 높게 나왔지만 (약 50%), 실용화하기에는 다소 부족한 수치였다. 모델의 정확도에 영향을 미치는 것으로 알려진 요소는 매우 다양하다. 본 프로젝트의 경우는 데이터셋의 크기가 상대적으로 적어 데이터의 질이 상당히 중요한데, 데이터의 질이 고르지 않다는 것이 낮은 정확도의 원인이었다. 원하는 카테고리 분류 모델을 학습시키기 위한 데이터셋을 찾아보았으나 인터넷에 공개되어있는 데이터가 없었다. 이에 데이터 크롤링을 이용하여 데이터셋을 구축하였는데, 인터넷 글 특성상 카테고리가 불분명하거나, 카테고리가 잘못 설정된 경우가 다수여서 데이터의 질이 떨어진다는 것이 가장 큰 문제였다. 따라서 데이터의 질을 높여주기 위하여 데이터를 하나하나 확인하면서 카테고리를 다시 분류해주었고, 이에 현재 **72.4%의 정확도**로 큰 향상을 끌어낼 수 있었다.

(5) GPU 서버

해당 프로젝트는 고용량의 연산을 요구한다. 그 때문에 GPU가 필수적으로 필요한데, GPU 서버의 가격은 일반 서버보다 매우 비쌌다. 따라서 팀원의 GPU 컴퓨터를 이용해 Windows의 WSL2 환경에서 직접 서버를 구축하였다. 팀원들은 SSH로 해당 컴퓨터에 접속할 수 있고 바인딩 규칙과 port forwarding으로 포트를 열어 어느 곳에서든지 http로 request를 할 수 있게 되었다.

(6) Docker

프로젝트를 진행하기 위해 간단한 API 구성이 가능한 Node.js 서버, 머신러닝 동작을 위한 Flask 서버, 트래픽을 처리하기 위한 Nginx 등 여러 언어와 프레임워크를 사용해서 개발을 진행해야 했다. 프레임워크마다 서버를 구성할 경우 트래픽이 많이 발생하게 된다. 따라서 가볍고 빠른 서버 환경 구성을 위해 Docker를 사용했고, 모든 환경 구성 요소가 상호작용하며 원활히 동작하도록 했다. 이외에도 모든 개발이 끝나고 실제로 서비스를 돌릴 때는 AWS 서버로 이전해야 하므로 모든 개발 환경 및 동작 환경을 그대로 옮길 때 한 번에 쉽게 옮기는 경우도 고려하였다.

3. 프로젝트를 통해 배우거나 느낀 점

- 1) ICT 분야에서 오랫동안 프로젝트를 진행해온 멘토님의 조언과 한이음 멘토링의 가이드를 통해 서비스 설계부터 배포까지 전 과정을 아우르는 경험을 할 수 있었다. 이러한 과정을 통해 소프트웨어 공학적인 관점을 가지고 devops를 제대로 지키는 현업과 비슷한 방식의 개발 프로세스를 배워볼 수 있었다.
- 2) 개발을 시작하기 전에 Backlog를 작성하고 sprint마다 개발할 수 있는 정도의 계획을 세우는 것이 매우 중요하다는 것을 느꼈다. 과도한 개발 일정은 계속 지체되고 제대로 개발이 이루어질 수 없음을 느끼고 다음 개발에는 더욱 세세하고 정확한 개발 일정을 세우려고 노력할 것이다.
- 3) 반년간의 장기 프로젝트를 진행하면서 효율적인 개발 성과를 내기 위한 팀원들과의 소통의 중요성을 느꼈다.
- 4) AWS, MySQL Workbench, Pytorch등 많이 사용되는 상용 프로그램과 소프트웨어를 사용해보고, 직접 Flutter와 Node.js 프레임워크로 개발하면서 개발실력을 올리고 실무와 비슷한 개발을 경험해볼 수 있었다.
- 5) 오픈소스 라이브러리를 사용해보면서 라이선스 및 오픈소스의 중요성을 깨달았으며 팀원들 또한 오픈소스에 이바지하고자 하는 다짐을 할 수 있었다.
- 6) 현재 산업에서 가장 수요가 많은 인공지능, 빅데이터 분야는 플랫폼 사업에서부터 임베디드, 다양한 정부 사업 등 많은 분야에서 해당 기술을 요구한다. 애플리케이션에 필요한 추천 시스템을 구현해보면서 인공지능 혹은 빅데이터 분석 기술을 접해보고 구현해볼 수 있었다. 이를 통해 인공지능, 빅데이터의 이점과 필요성, 성장 가능성 등을 인지할 수 있었다. 또한, 사회에서 발생할 수 있는 문제점을 해결하는 방식 중 인공지능과 빅데이터라는 새로운 해결방식을 인식했고 대학뿐 아니라 인공지능과 관련한 연구, 산업 분야에 진출할 기회를 만들 수 있었다.
- 7) 학교 수업에서는 높은 정확도를 지닌 인공지능 모델을 제작하는 것에만 집중했는데, 한이음 프로젝트를 통해 인공지능 모델을 학습시키는 것에서 나아가 학습시킨 모델을 직접 활용하는 애플리케이션을 제작해보니 인공지능의 활용성과 중요도를 더욱 깊게 체감할 수 있었다.
- 8) Collaborative filtering, content based filtering을 공부하며 직접 개발해봄으로써 SVD, 피어슨 유사도와 같은 여러 정보에 대해 배울 수 있어서 많은 도움이 됐다.

다. 또한, 처음 개발하였을 시 user data 정보의 부재에 따른 검증의 어려움을 느꼈고 그로 인하여 user data가 얼마나 중요한 자료인지 다시 한번 느끼게 되었다. 그리고 더 나아가서 처음 data에 대한 정보가 없는 상태 즉 cold start일 때의 문제를 어떻게 해결해야 하는지 사색해볼 수 있었다. 그리고 그에 따른 확장적인 다른 추천 시스템에 관해서도 흥미가 생겼다.

IV. 작품의 기대효과 및 활용 분야

※ 평가항목 : 기획력 (활용 가능성)

1. 작품의 기대효과

1) 높은 시장성

Findmap과 가장 비슷한 기존 서비스는 Pocket이 있다. Pocket은 모바일 기준 292.4K의 이용자 수를 보유하고 있으며 뉴스, 스토리, 동영상 등을 스크랩하는 서비스이다. 많은 부분이 Findmap과 닮아있지만, 대부분 콘텐츠가 영어로 이루어져 있으므로 한국에서 사용하기는 부적합하다. 하지만, 사용자 수에서 콘텐츠를 스크랩하며 소비하는 서비스의 시장성이 얼마나 가치가 높은지 알 수 있다.

2) 차별성

Pocket과 비교되는 두드러지는 차별점은 크게 세 가지이다. 첫 번째로 대부분 콘텐츠가 한글로 이루어져 있다는 것이다. Findmap의 타겟은 스마트폰으로 검색을 하는, 한국어가 모국어인 모든 연령층이기 때문에 콘텐츠가 한국어로 만들어져야 한다. 따라서 모든 콘텐츠를 다음과 네이버에서 가져온다. 따라서 국내의 대부분 사용자가 불편함 없이 콘텐츠를 즐길 수 있을 것이다. 두 번째 차별점은 개인화 추천 서비스이다. Findmap에서 검색하면 사용기록을 이용해 비슷한 검색어를 추천해주고, 검색 결과를 사용자 취향에 맞게 재정렬하여 보여준다. 마지막 차별성은 실시간 검색어의 제공이다. 지난 2020년 1월, 전국 성인남녀 1,000명을 대상으로 ‘실시간 검색어’ 관련 인식을 조사한 결과(출처: 엠브레인) 약 75.4%가 실시간 검색어를 확인하는 습관이 있고, 약 66.2%가 실시간 검색어 서비스가 필요하다고 답했다. 이처럼 네이버와 다음에서 실시간 검색어가 사라진 이후로 많은 소비자는 실시간 검색어의 부활을 기다리고 있다. Findmap은 검색 기능이 주요 기능인 애플리케이션이기 때문에 실시간 검색어 기능을 서비스로 제공하면 사용자의 만족감을 높일 수 있을 것이다.

2. 작품의 활용 분야

- 1) 검색어 추천: Findmap에서 훈련한 AI를 이용해 사용자가 검색어를 입력하면 연관된 검색어를 추출하고 다시 사용자의 취향으로 재정렬하여 마인드맵 형태로 제공한다. 이를 통해 평소 검색보다 더욱 넓은 범위의 검색이 가능하다.
- 2) 실시간 검색어: 네이버와 다음이 실시간 검색어를 폐지함에 따라 실시간으로 알아야 하는 정보나 원하는 정보를 놓치는 사용자가 많아졌다. Findmap에서는 실시간 검색어 기능을 제공하기 때문에 사용자는 최신 정보를 놓치지 않고 볼 수 있다.
- 3) 맞춤형 검색 결과: 검색 결과를 사용자 취향에 맞게 정렬하여 사용자에게 보여준다. 이를 이용해 사용자는 원하는 검색 결과를 더욱 빠르게 찾을 수 있다.
- 4) 맞춤형 게시물 추천: 흥미롭고 재미있어할 만한 글을 사용자가 직접 찾지 않아도 Findmap에서 추천해주기 때문에 관심 있는 분야의 새로운 정보를 쉽고 빠르게 접할 수 있다.
- 5) 스크랩: 나만의 아카이브에 원하는 글을 스크랩하여 언제든지 다시 쉽게 볼 수 있으며, 폴더 기능으로 관리를 체계화된 스크랩 관리를 할 수 있다.
- 6) SNS: 스크랩한 콘텐츠를 공유하고, 다른 사람을 팔로우할 수 있다. 정보를 다른 사람과 공유함으로써 콘텐츠를 주도적으로 생산하고 소비할 수 있다.