



방제형 포트폴리오



기술 역량을 확장하는 것에 많은 관심을 갖고 있습니다. 배움을 좋아하는 개발자입니다.

연락처

tfzangin9@gmail.com

github.com/BJH9

교육

한동대학교 AI컴퓨터공학심화(공학인증) 학사

2017년 3월 입학

2023년 8월 졸업

네이버 부스트캠프 수료

GPT, BERT 등 LLM 모델에 대해 심화 학습, LLM 모델 파인 튜닝, AI 기반 웹 서비스 개발

2023년 11월 시작

2024년 4월 수료

웹 서비스 개발 캠프 수료

React 기반 프론트엔드 개발, Spring Boot 기반 백엔드 개발 역량 강화, Git flow 전락으로 Git 협업

2022년 12월 시작

2023년 1월 수료

경력

소프트웨어공학 연구실 ISEL 인턴

(2021년 03월 - 2021년 12월)

- '머신러닝 어플리케이션에 대한 결함 사례 조사' 논문 작성
- 위 논문, KCSE 2022 학술대회 최우수 논문 선정
- 프로그램 자동 수정 기술과 결함 예측 기술 논문 연구

신태양 특허법인 인턴

(2021년 9월 - 2022년 1월)

- (주)라쉬반 기업의 의류 특허 기술 및 청구항 분석
- 키프리스 톨 기반 선행 특허 조사
- OS 매트릭스 작성
- (주)라쉬반 특허 전략 보고서 작성, 비즈니스 문서 작성

논문

'오픈소스 머신러닝 어플리케이션에 대한 결함 사례 조사'

소프트웨어 공학 소사이어티 2022 학술대회 논문 최우수상


(2022년 01월 19일)


기술 스택

- 백엔드

 [Java](#)

 [Spring](#)

 [스프링부트](#)

 [스프링 시큐리티](#)


 [MySQL](#)

 [JPA](#)

 [Redis](#)

 [JUnit5](#)

- 프론트엔드

 [Javascript](#)

 [React](#)

 [HTML](#)

 [CSS](#)

- 기타


 [Git](#)

 [Github](#)


 [Docker](#)

- AI 및 백엔드

 [Python](#)

 [FastAPI](#)

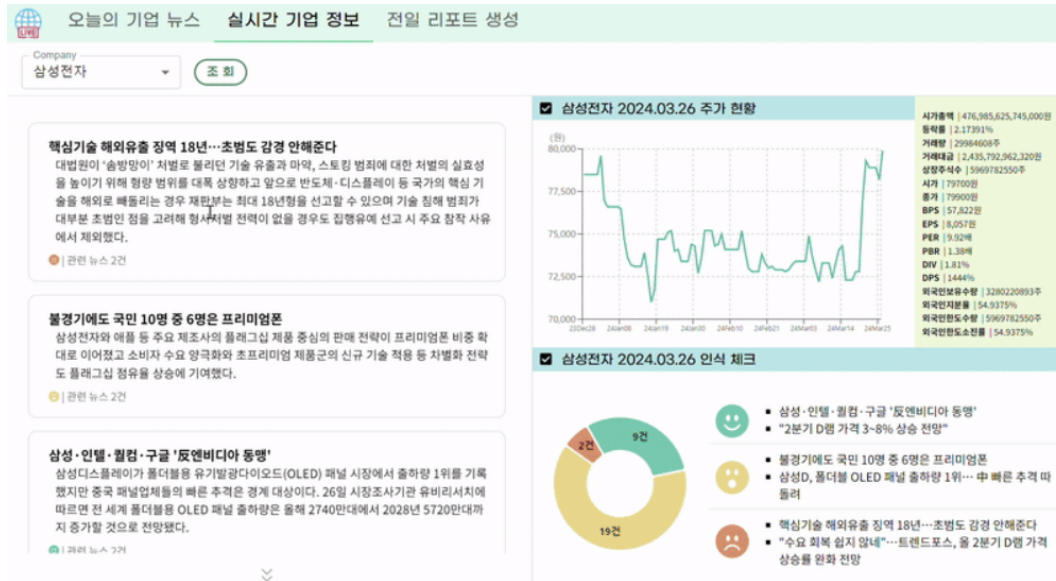
 [Pytorch](#)

 [Huggingface](#)

수상

- (주)Bizflow 노코드/레스코드 개발 툴 AppDev 기반 앱 개발 콘테스트 **우수상**
- 소프트웨어 공학 소사이어티 2022 학술대회 논문 **최우수상**
- 캡스톤 페스티벌 (공간 대여 플랫폼 웹 서비스 개발) **인기상**
- 문장 내 개체 간 관계 추출 AI 모델링 대회 **3등**
- MRC AI 모델링 대회 **4등**

Advanced Fin GPT: AI 기반 주식 보조 웹 서비스



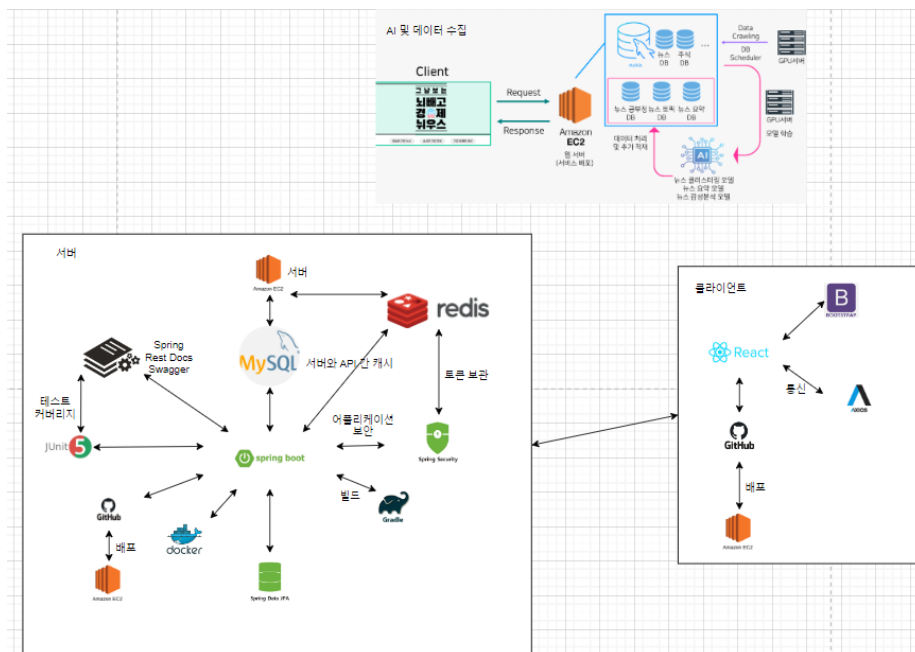
설명

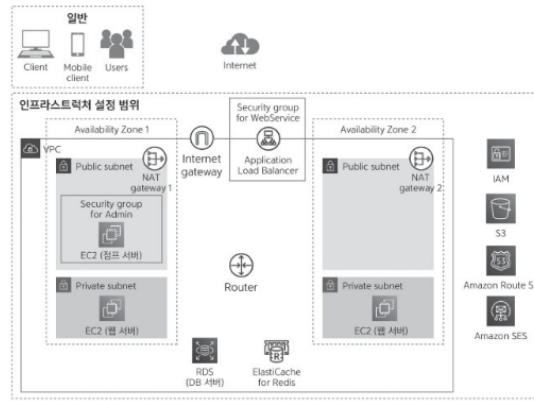
- 뉴스 정보가 대량으로 쏟아져 나와 주식 투자자가 정보 선별에 어려움을 겪는 문제 발견
- AI 기술을 활용해, 중복과 유사한 뉴스를 줄이고, 중복 별로 중요한 정보만 선별하여 제공

역할

- 백엔드 개발

아키텍처





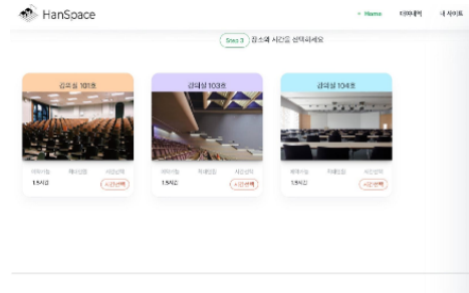
Challenge

- FastAPI 코드를 Spring Boot 코드로 업데이트하는 과정에서 작은 백엔드 로직 오류 발생
→ 설계, 테스트, 문서화의 중요성 인식, Junit5를 활용하여 테스트 코드 작성, 테스트 코드 커버리지 65% 달성, Spring API docs(Swagger)를 활용하여 API 설계 및 문서화
- DB에 유저 데이터를 그대로 저장하여, 유저 보안이 취약한 문제점 발견
→ Spring Security 구현으로, Filter layer에서 request와 response에 대해 JWT를 발행하도록 하여 보안 향상, 로그아웃할 때 토큰을 Redis에 블랙리스트로 저장하도록 구현하여 보안 강화
- 종목 별 뉴스를 실시간으로 수집하기 때문에 시간이 갈수록 DB의 뉴스 데이터 크기가 방대해지는 문제, DB 탐색 속도가 저하되는 문제
→ Redis를 캐시 서버로 활용할 수 있도록 구현, REST API에 Redis Repository 및 Redis Template 적용, DB 탐색 속도 약 400% 향상, AWS 환경 로드 밸런서 구축
- 로컬 환경과 배포 및 운영 환경을 일관성 있게 구현할 필요가 있다는 점을 발견
→ Docker image 생성 및 배포, Docker container, Docker compose로 환경의 일관성 확보, AWS 인프라스트럭처 구축(VPC, IAM, S3, NAT, gateway, RDS, ElasticCache, ...)

회고

- 공학적 설계 및 개발의 중요성 인식
 - 구현을 넘어 아키텍처를 설계하고 유지보수하는 역량이 매우 필요하다는 것을 배웠습니다. FastAPI로 작성된 백엔드 로직을 Spring Boot 로직으로 바꾸는 과정에서, 많은 오류를 마주하며 어려움을 겪었습니다. 설계, 테스트, 문서화의 과정이 있었다면, 더 안전하고 빠르게 코드를 바꿀 수 있었으리라 생각했습니다. RestTemplate, Mock, Junit5 등을 활용하여 테스트 코드를 작성했고, Spring API Docs를 적용하며 테스트 커버리지 수행 및 API 문서 작성 자동화를 이루었습니다.
- Spring Security에 대한 이해
 - 인증, 인가 등의 과정에서 보안을 강화하기 위해 Spring Security를 적용했습니다. Spring Security는 어떤 과정을 거치는지, 어떻게 보안을 강화하는 지 이해할 수 있었고, 구현 역량을 발전시킬 수 있었습니다. 또한 만료 기간을 이용하여 토큰을 탈취하는 것을 방지하기 위해, 로그아웃 시 Redis 캐시 서버에 블랙리스트로 해당 토큰의 정보를 저장하도록 구현하며 JWT 토큰 방식에 대해 더 이해할 수 있었습니다.
- DB 탐색 속도 향상에 대한 중요성 이해 및 역량
 - 실시간 뉴스 데이터를 DB에 저장하는 과정을 거치기 때문에 해당 서비스는 빅데이터를 처리할 수 있어야 합니다. 종목 별 뉴스 요약 정보 Get 처리 시간이 오래 걸리는 문제로부터, DB 탐색 속도에 대한 고민이 시작되었습니다. 해당 API는 동일한 데이터에 대한 접근이 잦다는 것을 발견했고, Redis를 통해 문제를 효과적으로 해결할 수 있을 것이라고 판단했습니다. Redis를 API와 서버 간 캐시로 사용함으로써 문제를 해결할 수 있었습니다.

Hanspace: 공간 대여 플랫폼 웹 서비스



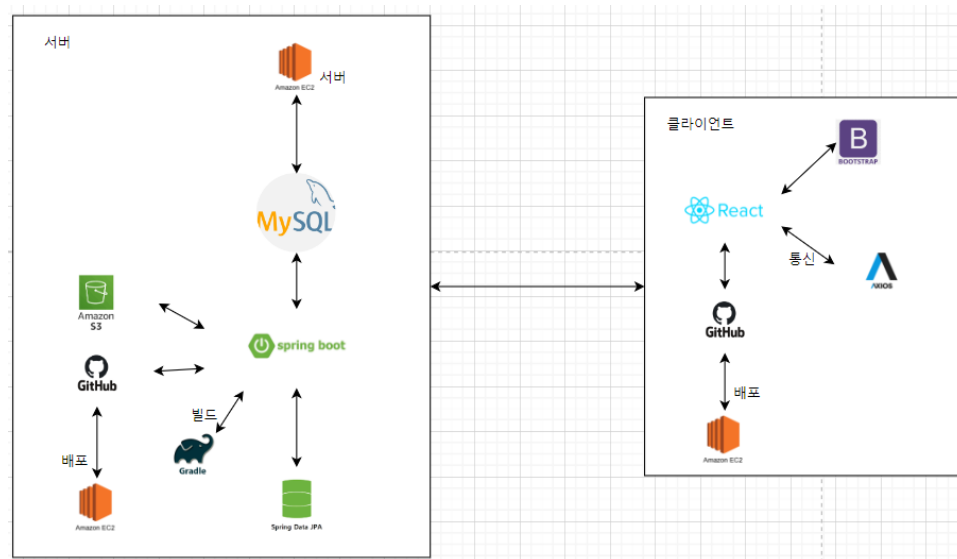
설명

- 공간을 대여할 수 있는 서비스의 부재, 기존 서비스는 호텔 등 특정 카테고리에만 치중되어 있는 문제
- 공간 대여 서비스를 개설할 수 있는 플랫폼을 웹 형태로 구현하여 문제 해결

역할

- RDB 설계, 백엔드 개발, 프론트엔드 개발

아키텍처



Challenge

- 백엔드 로직 개발 담당
 - RDB 설계, Spring Boot와 JPA 기반으로 REST API 구현, Collections 등 문제를 효과적으로 해결할 수 있는 알고리즘과 라이브러리를 적용해 비즈니스 로직 효율화
- 개발 자원 효율화를 위해 풀스택 개발 요구
 - React를 기반으로 대여 기록 페이지 및 공간 대여 페이지 구현, React 라이브러리, 상태 관리, 세션 관리, 컴포넌트 관리 기술 반영, 백엔드 API를 비동기 함수로 연동
- DB 설계 실수로, '공간 정기 대여' 기능을 추가했을 때 하나의 Create에 대해 2개 이상의 tuple이 생성되는 문제 발견
 - RDB의 정규화, 비정규화와 테이블 간 관계를 면밀히 고려하여 DB 재설계, DB '예약 테이블'의 용량을 약 50% 이상 절감, 탐색 속도 향상
- 고객이 대여 과정에서 불편함을 느끼는 문제 발견
 - 주기적인 설문문을 통해 고객과 소통하며 고객이 어떤 부분에서 왜 어려움을 느끼는지 끊임 없이 요구 사항을 분석, UI/UX 개선, API 기능 추가, 페이지 추가 등의 방안으로 초기 설문 만족도 30%에서 최종 설문 만족도 90%로 향상

회고

- 제품의 최종 도착지는 고객
 - 제품을 1차적으로 완성했을 때에는 **고객의 만족도가 굉장히 낮았습니다**. 저희 개발팀은 고객이 무엇을 원하는지 보다 프로젝트를 만드는 것에만 관심을 가졌었고, 이 태도를 반성했습니다. **주기적인 알파 테스트와 설문**을 통해 **고객이 원하는 방향으로** 프로젝트를 발전시켜 갔습니다. **페이지의 UI/UX를 개선**하고, **고객이 필요로 하는 기능을 추가**했습니다. **최종 만족도를 90%까지** 높일 수 있었고, **37개의 프로젝트 중 102개로 가장 많은 투표를 받아** 캡스톤 페스티벌에서 **가장 인기 있는 제품으로 선정**되기도 했습니다. 고객의 요구사항을 치열하게 분석하고, 요구 사항에 합당한 기술을 적용하여 서비스를 완성해 가는 경험을 했습니다.
- 많은 시행착오를 겪으며 DB에 대한 이해 강화
 - **RDB를 설계할 때**, 편의성에 초점을 뒀서 정규화 원칙을 다소 희생했습니다. 그러나 새로운 API를 도입함에 따라 **DB 처리 비효율**이 발생하는 것을 발견할 수 있었습니다. 어떻게 하면 일관된 DB를 설계할 수 있을 지 고민하며 많은 시행착오를 겪었습니다. **테이블 별 의존도를 낮출 수 있는 방향으로 설계하여 문제를 해결**했습니다. 해당 과정을 통해 RDB의 테이블 관계에 대해 더 깊이 이해할 수 있었고, DB 문제를 다룰 수 있는 역량을 강화했습니다.
- 프론트엔드와 백엔드를 연동할 수 있는 역량
 - 처음에는 백엔드 개발자로 합류했지만, 팀의 프론트엔드 개발 역량이 부족하여 **프론트엔드 개발도 겸하며 프로젝트를 수행**했습니다. **React 초기 설정, 컴포넌트 단위 프론트엔드 개발, 상태 관리, 세션 관리 등에 대해** 이해할 수 있었습니다. 또한 **프론트엔드의 요청과 백엔드의 응답을 모두 개발해보며, 프론트엔드와 백엔드의 연동을 더 잘 이해**할 수 있었습니다.

Fin GPT: AI기반 주식 보조 웹 서비스



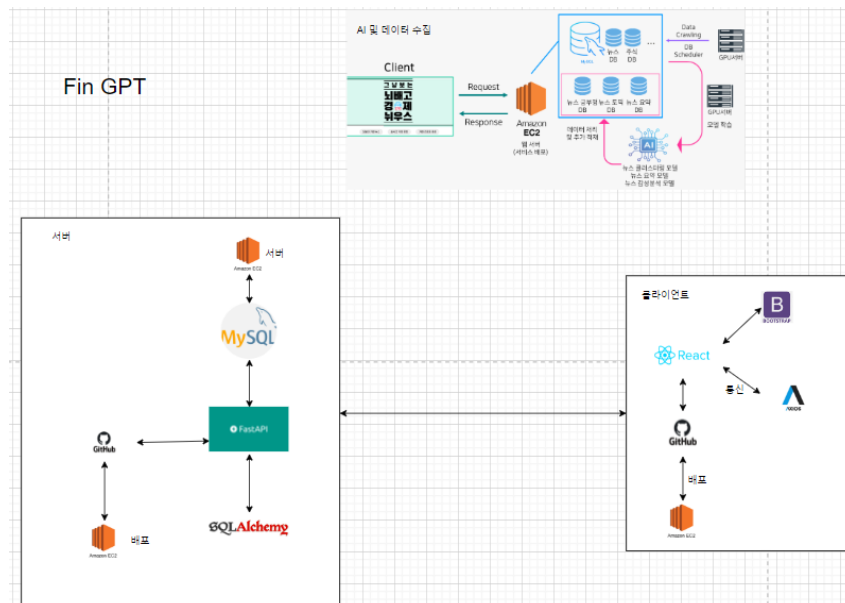
설명

- 뉴스 정보가 대량으로 쏟아져 나와 주식 투자자가 정보 선별에 어려움을 겪는 문제 발견
- AI 기술을 활용해, 중복과 유사한 뉴스를 줄이고, 종목 별로 중요한 정보만 선별하여 제공

역할

- DB 설계, 백엔드 개발, AI 모델 개발, 프론트엔드 개발

아키텍처



Challenge

- 모델 학습 및 DB를 위한 데이터를 수집해야 하는 문제
→ Python으로 네이버 뉴스 기사에 대한 데이터 크롤링, 전처리, 적재 코드를 구현, 데이터 파이프라인 구축
- 뉴스 요약 AI 모델과 원활한 연동을 이루는 웹 서비스 기능 필요
→ RDB 및 API 설계 후, AI 모델 배포에 유리한 FastAPI(Python)를 기반으로 REST API 구현, Pydantic을 통한 데이터 유효성 검사
- 고객의 뉴스 정보 판별을 돕기 위한 도구의 필요성
→ AI 감성 분석 모델 구현으로 뉴스의 긍정 부정 정보를 함께 고객에게 제공, AI HUB로부터 데이터 수집, 직접 라벨링 작업을 수행하여 데이터셋 품질 향상, Pytorch와 huggingface로 GPT 모델을 사용해 데이터를 증강하여 모델 성능 향상, AI를 웹 서비스 기능에 도입

회고

- 데이터 파이프 라인 구축 경험
 - 해당 서비스를 만들기 위해서는 직접 데이터셋을 구축할 필요가 있었습니다. Python으로 beautiful soup를 통해 네이버 뉴스 데이터를 직접 크롤링했고, 목적에 맞게 전략을 만들어 전처리했습니다. 해당 방식을 통해 백엔드를 위한 DB와 모델 학습을 위한 학습 데이터셋을 구축했습니다. 좋은 서비스와 모델을 만들기 위해 데이터에 대해 끊임없이 고민하는 과정을 거쳤고, 좀 더 성장할 수 있었습니다.
- AI를 웹 서비스에 도입한 경험
 - 이 프로젝트에서 가장 새로웠던 경험은 AI를 웹 서비스에 도입했던 경험입니다. Pytorch와 Huggingface를 통해 감성 분석 AI 모델을 만들었고, GPT를 통한 데이터 증강으로 모델 성능을 더욱 향상했습니다. 부족한 데이터는 직접 데이터 라벨링을 수행하며, 데이터셋의 품질을 개선하기도 했습니다.
- 새로운 기술에 두려움 없이 도전하는 태도
 - AI 모델링, 데이터 구축, FastAPI 기반 백엔드 개발 등, 새로운 기술적 요구들을 계속 마주했습니다. 그러나 겁 없이 자원하여 새로운 기술에 도전함으로써 기술 역량을 확장하며 성장했습니다. 이처럼 새로운 기술에 적극적으로 도전하는 태도가말로 저의 강점이라는 것을 깨달을 수 있었습니다.

NLP, LLM 모델 / 문장 내 개체 간 관계 추출 AI 모델 구축

순위	팀 이름	팀 멤버	micro_f1 ↕	aupec ↕	제출 횟수	최종 제출
3 (1 ▲)	NLP_10팀		75.4779	82.9782	132	4d
1	NLP_07팀		75.9857	83.5148	77	4d
2	NLP_06팀		75.9685	83.7397	58	4d
3	NLP_10팀		75.4779	82.9782	132	4d

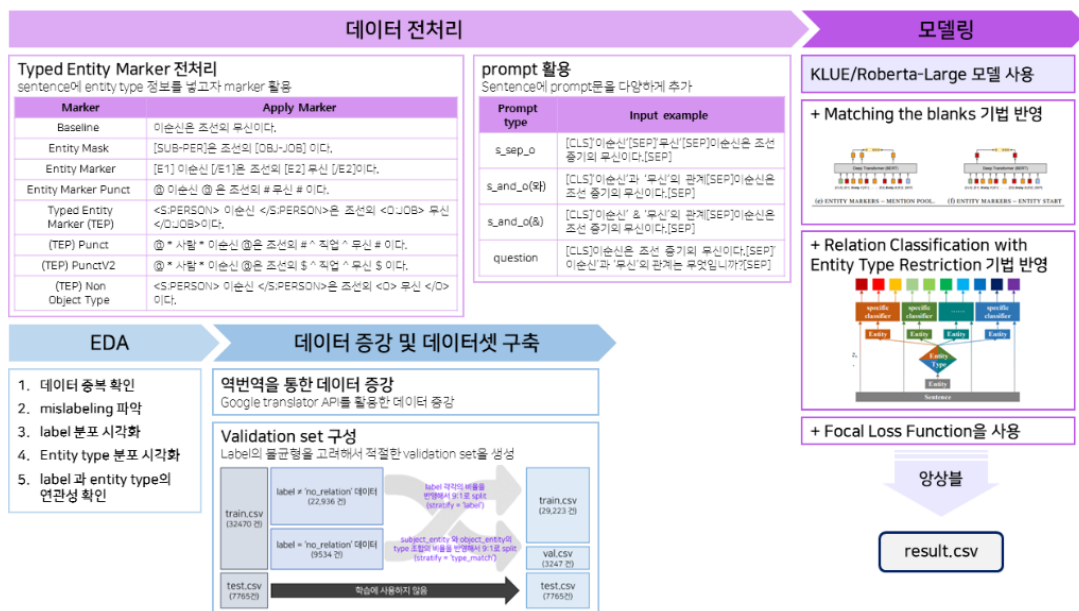
설명

- 주어진 데이터로 Relation Extraction (RE) 모델의 성능을 높이는 대회
- 32,470개의 학습 데이터
- 7,765개의 평가 데이터

역할

- 데이터 분석
- 데이터 전처리
- AI 모델링

프로젝트 구조도



Challenge

- 데이터의 특징을 파악하지 못해 성능 향상에 어려움이 있었던 문제
 - WandB와 Pandas로 데이터의 약점 분석 진행, 유사한 라벨을 모델이 혼동하는 문제와 Train data와 Test 데이터 간 라벨 분포가 많이 다른 문제 발견, 해당 부분에 대해 데이터 전처리와 증강 기법 적용으로 모델 성능 (F1-score) 65.12에서 67.25으로 향상
- 데이터 간 관계의 특징에 주목하지 못했던 문제
 - 관계 추출 Task에서 데이터 간 관계를 학습에 반영할 수 있는 논문 탐색, Entity Type을 Input에 포함시켜주는 논문 기법 적용으로 모델 성능 67.25에서 71.93으로 향상

- BERT 모델의 Input 특징에 주목하지 못했던 문제
→ **BERT 모델은 2개의 input을 받으므로, 추가 정보를 줄 수 있도록 활용하는 논문 탐색, Input에 요구 사항을 추가하여 모델 성능 71.93에서 73.28로 향상**
- 데이터 간 관계와 출력 결과 사이의 연관성에 주목하지 못했던 문제
→ **데이터를 분석했을 때, 특정 입력 데이터 관계로부터 도출될 수 없는 output 발견, 모델 후처리로 해당 output을 회피하여 결론을 도출하도록 모델 구조 변경, 모델 성능 73.28에서 73.91으로 향상**
- 대회 3등 달성

회고

- 데이터의 특징을 고려하여 모델의 성능 향상
 - 주어진 Domain의 데이터에서는 데이터의 어떤 점이 중요한 지 파악하고, 그 이해를 기반으로 데이터와 모델을 구축해야 한다는 것을 배웠습니다. 문장 내 개체 간 관계 추출 Task는 데이터 간 관계에 어떤 특징이 있는지 파악하는 것이 중요하다는 것을 발견했습니다. **WandB와 Pandas를 통해 데이터를 분석했고, 취약한 부분을 발견할 수 있었습니다. 해당 데이터들에 더 가산점을 줘서 데이터 증강을 진행했고 성능을 높일 수 있었습니다.**
- 논문 탐색 및 적용의 중요성
 - 주어진 문제에 관한 효과적인 해결책을 제시했던 논문을 찾고 구현해낼 수 있는 역량이 중요하고 필요하다는 것을 알게 되었습니다. 여러 논문들 중 특히 데이터의 Type을 input 문장에 함께 삽입해주는 방법론에 관한 논문이 인상 깊었고, 매우 효과적이었습니다. 해당 기법을 통해 점수를 획기적으로 상승시킬 수 있었습니다.