

Portfolio

Problem Solver

이예준

포트폴리오

기술 스택

- **Apache Airflow:** ETL 파이프라인의 자동화 및 워크플로우 스케줄링 관리에 활용
- **Google ADK(Agent Development Kit):** Stateful Multi-Agent 시스템 설계 및 에이전트 간의 도구 호출(Tool Calling), 에이전틱 워크플로우 구현
- **Vector DB(DuckDB vss):** 멀티모달 RAG 파이프라인 구축을 위한 벡터 데이터 인덱싱 및 유사도 검색 엔진 구현 역량
- **WandB (Weights & Biases):** 실험 지표의 시각적 분석을 통한 모델링 의사결정 수행
- **Pandas & Scikit-learn:** 텍스트/이미지 메타데이터 분석 및 Stratified K-Fold 등 통계적 검증 기반의 Feature Engineering 전략 수립
- **Huggingface Transformers:** Custom Tokenizer 구축을 통한 비정형 데이터의 OOV 문제 해결 및 Label Smoothing, LR Scheduler 등 모델링 역량
- **Albumentations:** 의료 영상의 기하학적 벡터 분석 및 도메인 특화 증강(SSR, Elastic Transform)을 통한 데이터 분포 불일치 해결 경험
- **Git & GitHub:** Branch 전략기반의 버전 관리 및 Pull Request를 통한 코드 리뷰 및 협업 환경 운영 경험
- **Notion & Slack:** 프로젝트 일정 관리 및 다양한 외부 툴(GitHub, Airflow, WandB 등) 연동을 통한 팀 커뮤니케이션 최적화

학습 기록

논문을 읽고 아이디어를 실제 프로젝트에 담아냈어요.

벡터 구성성분(Compositionality) 기반의 HyDE(Hypothetical Document Embeddings) 검색 최적화

[관련 논문] *Distributed Representations of Words and Phrases and their Compositionality* (Mikolov et al., 2013) [블로그](#)
[논문리뷰 링크](#), [embedding 시각화 실습 링크](#), [FAISS를 이용한 VectorSearch 실습 링크](#)

[핵심 성과] 단어 벡터 간의 선형적 산술 연산($V_{King} - V_{Man} + V_{Women} \approx V_{Queen}$)이 가능하다는 논문의 Compositionaliy(구성성분성) 이론을 '이미지기반 카페 추천 프로젝트'의 HyDE기반 피드백 루프로직으로 실체화했어요.

[엔지니어링 세부 사항]

- **벡터 선형성 응용:** 사용자의 모호한 '분위기' 쿼리를 단순 검색어에서 다차원 임베딩 벡터로 정의
- **피드백 루프 설계:** 생성된 분위기 벡터들의 임베딩 평균값을 피드백 루프를 통해 원본 쿼리 벡터와 결합(산술평균)
- **검색 공간의 심화:** 반복적인 루프를 거치며 쿼리 벡터가 분위기 벡터들의 평균 지점인 $\frac{1}{n} \sum_1^n V_{HyDE}$ 으로 수렴하게 함으로써, 추상적인 속성 벡터들이 조합된 '합벡터'가 가리키는 의미적 목적지를 정밀하게 보정
- **결과:** 논문에서 증명된 벡터 공간의 기하학적 특성을 활용해, 단순 키워드 매칭으로는 불가능했던 '공간의 감성적 맥락(Vibe)'을 Vector 평균으로 찾아냄
- [구체적인 agent에서의 HyDE심화 로직은 블로그에 정리해뒀어요](#)

학습한 것들을 항상 회고하고 기록하고 있어요.

- 부스트캠프에서 진행한 학습내용을 매주마다 회고하고 정리했어요. [학습회고 링크](#)
- [프로젝트 WrapUp Report 우수사례 선정](#)

협업 능력

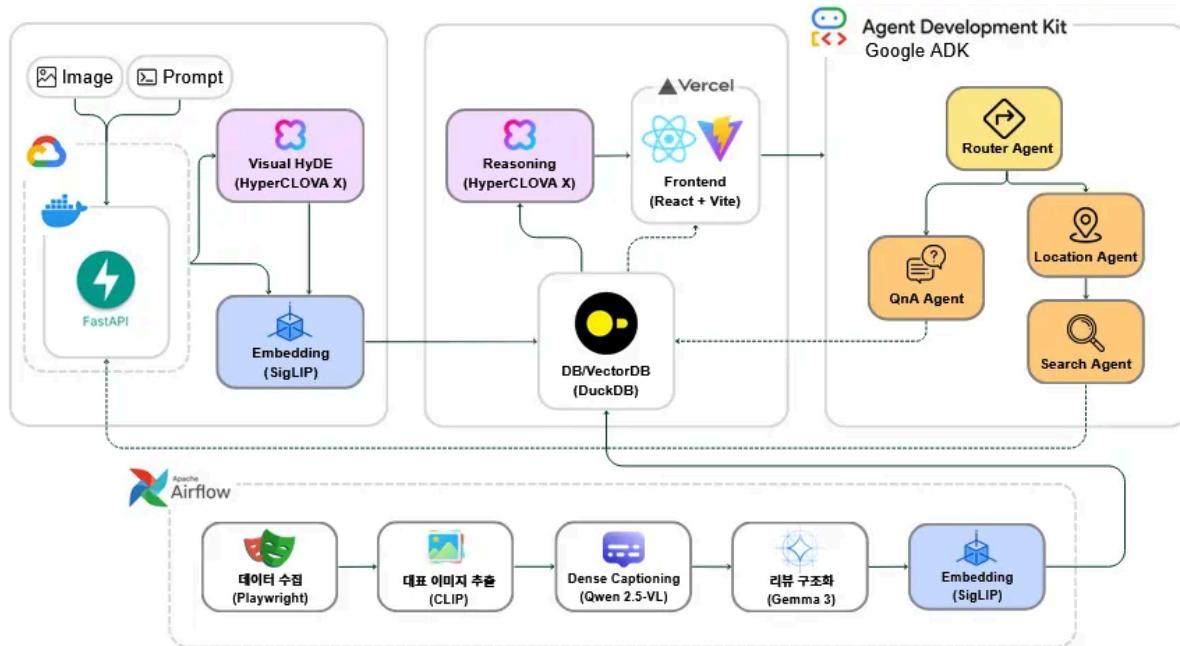
- 팀원과 함께 성장하려고 노력해요.
 - GitHub & Slack:** 모든 작업은 Issue 단위로, PR 할 때는 동료의 승인 후 머지하는 문화를 주도했어요. 그리고 Slack 알림 연동을 통해 팀 내 코드 리뷰 및 이슈 대응에 활용했어요
 - Notion:** 팀 회의록, 타임테이블, 실험진행내역, 인사이트 등을 팀원들과 공유했어요
 - 구체적인 팀 협업방법은 블로그에 정리해뒀어요

프로젝트

이미지기반 카페추천 프로젝트 (2026.1.29-2026.2.11) / 네이버 클라우드 해커톤 / 팀 프로젝트

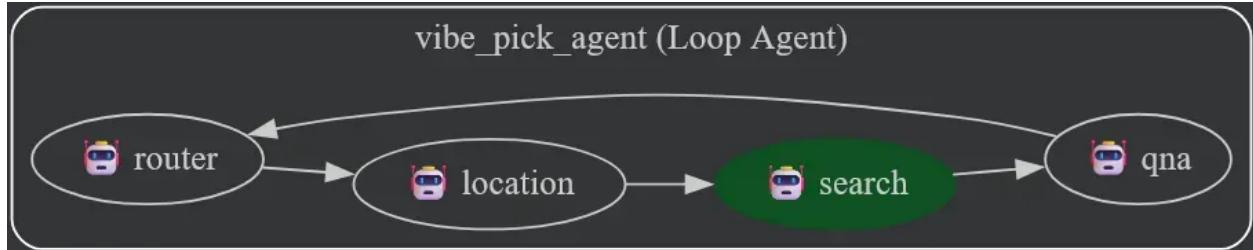
- 역할:** AI/Data Engineer & MLOps (데이터수집, 전처리, ETL파이프라인 구축 및 ADK기반 챗봇개발)
- 프로젝트 목적:** 사용자가 원하는 이미지 감성과 분위기를 반영할 수 있는 이미지-텍스트 결합 멀티모달 추천 서비스 구현

아키텍처



문제 정의 및 해결

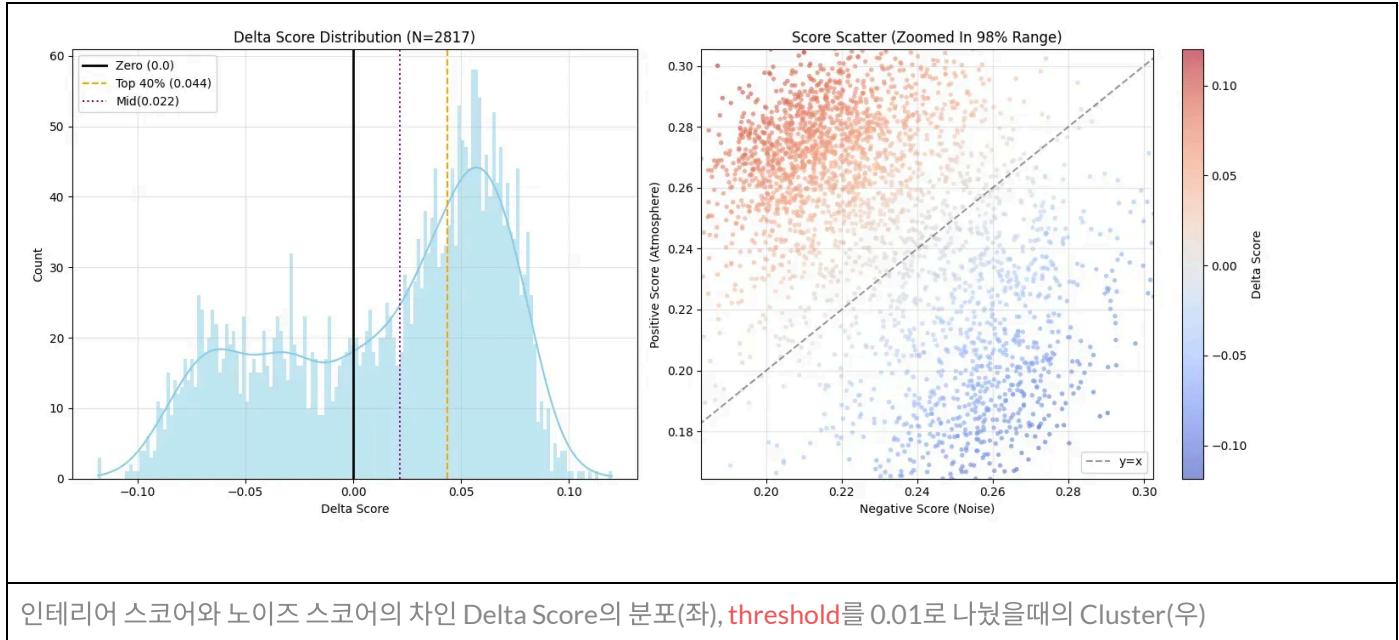
1. [AI Engineering] Google ADK 기반 Stateful Multi-Agent 챗봇 설계



- **문제**
 - 기존 1회성 검색 서비스는 "더 조용한 곳", "첫 번째 카페 주차 돼?"와 같이 매 턴 변화하는 유저의 의도(검색, 피드백, Q&A)를 파악하지 못하며, 이전 대화의 맥락이 유실되어 연속적인 탐색이 불가능함.
- **분석 및 인사이트**
 - 유저 쿼리를 단일 LLM으로 처리하는 방식에서 벗어나, 의도에 따라 agent에게 Routing하고 AgentState를 전역적으로 공유하며 반복 추론할 수 있는 워크플로우 기반의 Multi-Agent 아키텍처가 필요함.
- **해결 방안**
 - Google ADK의 LoopAgent 구조를 도입하여 Router, Location, Search, QnA 에이전트가 순차적으로 협업하는 시스템 구축
 - Router로 발화 의도를 분류하고, Pydantic과 DuckDB를 연동하여 세션 상태를 관리
 - FEEDBACK 상황에서 이전 HyDE 쿼리를 누적하여 "더 ~한 곳"과 같은 비교 검색 시 분위기 맥락을 유지하도록 설계
- **결과**
 - 자연어 내 랜드마크/역 이름 추출 및 행정동 변환(Location Agent)을 통해 검색 재현율 약 4배 향상
 - 이전 답변을 기억하는 상세 정보 질문(QnA) 및 분위기 누적 피드백이 가능한 Stateful 에이전트 서비스 구현

2. [Data Engineering] 모호한 감성 데이터의 정량적 정제

- **문제**
 - 크롤링된 이미지 중 메뉴판, 음식 사진 등 '장소 분위기'와 무관한 노이즈 데이터가 40% 이상 존재되어 검색 품질이 저하됨.
- **분석 및 인사이트**
 - '분위기'는 주관적이지만, CLIP 모델을 이용한 Prompt Engineering을 통해 정량적인 '분위기 점수'를 산출할 수 있음을 발견함.
 - Metric Score 분포가 Bimodal Distribution을 보임을 확인하여, 두 클러스터를 가르는 최적의 Threshold 설정이 핵심임을 파악함.
- **해결 방안**
 - CLIP 프롬프팅 메트릭을 정의하고 Threshold 0.01을 적용하여 노이즈 이미지 자동 필터링 파이프라인 구축.
- **결과**
 - 노이즈 데이터 40% 이상 제거 및 검색 결과의 시각적 일관성 확보.



3. [ETL Pipeline] 대규모 데이터 처리 병목 해소

- 문제
 - 420여 개의 법정동 단위 데이터를 순차적으로 처리하면서 크롤링 실행 시간이 72시간에 달해 GPU유휴시간 발생.
- 분석 및 인사이트
 - 각 지역 단위 태스크는 독립적이므로, 고정된 워크플로우 대신 데이터 유무에 따라 동적으로 확장되는 구조가 필요함
- 해결 방안
 - Apache Airflow의 **Dynamic Task Mapping**을 도입하여 태스크를 동적으로 병렬화하고, **FileSensor** 기반 ETL 파이프라인으로 수집-가공 간의 **병목 해결**.
- 결과
 - 전체 실행 시간 약 50% 단축 (144시간 → 72시간)

4. [AI Engineering] 모델 제약(64 Tokens)을 극복한 Sliding Window 기반 텍스트 임베딩 고도화

- 문제
 - 임베딩 모델로 선정한 **SigLip**이 64토큰까지만 처리할 수 있도록 사전 학습되어, 문장 형태의 HyDE 쿼리나 긴 캡션 데이터가 **중간에 잘리는 현상(Truncation)** 발생
- 분석 및 인사이트
 - 모델 자체를 교체하는 것은 개발 비용 면에서 비효율적임
 - 한국어 특성상 50자 내외의 짧은 글도 130토큰 이상을 생성하므로 단순한 글자 수 제한으로는 해결이 불가능
 - **프롬프트 고도화**를 통한 정보 밀도 압축과 **슬라이딩 윈도우(Sliding Window)** 기법을 통한 전체 맥락 유지가 병행되어야 함을 도출
- 해결 방안
 - **Prompt Compression**: HyDE 생성 시 문장 형식이 아닌, 콤마(,)로 구분된 **명사구 나열식**으로 출력하도록 프롬프트를 수정하여 정보 밀도를 극대화함.
 - **Sliding Window Embedding**: 64토큰 윈도우와 32토큰 Stride를 적용하여 텍스트를 중첩 분할함
 - 분할된 각 Chunk의 임베딩을 구한 후 **Mean Pooling**을 통해 하나의 최종 벡터로 결합하는 로직 구현함

- **결과**

- 모델의 토큰 제한을 넘어서서 텍스트의 전체 맥락을 벡터 공간에 보존하는 데 성공함. 프롬프트 최적화와 윈도우 전략의 조합으로 검색 엔진의 의미론적 정확도를 유지함

5. [Data Engineering] Gemma3 기반 비정형 리뷰 데이터의 정제 및 검색용 구조화

- **문제**

- 크롤링된 Raw 리뷰 데이터는 이모지, UI 텍스트(예: "더보기"), 비한국어 리뷰 등 노이즈가 많아 검색 및 추천 시스템의 재현율을 저하시킴
- 일관되지 않은 리뷰 형식으로 인해 검색에 필요한 핵심 태그(#)와 Retrieval용 요약 정보를 추출하기 어려운 상태임

- **분석 및 인사이트**

모델	실행 시간 (리뷰 50개)	요약 품질	비고
Gemma3-4B-IT	약 10초 (GPU)	완벽함	최종 선정 (4bit 양자화 활용)
Gemma3-1B-IT	약 7초 (GPU)	문맥이 약간 어색함	속도는 빠르나 표현력이 부족하여 제외함
Gemma3-4B-GGUF	약 10분 (CPU)	좋음	4코어 CPU/8GB RAM에서도 동작하나, 대량 처리에 부적합함

- 고품질 메타데이터 추출을 위해 1B 모델보다 문맥 파악력이 우수한 Gemma3-4B-IT(4bit 양자화)를 최종 선정
- 단순 필터링만으로는 데이터 유실이 크므로, 정규식 기반의 전처리를 통해 입력 토큰을 최적화하고 모델이 핵심 정보에 집중할 수 있는 환경을 조성해야 함을 파악
- Single Inference시 V100 GPU(32GB) 리소스가 유휴 상태로 방치되어 자원 낭비가 발생하는 것을 확인, Batch Inference를 통한 연산 효율화가 필수적임을 파악

- **해결 방안**

- **데이터 정제:** 정규표현식을 사용하여 노이즈를 제거하고, 글자 수(5~200자) 및 한국어 비율(50% 이상) 필터를 적용하여 유의미한 상위 30~50개의 고품질 리뷰만 LLM의 입력값으로 사용함
- **Structured Prompting:** 가게 소개를 위한 '태그 생성(Task 1)'과 가게 'QnA'시 Retrieval을 위한 '가게요약(Task 2)'으로 프롬프트를 분리 설계하여 목적에 맞는 구조적 정보를 추출
- **Batch Inference 구현:** 배치 사이즈 8 단위로 병렬 추론을 수행하여 VRAM 활용도를 극대화

- **결과**

- 비정형 리뷰를 **summary_for_tag** 및 **summary_for_display** 필드를 가진 정형 JSON 데이터로 변환
- **summary_for_display**를 Retrieval해서 유저에게 QnA 및 Context 기반의 다음 추천 질문을 제공하는 인터랙티브 환경을 구축
- 단건 처리 대비 처리 속도 약 3~4배 향상

결과

- 이미지 기반 카페 추천 데이터 구축 시 메뉴판, 음식 등 분위기와 무관한 노이즈 사진이 40% 이상 존재되어 검색 품질이 저하되었으나, CLIP 기반 분위기 메트릭 정의 및 Bimodal 분포 기반 Threshold(0.01) 필터링을 적용하여 노이즈 40% 제거 및 시각적 일관성 확보를 달성
- 420여 개 법정동 카페 데이터 수집 및 전처리 시 순차적 처리로 인한 144시간의 긴 실행 시간 및 데이터 병목이 발생했으나, Apache Airflow의 Dynamic Task Mapping 및 FileSensor 기반 ETL 파이프라인을 설계하여 전체 실행 시간 50% 단축(144h → 72h)
- 자연어 기반 카페 검색 및 멀티턴 대화 서비스 구현 시 사용자의 모호한 검색 의도 분류 불가 및 이전 대화 맥락 유실 문제가 있었으나, Google ADK 기반 Multi-Agent 아키텍처 및 DuckDB 상태 관리를 도입하여 검색 재현율 약 4배 향상 및 연속적인 탐색 구현
- 긴 HyDE 쿼리 및 캡션 데이터 임베딩 시 SigLip 모델의 64토큰 처리 제한으로 인한 정보 누락 및 검색 정확도 저하가 우려되었으나, 명사구 중심 프롬프트 압축 및 Sliding Window(Stride 32) 기법을 적용하여 긴 문장의 전체 맥락 벡터 보존 및 의미론적 검색 성능 유지
- 비정형 리뷰 데이터의 구조화 및 메타데이터 추출 시 노이즈 텍스트로 인한 토큰 낭비 및 단건 처리의 낮은 GPU 활용도가 병목이었으나, 정규식 기반 토큰 최적화 및 Gemma3-4B-IT(4bit) Batch Inference를 수행하여 입력 토큰 30% 절감 및 처리 속도 약 3.5배 향상

손가락 뼈 Semantic Segmentation (2025.12.15~2026.1.7) / 네이버 커넥트재단 / 팀 프로젝트

- 역할:** AI Engineer (모델링 및 전략 수립)
- 목표 및 배경:** 29개 클래스의 손 뼈 X-ray 이미지를 픽셀 단위로 정밀하게 Segmentation하여 의료 진단을 보조할 수 있는 Semantic Segmentation 모델 개발.



문제정의 및 해결

1. [AI Engineering] 고해상도 정보 보존을 위한 HRNet 기반 Segmentation 최적화

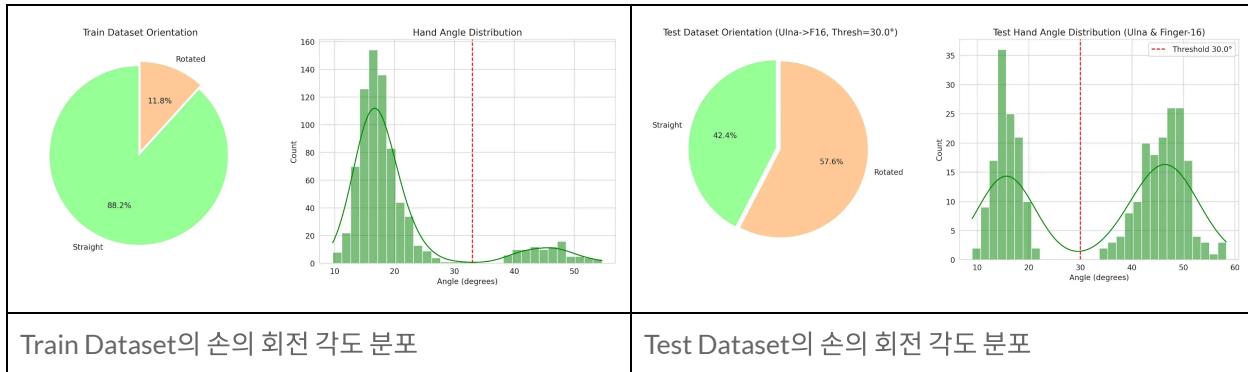
- **문제**
 - 2048px 원본 이미지를 512px로 **다운샘플링**하는 과정에서 **미세한 뼈 경계 및 작은 객체**(손가락 끝 뼈 등)의 정보 소실이 발생하여 성능 병목이 확인됨
- **인사이트**
 - 모델 파라미터 수를 늘리는 것보다 **입력 해상도를 최대한 보존**하는 것이 경계선 디테일 확보에 더 결정적임을 파악
 - 제한된 VRAM 환경에서 해상도를 키우기 위해 배치 사이즈를 줄여도 학습이 안정적인 구조가 필요함을 도출함
- **해결 방안**
 - 고해상도 특징을 끝까지 유지하는 **HRNet**을 도입
 - AMP(Automatic Mixed Precision)를 적용하여 **VRAM 효율을 극대**
 - **Batch Size 1**에서도 안정적인 학습을 위해 **Group Norm**을 구현하여 입력 해상도를 원본 수준(최대 1536px)으로 유지하는 전략 수립
- **결과:** 베이스라인(Dice 0.94) 대비 약 3.5%p 성능 향상



뼈 경계부분에서 False Negative와 False Positive 발생됨을 시각화

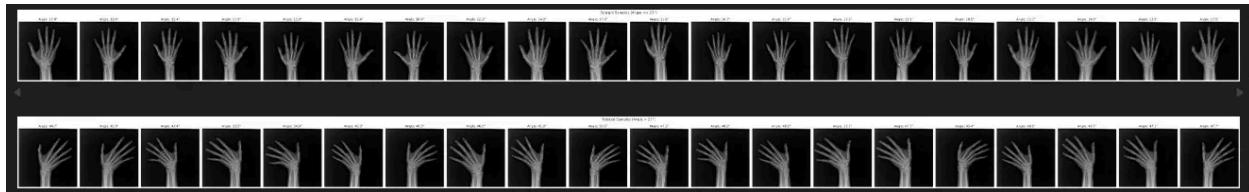
Encoder	Decoder	Resolution	Dice Score	비고
DeepLabV3+	ResNet50	512px	0.94	Baseline
HRNet W48		2048px	0.9648	배치사이즈 1, Batch Norm 사용시 해상도를 올려도 오히려 성능하락
HRNet W48		1536px	0.9755	배치사이즈 1, Group Norm 사용시 최고 Dice score 달성

2. [Data Engineering] 벡터 분석 기반 기하학적 분포 불일치 해결



- 문제

- EDA 결과, Test Set에 포함된 '회전된 손'의 비율이 Train Set 대비 약 10배 높음을 발견하여 **기하학적 분포 불일치 확인**



회전되지 않은 손(위), 회전된 손(아래)

- 인사이트

- 새끼손가락 끝과 손목뼈 중심을 잇는 **벡터 분석**을 통해 손의 회전 각도의 분포를 확인하고 **33도**를 threshold로 정해서, 회전여부를 파악
- 모델이 특정 각도에 편향되지 않도록 회전에 Robust한 증강 전략이 필수적임을 파악

- 해결 방안

- 벡터 분석 결과를 바탕으로 **ShiftScaleRotate(SSR)** 증강 전략을 수립하여 기하학적 분포 차이를 인위적으로 보정

- 결과

- 분포 불균형 문제를 해결하고, **Public Score 0.2%p 향상**을 달성

3. [AI Engineering] PointRend/EMA 기반 정밀도 고도화 및 CosineRestart를 통한 일반화 성능 개선

- 문제

- 의료 영상 특성상 미세한 뼈 경계면이 뭉개지는 현상과 VRAM 제약으로 인한 소규모 배치(BS=1) 환경에서의 불안정성 문제가 발생함

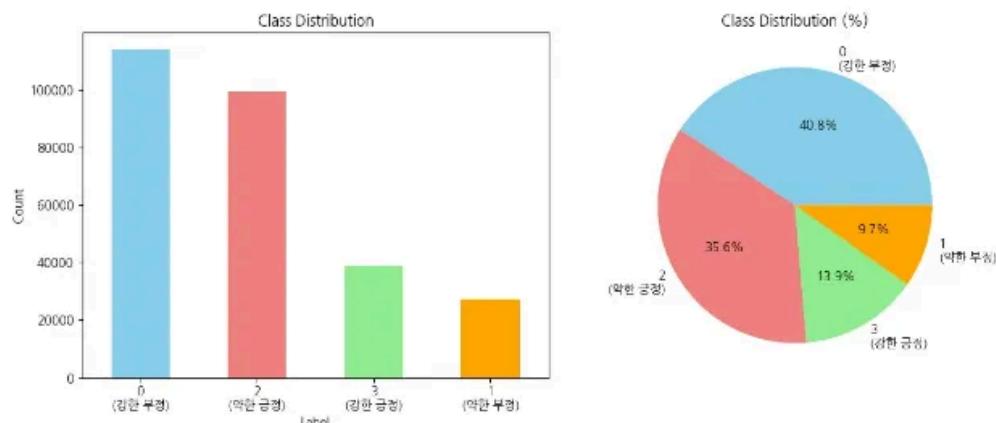
- Public 및 Validation Score 간의 점수 차이가 모델마다 불규칙하게 나타나며, Local Minima 고착으로 인한 일반화 성능 저하가 확인됨
- 인사이트
 - 단순 Upsampling의 한계를 극복하기 위해 PointRend를 통해 불확실 영역만 적응적으로 재검색(Point-based sampling)하고, EMA(Exponential Moving Average)을 통해 수렴의 시간적 일관성을 확보해야 함
 - 주기적으로 학습률을 재시작하는 CosineRestart가 Local Minima를 탈출하여 Flat Minima를 찾는데 기여할 것으로 판단함
- 해결 방안
 - 경계 정밀도 향상을 위한 PointRend 모듈과 소규모 배치에서의 가중치 수렴 안정화를 위한 EMA를 도입함
 - CosineRestartWarmupScheduler를 구현 및 적용하여 수렴 후반부의 탐색 공간을 확장하는 Ablation Study를 수행함
- 결과
 - 정량적 성능 향상 (PointRend & EMA): 적용 전(0.9697) 대비 Public Score 0.31%p(0.0031) 상승을 통해 미세 골격 경계면 예측 품질을 개선함.
 - 일반화 성능 확보 (LR Scheduler): Public 및 Validation Score 간의 점수 차이를 0.0028 수준으로 하락시키며 과적합을 억제하고 안정적인 추론 성능을 입증함.

결과

- 고해상도 의료 영상 Segmentation 시 512px 다운샘플링 과정에서 발생하는 미세 뼈 경계 및 소형 객체의 정보 소실 문제를 HRNet 도입 및 Group Norm 기반의 1536px 고해상도 유지 전략으로 해결하여 베이스라인 대비 약 3.5%p 성능 향상(Dice 0.9755)을 달성
- 데이터 분포 불일치 해결 과정에서 Train 대비 Test Set의 회전된 손 비율이 10배 높은 기하학적 분포 불균형을 규명하고, 벡터 분석 기반의 ShiftScaleRotate(SSR) 증강 전략을 수립하여 Public Score 0.2%p 향상 및 모델 일반화 성능 개선
- 소규모 배치 학습 시 경계면 둥개짐 및 일반화 성능 저하 문제를 PointRend/EMA 도입 및 CosineRestart 스케줄러 적용으로 해결하여 Public Score 0.31%p 향상 및 점수 차 0.0028 수준 하락을 달성하며 정밀도와 일반화 성능을 확보

영화 리뷰 감성 분류 및 예측 (2025.10.20~2025.10.30) / 네이버 커넥트재단 / 개인 프로젝트

- 목표 및 배경: 영화 리뷰 텍스트의 문맥을 파악하여 4가지 감정 클래스(강한/약한 긍정 및 부정)로 분류하는 BERT 기반의 다중 클래스 분류 모델 개발.



--- 수집된 라벨별 상위 감성 패턴 --

[Label 0]: [('ㅠㅠ', 33867), ('ㅋㅋ', 7427), ('—', 2363), ('ㅎㅎ', 954), ('ㅜㅜ', 519), ('Ѡ', 44), ('★', 33), ('ㅋㅋㅠㅠ', 30), ('—ㅋ', 26), ('ㅜㅠ', 24)]

[Label 1]: [('ㅠㅠ', 9489), ('ㅎㅎ', 2362), ('ㅋㅋ', 1856), ('ㅜㅜ', 86), ('—', 78), ('★', 26), ('Ѡ', 22), ('*', 18), ('ㅠ', 12), ('♥', 11)]

[Label 2]: [('ㅎㅎ', 19727), ('ㅋㅋ', 9407), ('ㅠㅠ', 5677), ('♥', 832), ('*', 640), ('♦', 510), ('ㅜㅜ', 477), ('Ѡ', 415), ('♡', 360), ('👍', 351)]

[Label 3]: [('ㅎㅎ', 5713), ('ㅠㅠ', 4948), ('ㅋㅋ', 3900), ('♥', 2067), ('♡', 571), ('ㅜㅜ', 391), ('*', 330), ('★', 206), ('♦', 142), ('👍', 137)]

문제정의 및 해결

1. [Data Engineering] 한국어 비정형 표현 처리를 위한 Custom Tokenizer 구축

- 문제
 - 영화 리뷰의 핵심 감성 지표인 자음/모음 반복(ㅋㅋ, ㅠㅠ) 및 이모지가 기존 Tokenizer에서 OOV(Out-Of-Vocabulary)로 처리되어 문맥적 감성 정보가 소실되는 현상 발생
- 인사이트
 - 한국어 감성 분류 Task에서 비정형 감정 표현은 정보 밀도가 높으므로, 이를 [LAUGH], [SAD] 등의 특수 토큰으로 명시적으로 매핑하여 학습시키는 것이 모델의 문맥 파악에 결정적임을 도출함
- 해결 방안
 - 'ㅋㅋ', 'ㅠㅠ', 이모지 등을 특수 토큰으로 매핑하는 Custom Tokenizer를 구축하고, 감정 표현을 명시적으로 토큰화하여 임베딩 레이어가 핵심 감성 정보를 보존하도록 전처리 파이프라인을 설계함.
- 결과
 - OOV 문제를 근본적으로 해결하고 텍스트의 감성 표현력을 극대화하여 베이스라인 대비 Public Score 1.8%p 성능 향상을 달성함

2. [AI Engineering] 일반화 성능 확보를 위한 SWA 및 Weighted Ensemble 전략

- 문제
 - 단일 모델 학습 시 Local Minima에 빠지거나 과적합되어, Validation과 Private Score 간의 높은 변동성으로 인한 안정적인 일반화 성능 확보의 어려움이 확인됨
- 인사이트
 - 서로 다른 시드(Seed)로 학습된 모델들의 예측값을 결합하고, 가중치 평균(SWA)을 통해 Loss Landscape를 평탄화하여 모델의 강건성을 확보해야 함을 파악함
- 해결 방안
 - TAPT(Task-Adaptive Pre-Training)를 수행한 3개의 BERT 기반 모델에 Weighted Soft-voting과 SWA(Stochastic Weight Averaging)를 결합 적용하여 양상을 함
- 결과
 - 모델의 과적합을 방지하고 일반화 성능을 강화하여 Public Score 0.8%p의 성능 향상

결과

- 영화 리뷰 감성 분류 모델 개발 시 비정형 표현의 OOV 처리로 인한 정보 소실 및 단일 모델의 과적합/일반화 부족 문제를 Custom Tokenizer 구축 및 SWA 기반 Weighted Ensemble 적용으로 해결하여 Public Score 1.8%p 및 0.8%p의 단계적 성능 향상