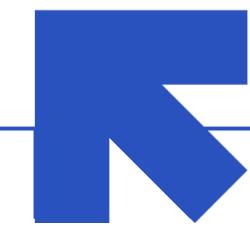


무한혁신상사



세상을 놀라게 할 온라인 리서치 9기 <C2 민혁 없는 민혁 팀>입니다.



목차



1. 프로젝트 워크플로우 개요

프로젝트 컨셉 소개, 프로젝트 워크플로우

2. 실험단계 1 : 모델 선별 과정

모델 리스트 : 특성과 장단점, 통제조건과 평가항목, 선별된 모델 소개

3. 실험단계 2 : 모델 개선 과정

성능개선용 훈련 데이터 특성, 모델 개선 과정, Further Implementation

4. 응용 : 상품화 & 타겟 마켓 선정, MVP 구상

모델 상품화 & 타겟 시장 및 고객 선정, MVP 제작 시도

프로젝트 수주

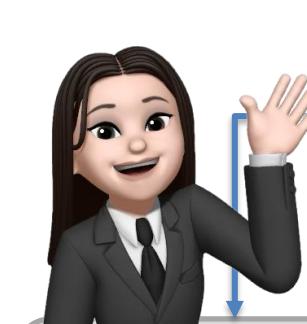
항목	내용
프로젝트명	텍스트 분류 기반 범죄 예측 및 대응 시스템 개발
클라이언트	서연일렉트로닉스 스마트폰 SW사업부
프로젝트 목표	텍스트 분석을 통해 협박, 갈취, 괴롭힘 등의 범죄 대화를 분류하고, 범죄 예방 및 대응 솔루션 제공
핵심 기술	사전훈련 모델 기반 텍스트 분류 모델, 합성 데이터 사용한 일반 대화 데이터 추가, Augmentation 기법 활용
응용 분야	<ol style="list-style-type: none"> 범죄 예측 및 대응 챗봇 (어린이 보호) 폭언 감지 시스템 (콜센터, 고객 대응) 범죄 예방 교육 프로그램 악플 방지 시스템
프로젝트 범위	<ul style="list-style-type: none"> - 협박, 갈취, 괴롭힘, 일반 대화 5가지 클래스 분류 - Pre-trained 모델(KoBART) 사용 - 일반 대화 합성 데이터 생성 및 학습
성과 목표	<ul style="list-style-type: none"> - 범죄 대화 감지 정확도 85% 이상 - 일반 대화 및 범죄 대화 분류 모델 성능 평가 및 최적화
예상 소요 기간	3개월
기대 효과	범죄 대화 사전 감지 및 대응을 통해 사회 안전 강화 교육 프로그램 및 서비스 품질 향상

인공지능개발혁신부서

부서 조직도 (역할)



김민혁 팀장
(총괄)



담안용 책임
(PO)



이준범 책임
(엔지니어)



정서연 책임
(데이터)

프로젝트 진행관리
발표 로직/모델링

데이터 증강
모델 조사

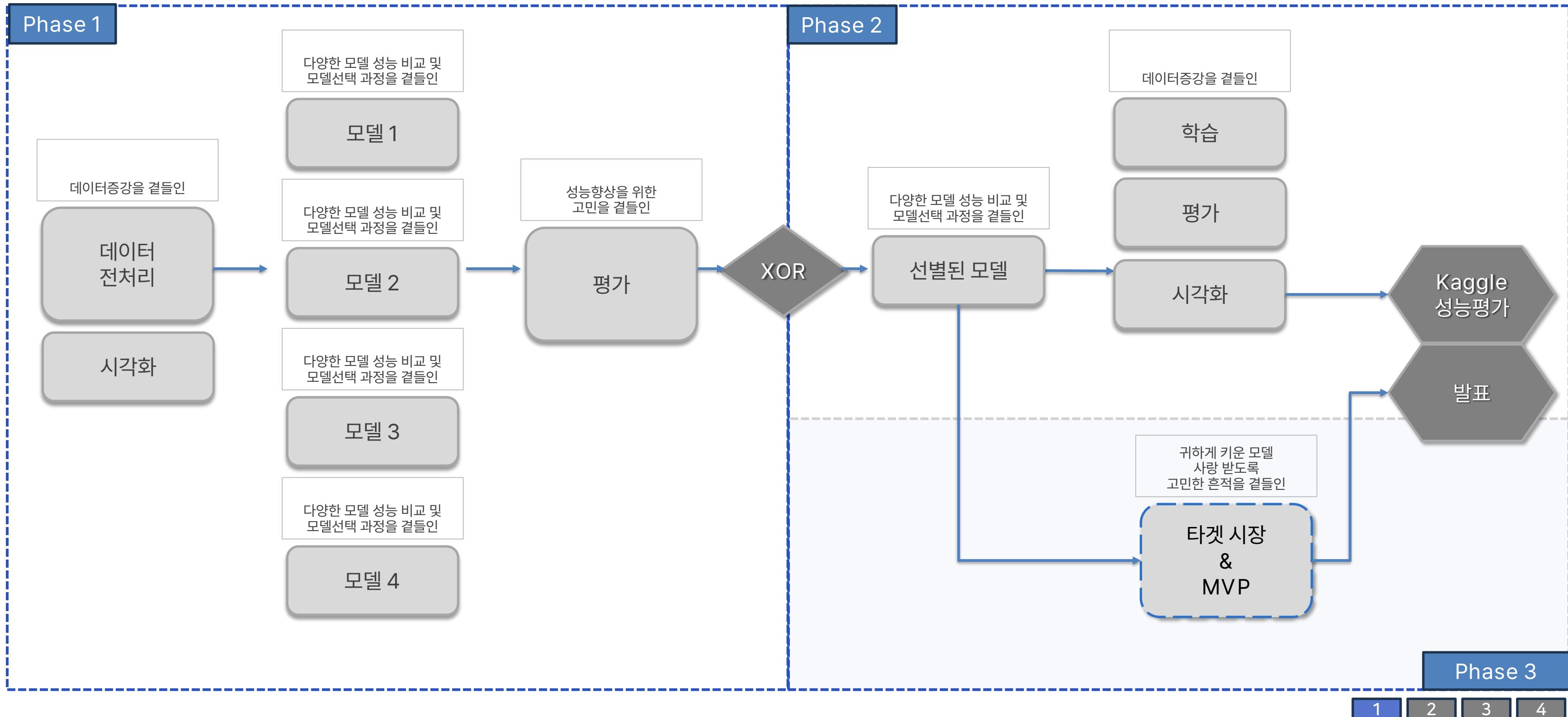
데이터 전처리
모델링

계획 및 관리
모델링

1-1. 프로젝트 워크플로우

프로젝트 개요

프로젝트는, 모델선별과정(phase 1), 모델개선과정(phase 2), 타겟시장 & MVP(phase 3) 총 3 단계로 구성

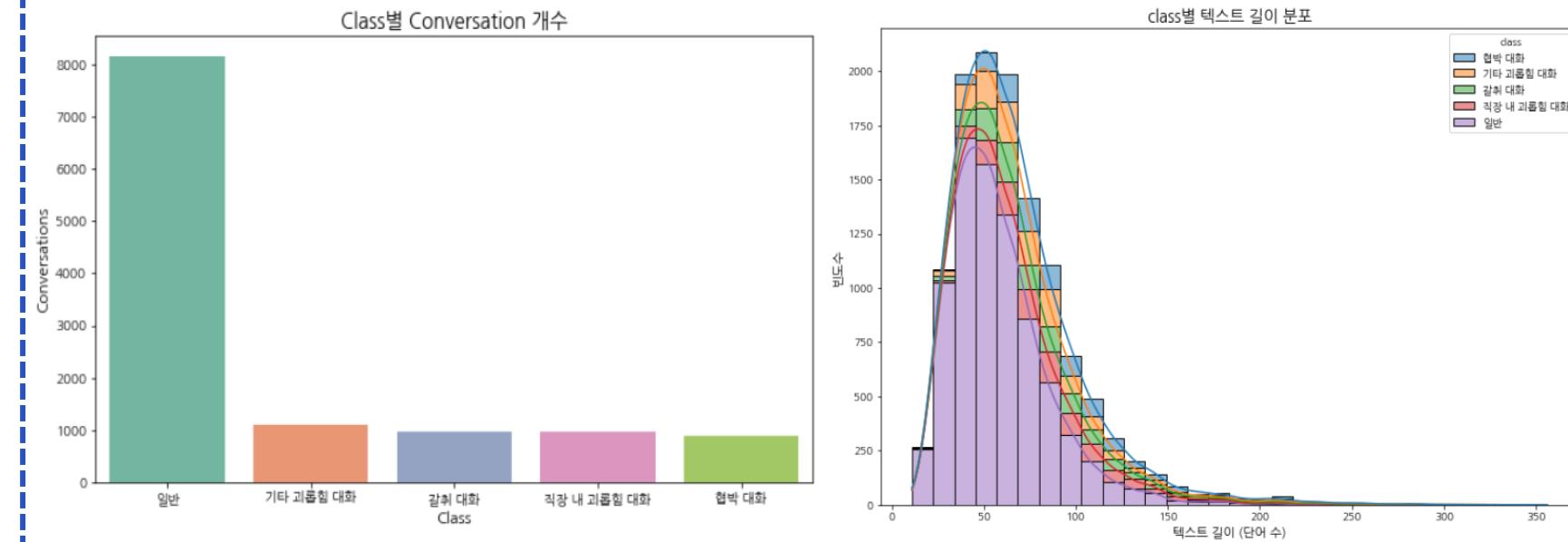


2-1. 성능개선용 훈련 데이터 특성

모델 선별 과정

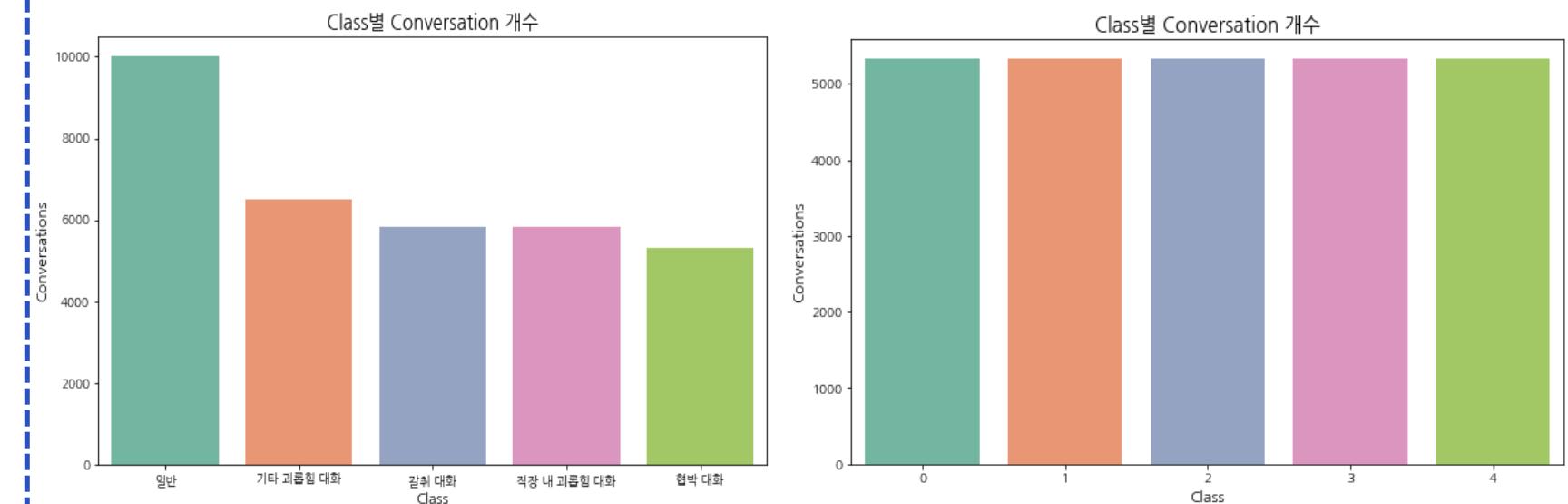
New_train

- 모델 선정을 위한 데이터 처리 과정



Aug_train

- 모델의 성능 개선을 위한 데이터 처리 과정



- 동일 Class data가 연속적으로 존재하지 않게 배치
→ Random Sampling 수행
- 데이터 불균형 해소 및 모델 성능 일반화를 위해 충분한 데이터 필요
→ Data Agumentation 및 down sampling 수행
- 사전훈련된 토크나이저 별 전처리 수행 확인 필요
→ the-tokenizer-playground 확인

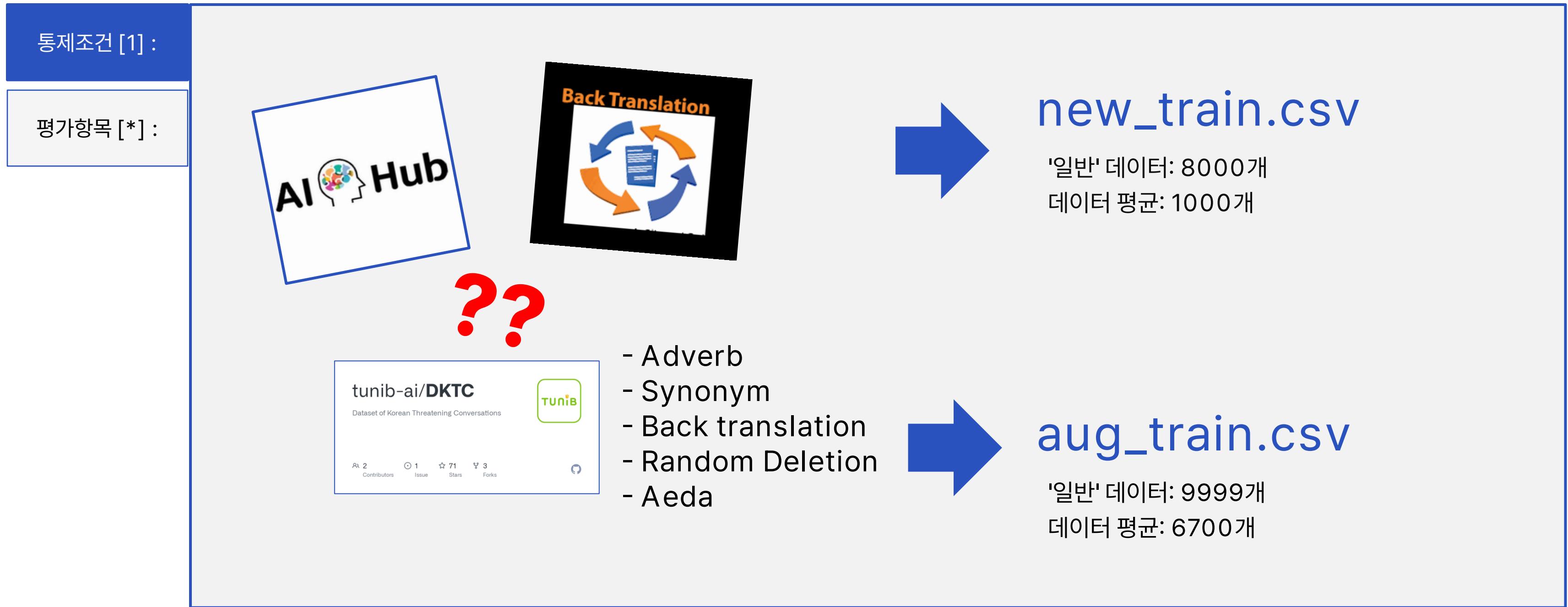
증강 방식	특징	장점	단점
Adberb	문장에 부사 추가	문장 의미 변화 적음	적절한 부사 적정 빈도 내에서 사용 필요
Synonym Replacement	원문 단어 동의어 대체	구조 변형 없이 쉽게 적용	동의어 사용 제한적
Back translation	외국어로 번역 후 한국어로 재번역	자연스러운 표현, 문장 의미 다양화	번역 품질 중요
Random deletion	단어 무작위 삭제	방식 간단, 불필요한 단어 삭제	과도한 삭제 시 데이터 품질 감소
Aeda	적대적 예시 사용	모델 약점 강화	많은 자원 필요, 잘못된 문제 생성 시 모델 학습력 감소

2-2. 전처리 & Leader board 점수

모델 선별 과정

위 모델들을 동일한 통제조건하 출력한 성능을 비교 평가하여 최선의 모델 1개를 선정. 이후, 해당 모델의 성능을 최대로 올려 보기로 함

통제조건은 모든 모델이 모두 “데이터증강 진행한 성능평가용 데이터셋”을 사용하는 것.



2-3. 전처리 & Leader board 점수

모델 선별 과정

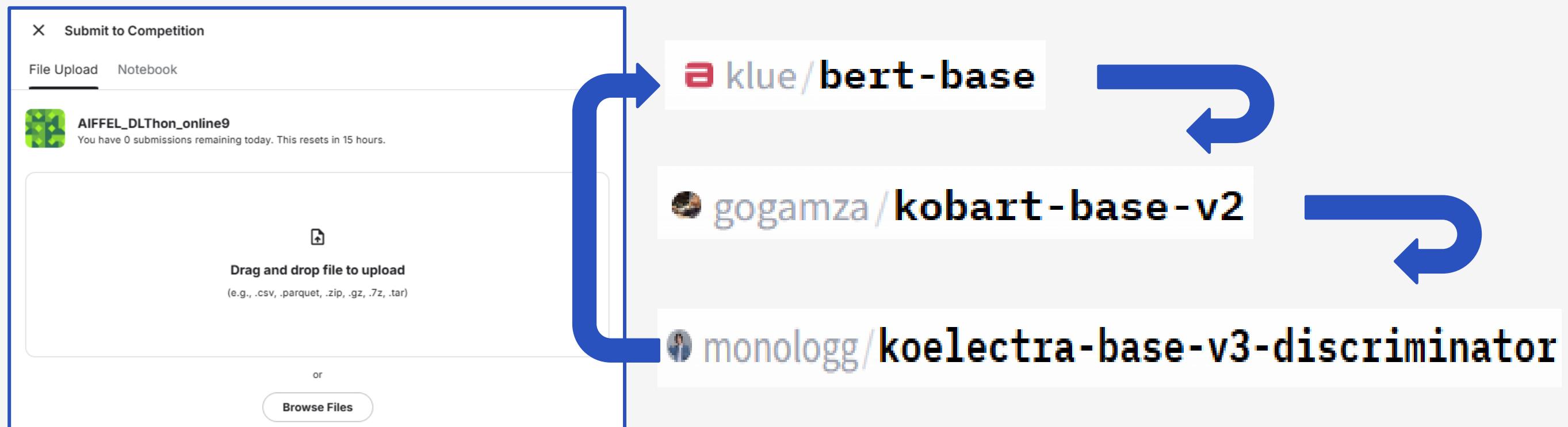
위 모델들을 동일한 통제조건하 출력한 성능을 비교 평가하여 최선의 모델 1개를 선정. 이후, 해당 모델의 성능을 최대로 올려 보기로 함

평가항목은 Kaggle 리더보드 점수로, 가장 높은 점수를 받은 모델을 Phase 2에서 성능 개선 시키기로 함.

통제조건 [1] :

평가항목 [2] :

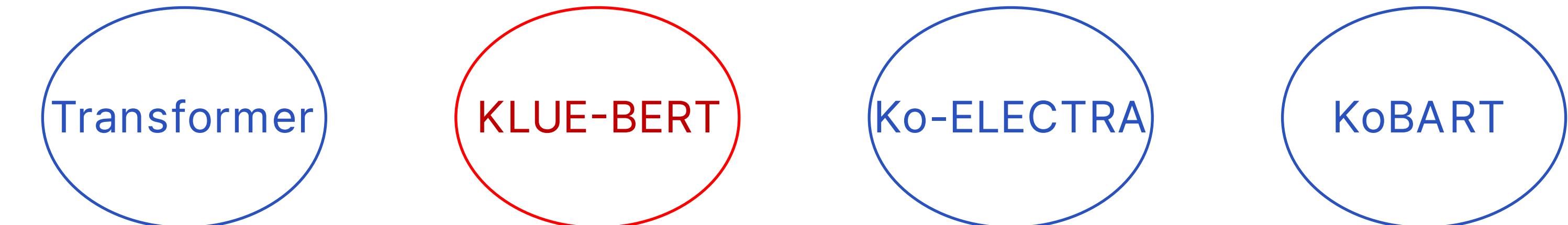
모델이 분류한 결과와 정답 간의 f1 score로 측정
측정 후 가장 성능이 좋은 모델 선택



2-4. 선별된 모델 특성 (한눈에 보기)

모델 선별과정

모델 별 특징 비교 및 KLUE-BERT 모델 최종 선정 이유



Seq2Seq 비사전훈련 모델

한국어NLP성능 벤치마크모델

Real-Fake 진위 판별 학습

Text filling 훈련 모델

장점 : 유연성, 크기&복잡도 조절

장점 : 감정분석, 분류, 유사도, 추론

장점 : 한국어 특화, 분류

장점 : 문맥 이해도, 생성

단점 : 사전지식 부족, 훈련비용 ↑

단점 : 연산시간 too long

단점 : 데이터 의존성 ↑, 단방향학습

단점 : 분류, 모델 크기 ↑

점수 : 0.15364

점수 : 0.24982

점수 : 0.17298

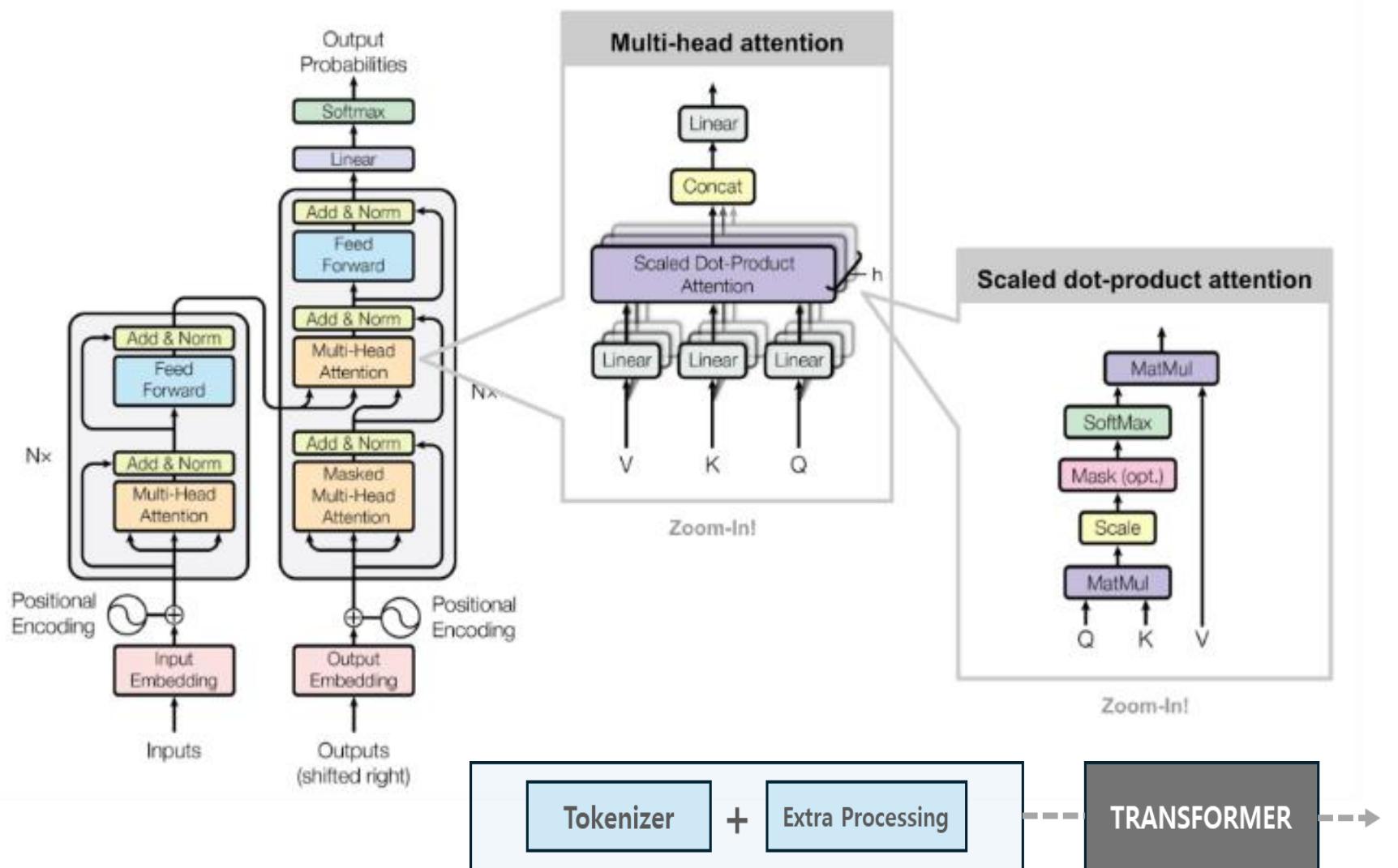
점수 : 0.13105

이론 학습 토대로 Transformer 모델 구현 → 데이터(사전 지식) 부족 → 사전훈련된 모델 활용 필요

1. KoBART의 경우, 문맥 이해도는 높지만 생성 작업에 특화되어 있어 분류 작업 성능 부족
2. Ko-ELECTRA의 경우, 분류에 특화되어 있지만 사전훈련된 데이터 도메인의 영향이 크고 높은 자원을 필요로 함.
3. KLUE-BERT의 경우, 모델 성능 평가를 위한 벤치마크 모델로 한국어 텍스트 분류에 뛰어난 성능을 보임.

최종적으로, 가장 높은 F1-score를 달성한 KLUE-BERT 모델을 선정하여 성능을 개선하고자 함.

1 트랜스포머 Seq2Seq 모델 (Non Pre-trained)



모델구조/토크나이저

토큰화 실행 함수와 트랜스포머류 단어 사전 정의 함수를 통한 생성

단어 수준 토크나이저: 고유한 단어 리스트 만들기

- 시작 토큰과 종료 토큰에 고유한 정수를 부여

성능

- 1) Positional 인코딩
- 2) Scaled Dot Product 어텐션
- 3) Multi-Head 어텐션
- 4) 마스크 패딩
- 5) 인코더, 디코더 구조

- Loss: 0.0236
- Accuracy: 0.9913
- F1 Score: 0.8538
- Score: 0.15364

장점: 생성 작업에 매우 적합, Task에 맞는 학습 가능

단점: 학습 비용이 큼, 사전 학습된 정보가 없음, 분류 작업 X

분류 작업 부적합!

→ Epoch을 50까지 줘봤을 때, Loss값은 잘 떨어지고 정확도 값은 계속 올라 99% 까지 되었다.

이후 f1 score 값이 85% 나왔지만 최종 리더 보드 점수는 굉장히 낮게 나온 걸로 보아,

시퀀스 모델이기에 아키텍처 구조 상 분류 작업에 특화되어 있지 않아서라는 것을 결론

2 KLUE/BERT (Pre-trained)



특성 :

- 사전학습모델,
- 한국어 자연어처리 성능 평가
벤치마크 모델

강점 :

- 감정 분석, 문서 분류,
- 유사도, 자연어 추론,
- 개체 명 인식

단점 : 거대모델이라 연산시간 LONG

모델 구조

트랜스포머 구조

- **입력:**
텍스트(토큰화된 단어) + 위치 임베딩 + 세그먼트 임베딩
- **인코더 레이어:**
12개의 트랜스포머 인코더 레이어
각 레이어는 다중 헤드 셀프 어텐션 + 피드포워드 신경망으로 구성
- **출력:**
[CLS] 토큰의 벡터(문장 전체 의미) + 각 단어에 대한 벡터(개별 단어 의미)

토크나이저

WordPiece는 서브워드 단위로 텍스트를 분할하려 처리

- - 띄어쓰기, 조사 등 알아서 처리
(ex."자연어처리" 자연어, ##처리로 나누어 학습 → 희귀단어도 서브워드 단위로 학습 → 일반화GOOD)
- 문맥 기반
- 서브워드 단위로 처리할 때 연산 비용 증가

오류 과정

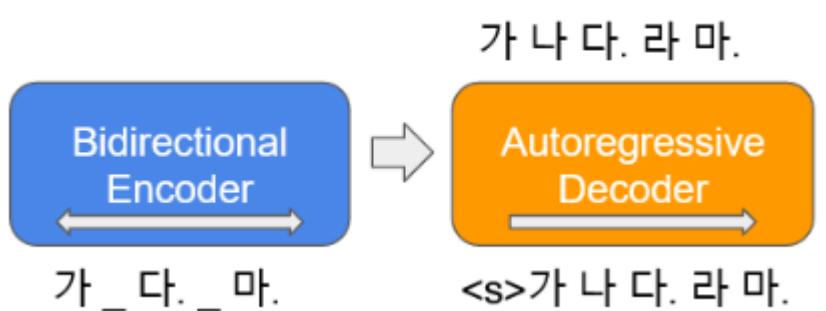
```
실행 중(49분 12초) 실행 중(38분 31초) 실행 중(16분 46초) <-- 실행 중(11분 58초) ···
# GPU 설정 (가능할 경우)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

성능

- [3 epochs]
- Loss: 0.06
 - Accuracy: 0.98
 - F1-score: -
 - Leaderboard f1-score: 0.24

→ 분류모델이자 벤치마크인 모델답게 분류에 탁월한 성과를 보임(klue0.24, trans0.15, kobart0.13),
띄어쓰기 등 한국어 데이터 특화 전처리 + 서브단위 텍스트분할 + 문맥, 유사도, 추론 특화 분류 모델이라는 점

3 KoBART (Pre-trained)



특성 : Text-filling 훈련 방식

장점

- 양방향 + 자기 회귀적 구조
- 대규모 한국어 데이터 학습
- 문맥 이해도 높음

단점

- 모델 크기가 큼
- 복잡한 구조로 인해 간단한 작업에는 비효율적

모델 구조

Model	# of params	Type	# of layers	# of heads	ffn_dim	hidden_dims
KoBART-base	124M	Encoder	6	16	3072	768
		Decoder	6	16	3072	768

토크나이저

- 공백 제거 및 특수문자 처리(인코딩)
- 서브워드 단위로 텍스트 토큰화
- 문장의 시작과 끝 <s>, <eos> 토큰 추가
- 패딩 및 트렁케이션
- 숫자 인코딩
- 어텐션 마스크 (패딩과 실제 토큰 0, 1로 구분)

성능

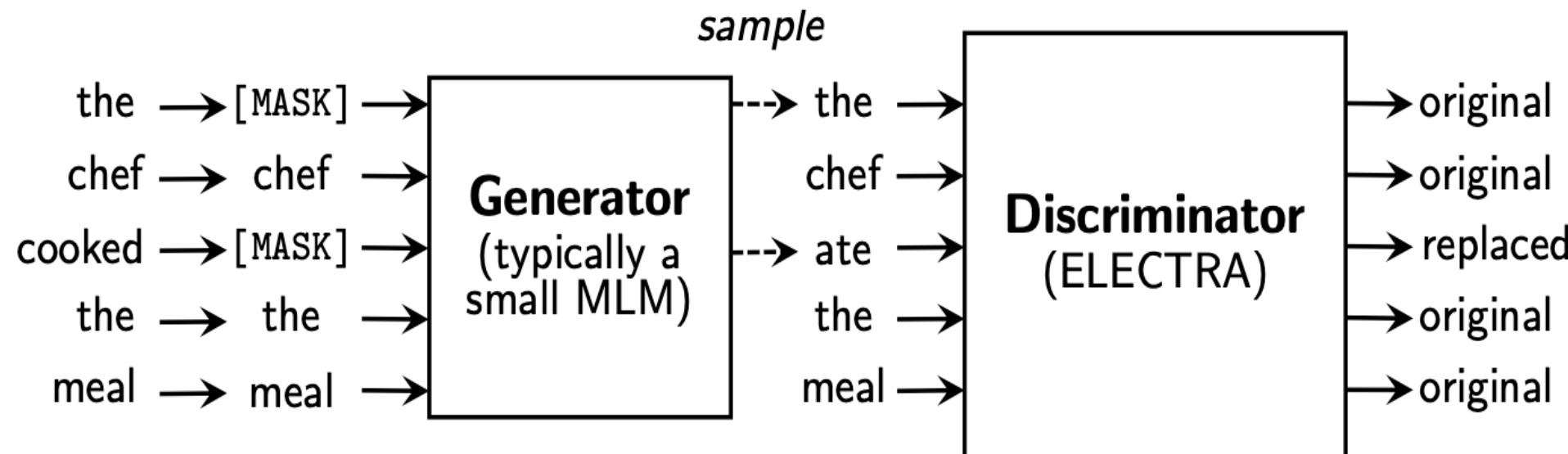
- [10 epochs]
- Loss: 0.03
 - Accuracy: 0.99
 - F1-score: 0.99
 - Leaderboard
f1-score: 0.13

오류 과정

- 문장의 끝에 <eos> 토큰이 생성이 돼도 인식X
-> 사전훈련 토크나이저로 처리함
- Dataset(5000개)기준, 1 epoch당 약 1시간
-> Tesla T4 사용 시 약 2분으로 단축
- 생성형 작업에 특화되어 있음
-> 분류용 아키텍처 변형 필요
- Batch size 오류 발생
-> Batch size 수정하여 최적화
- 용량이 커서 kernel이 자주 끊김
-> 체크포인트로 epoch 결과 저장

→ 10 epoch 훈련 결과, 약 0.99의 높은 F1-score를 달성했지만 최종 리더 보드 점수는 0.13105로 생성형 작업에 특화된 모델은 분류 작업에 부적합하다고 판단

4 KoELECTRA-Base-v3 (Pre-trained)



모델구조

Replace Token Detection 방식 사용.

[Generator – Discriminator \(생성자-판별자\) 네트워크](#)

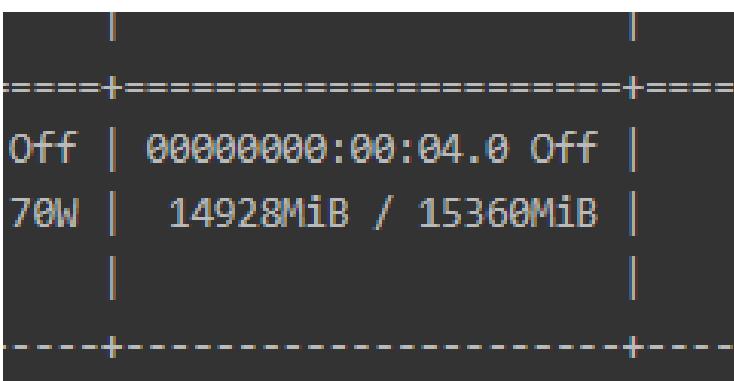
Generator에서 나온 Token 확인, 후 Discriminator에서 'real', 'fake' 진위를 판별하는 방법으로 학습

[BERT 계열의 WordPiece 토크나이저](#)

- 단어를 부분 단위로 분리하여 처리
→ 자주 등장하는 단어를 하나의 단위로 처리

장점: 적은 연산량, 분류 작업 고성능, 파인튜닝 효율성

단점: 생성 작업 적합X, 긴 텍스트를 처리못함



오류과정

문제:

모델 구조가 복잡하고 모든 input을 받기에 메모리 사용량이 많이 듈다.

메모리 부족 문제 때문에 진도를 나가기 버거웠다.

성능

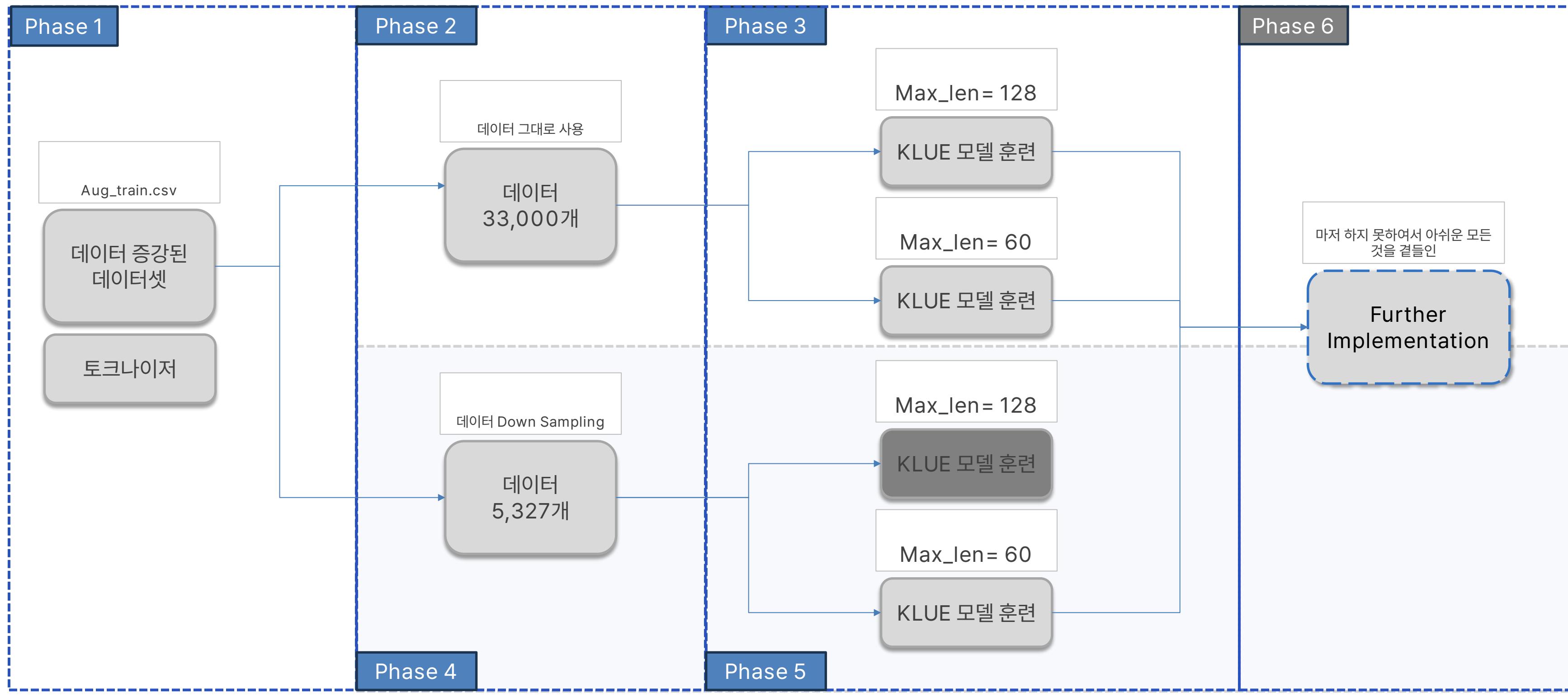
- Loss: 0.0590
- Accuracy: 0.9850
- F1 Score: 0.9850
- Score: 0.17298

→ 처음 KoGPT를 사용하려다가, KoBART와 같은 시퀀스 모델인 것을 확인하여 텍스트 분류 작업이 뛰어난 KoELECTRA 모델로 전환.
긴 텍스트를 잘 처리하지 못하여 전처리와 MAX_Length을 줄여가며 성능을 높여야겠다고 판단. 파인튜닝이 필요할 것 같다.

3-1. 모델 개선 과정 :

모델 개선과정

개선을 위한 실험은 크게 데이터 다운샘플링 여부, 토큰 최대 길이(Max_len)에 따른 결과 차를 비교관찰하여 개선하려 함



3-2. 모델 개선 과정 : 실험내용(Phase 2~3)

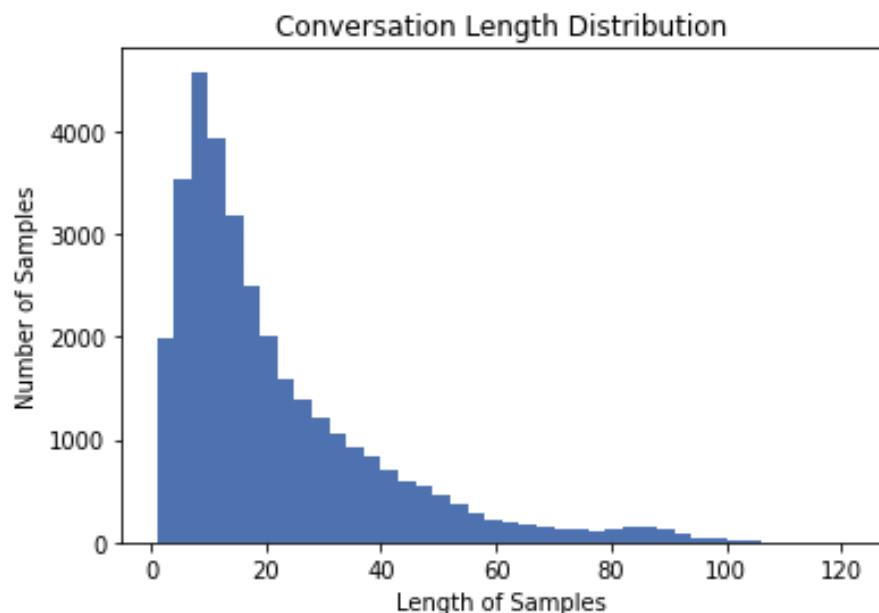
모델 개선과정

데이터 DOWN SAMPLING하지 않을 때 토큰 Max length 단축이 어떤 영향을 주는지 확인

과한 Max length로 인한 정보손실

문장 최대 토큰 length 커버하고자, Max length 128 설정 → 평균 length 21 고려하면 정보 손실 우려
⇒ Max length 128 인 경우와 60인 경우 Kaggle점수 비교하고자 함.

```
# 텍스트 데이터를 토큰화하고 각 샘플의 토큰 길이를 구함
token_lengths = [len(tokenizer.encode(text, truncation=True, padding='max_length', max_length=128)) for text in data['conversation']]
```



Max length, 왜 60인가?

- WordPiece 토큰ナイ저를 사용하면 단어가 서브워드 단위로 나눠 처리하여, 이를 감안하여 길이 설정하는 것 좋지만,
- 과하면 불필요한 연산으로 인한 속도저하, 정보손실 발생
- 고로, 토큰ナイ저 특성 고려하여 상위75% 커버하는 60 값으로 설정

Kaggle Leader Board

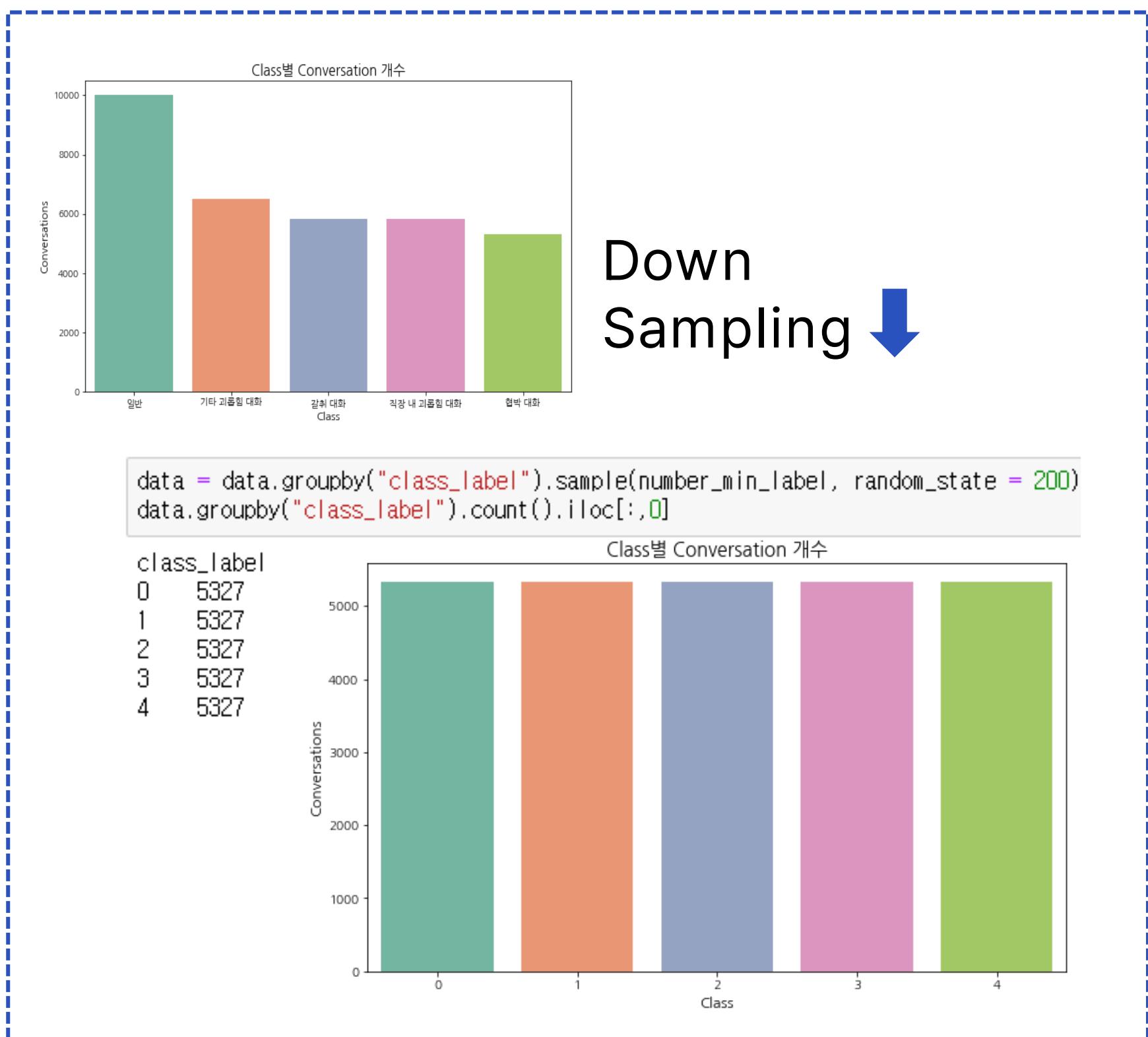
대화의 최소 길이 : 1
대화의 최대 길이 : 121
대화의 평균 길이 : 21.1943217

	submission_241008_0116.csv	0.24099
	kluebert_ex.csv	0.03104

3-3. 모델 개선 과정 : 실험내용(Phase 2~3)

모델 개선과정

데이터 다운 샘플링 후 평균 길이의 토큰 길이 설정



불균형 데이터 → 과대적합 문제
해결 - 균형 잡힌 데이터셋

다운샘플링 되면서 정보 손실 > 오버샘플링 하여 과대적합



다른 방법은?

- 오버 샘플링
- SMOTE
- 가중치 적용

Kaggle Leader Board

submission_241008_0116.csv
Complete (after deadline) · yongyii · 1m ago

0.24099

kluebert_ex.csv
Complete · mxnhxk · 1h ago

0.03104

3-3. 모델 개선 : Further Implementation

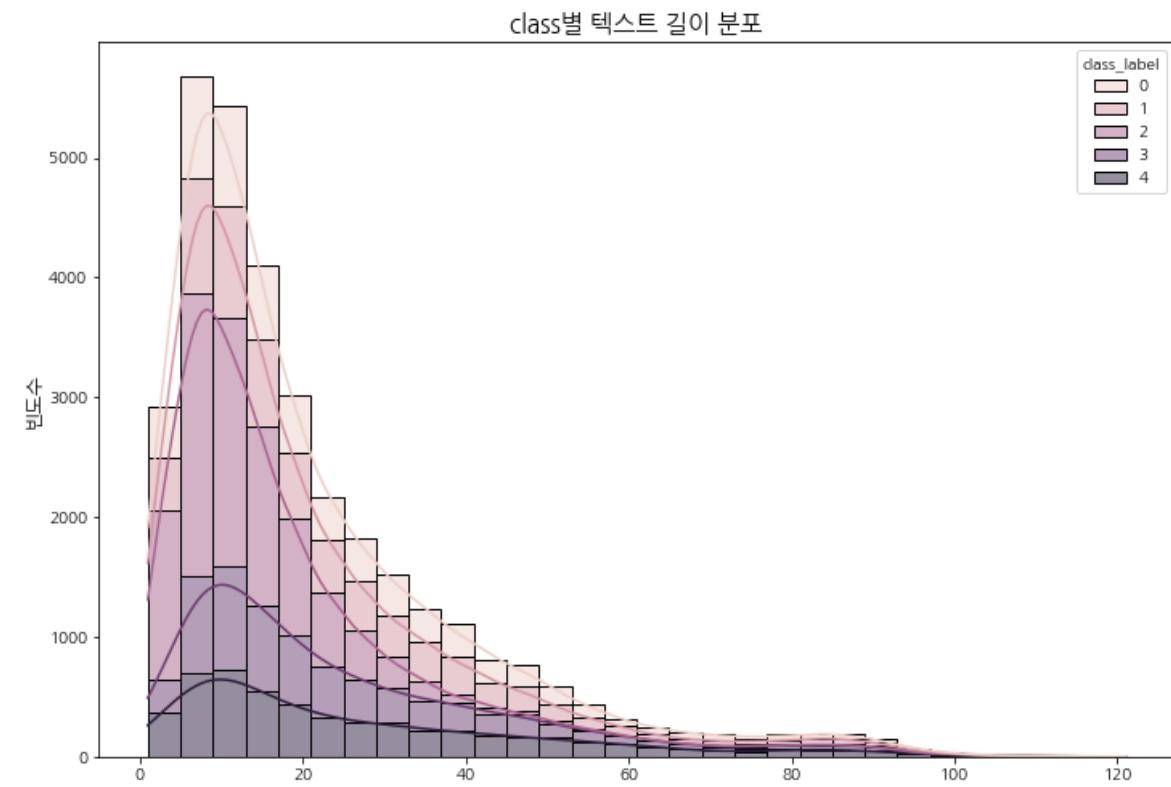
모델 개선과정

해당 프로젝트 추가 진행 시, 개선할 사항

Data

[전처리 추가적 실험]

- 일반 데이터 텍스트 길이 분포 불균형



- 형태소 분석기 사용 유무 성능 비교
- 불용어 처리 유무 성능 비교

Model

[모델 추가적 실험]

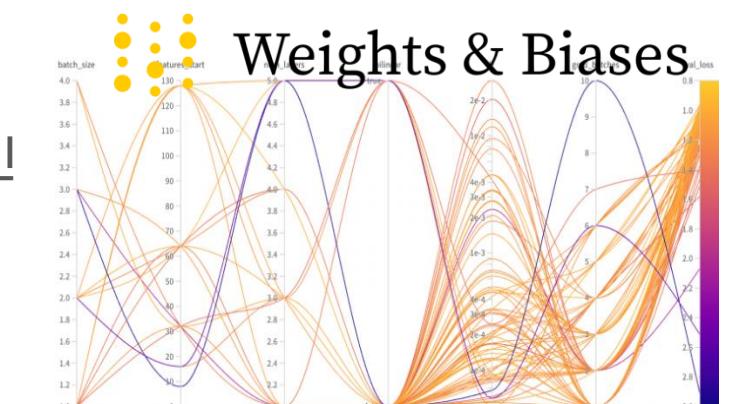
- 분류 특화 모델 별 성능 비교

- 원샷 러닝(One-Shot Learning)
데이터 효율성 : 소량의 데이터로 학습 가능
유사성 기반 : 문장 간 유사성을 통해 문맥 이해 및 학습
- HanBERT
- KoBERT



- hyperparameter 최적화

- Wandb 시각화 → epoch 별 실시간 metric 확인
- 핵심 hyperparameter 파악 및 최적화
- Confusion matrix 시각화



- 모델의 layer 변경

4-1. 모델 상품화 & 타겟 마켓 선정

응용

판사들이 학교폭력 사건의 방대한 서류를 신속하게 분류하고 핵심 정보를 추출해, 서류 검토 부담을 줄이고 중요한 정보를 빠르게 파악하도록 돋는 서비스로 모델을 상품화하고자 함.



천종호 판사: "헤어드라이기로 온몸을 뚫어 가지고 바다에 빠뜨린다고 위협하고, 돈을 상납 받지 않으면 때린다고. 112회 1,400만원 가까이 갈취하고. 한 애를 이렇게 집중적으로 괴롭히면 그 아이는 자살 안 한 것이 참 다행이라고 생각하네요."¹⁷

Target & Pain point

타겟 시장 :

법조계 및 교육 관련 법률 시장, LegalTech

타겟 고객:

판사, 변호사, 학교폭력 사건 관련 전문가

Pain Point:

방대한 서류 검토 부담, 중요한 정보의 빠른 파악 어려움, 문서 처리 지연

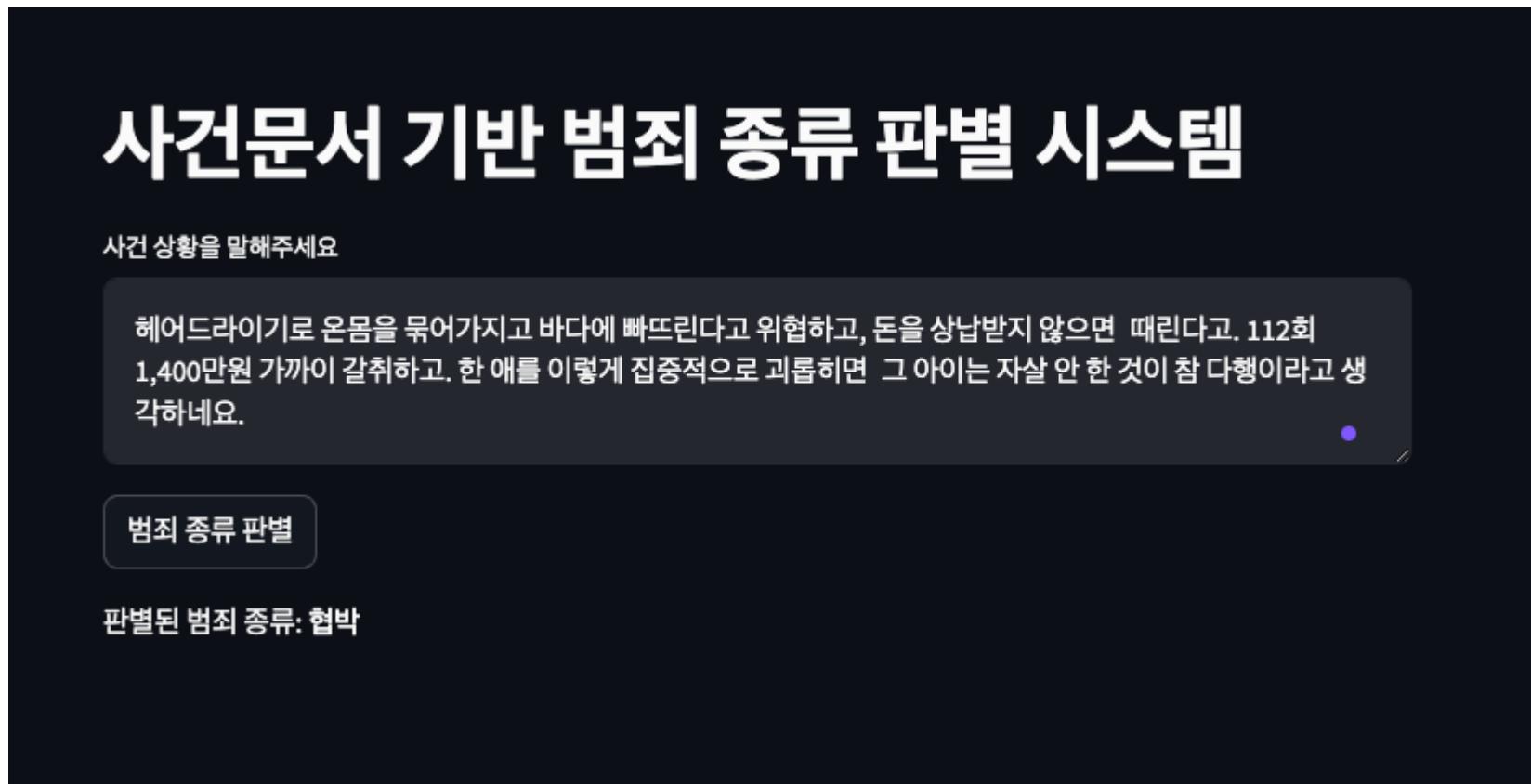
AI 기반 문서 분류 및 요약

법률 문서를 자동으로 분류하고 사건에 관련된 폭력의 종류, 빈도, 강도 빠르게 파악

4-2. 모델 MVP 구상

응용

AI기반 범죄 판별 프로토타입



사건문서를 기반으로 한 범죄 종류를 판별

1. 사건과 관련된 서류문서나 녹취록 등을 입력하면 범죄 유형이 판별
2. 해당 죄명을 판단해 법적 평가가 이뤄지면 죄가 성립되는데 기여



1) 범죄 예측 시스템 flow

1. 대화 채팅 입력 시 대화를 기반으로 한 위험 분류
2. 예측된 클래스에 맞는 위험을 위험 관리 데이터에 쌍임
3. 특정 조건(범죄 가능성)에 부합됐을 경우 관리 대상으로 지정

2) 위협 대화 종류에 따른 대응 방법 제공

1. 대화 채팅 입력 시 대화를 기반으로 한 위험 분류
2. 분류된 위험 클래스에 대한 대응 방법 제공(예) 학교폭력 – 117, 112 신고 메뉴얼 제공

행복한 우리 부서 회식 소회시간



김민혁 팀장
(총괄)

다양한 사전훈련 모델 공부한 경험

DLthon을 하면서 다양한 pre-trained 모델들의 아키텍처를 공부한 것이 가장 기억에 남습니다. 모델 학습을 실행하면서 원인모를 에러도 겪고, 에러가 에러를 낳는 경험을 통해 디버깅하면서 많이 자연스럽게 공부가 됐던 것 같습니다. 저희 팀원들이 저를 믿고 잘 따라와 주시고, 저도 팀원들을 믿으며 의지하였기에 비교적 리더보드 제출 성능은 낮게 나왔지만 DLthon 자체를 완성도 높게 마무리한 것 같아 후회 없이 좋았습니다.



담안용 책임
(PO)



정서연 책임
(데이터)



이준범 책임
(엔지니어)

DL프로젝트 현장 같은 경험

모델이 제 성능을 다하도록 일하는 모든 과정과 더불어 프로젝트를 선보이는 발표까지 정말 현장 프로젝트의 E2E 경험한 것 같아서 힘들지만 즐거웠습니다. 새벽 야간 작업, 동료들 놀리고, 야식 자랑한 것과 같은 소소하고 귀여운 추억도 얻어서 개인적으로 행복했습니다. 모든 면에서 유익한 경험이었습니다.

My First DLthon

첫 디엘톤인데 훌륭한 그루분들을 만나 다양한 실험과 경험을 해서 즐거웠습니다. 앞에서 이끌어주고 중간중간 캐치업 해주시는 그루분들 덕분에 정말 팀플 다운 팀플을 한 것 같습니다. 연구를 하는 느낌으로 모델 선정부터 했는데 모든 선택에 있어서 타당한 근거를 정리하는게 생각보다 큰 도움이 된 것 같습니다. 정리의 중요성을 한번 더 느꼈고, 실험 결과가 예상과는 다를 때가 많았는데 이에 대해 근거를 생각하는 과정도 유익했습니다.

데이터 증강 경험

여러가지 데이터 증강 방식을 사용해보며 각 방식 별 장단점을 비교할 수 있었습니다. 또한 모델 성능을 높이기 위하여 랜덤 샘플링, 토크나이저 전처리 방식 확인 등의 과정을 수행하였습니다. 데이터 전처리 과정에서 오류도 있었고 증강 처리 시간이 오래 걸렸지만 결과적으로 약 33000개의 데이터를 생성할 수 있었습니다. 향후에는 강화학습 및 어텐션 스코어 부분에 대해서도 다양한 실험을 해보고 싶습니다.



- **Social Media : [Notion]**

<https://www.notion.so/modulabs/2-1125c8e5427d81c8924ffad243020c87>

- **Website : [github, Lark]**

https://github.com/xvihaan/aiffel_dlthon_c2

<https://ksg2zirmtxp6.sg.larksuite.com/drive/home/>



THANK YOU

C2 민혁 없는 민혁팀
