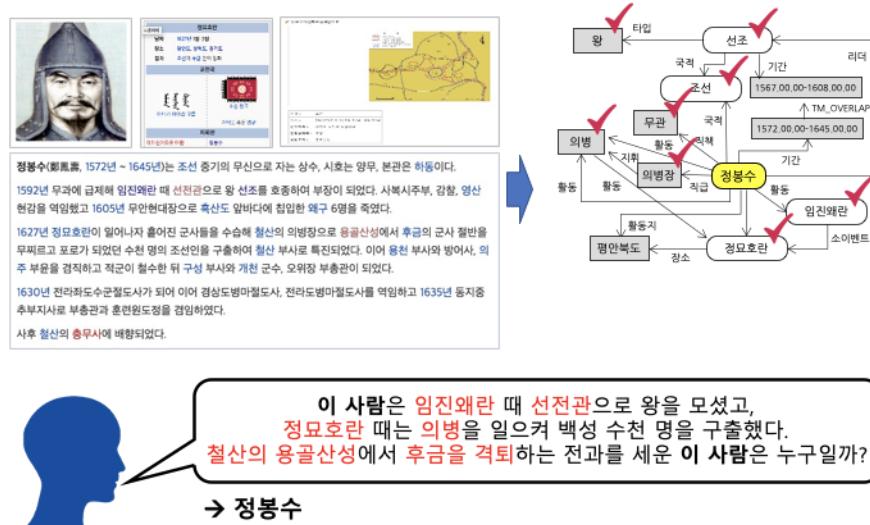


Wrap-Up Report

Team : 자만추 (NLP-01)

1. 프로젝트 개요



A. 프로젝트 주제



관계 추출(Relation Extraction)

- 문장의 단어(Entity)에 대한 속성과 관계를 예측하는 문제
- 지식 그래프 구축을 위한 핵심 구성 요소로, 구조화된 검색, 감정 분석, 질문 답변하기, 요약과 같은 자연어처리 응용 프로그램에서 중요하게 여겨짐
- 비구조적인 자연어 문장에서 구조적인 triple을 추출해 정보를 요약하고, 중요한 성분을 핵심적으로 파악할 수 있습니다

```
sentence: 오라클(구 썬 마이크로시스템즈)에서 제공하는 자바 가상 머신 말고도 각 운영 체제 개발자가 제공하는 자바 가상 머신 및 오픈소스로 개발된 구형 버전의 온전한 자바 VM도 있으며, GNU의 GCJ나 아파치 소프트웨어 재단(Apache Software Foundation)의 하모니(Harmony)와 같은 아직은 완전하지 않지만 지속적인 오픈 소스 자바 가상 머신도 존재한다.
```

```
subject_entity: 썬 마이크로시스템즈  
object_entity: 오라클
```

```
relation: 단체:별칭 (org:alternate_names)
```

B. 개발 장비 및 환경

GPU	Tesla v100 (6인 1팀)		
개발 환경	Ubuntu 18.04		
협업 툴	Github, Notion, Slack, Wandb		

pandas	1.1.5	numpy	1.19.2
torch	1.7.1	scikit-learn	0.24.2
transformers	4.10.0	tokenizers	0.10.3
black	23.3.0	tqdm	4.64.1
omegaconf	2.3.0	wandb	0.15.1

C. 데이터셋

- Train 데이터 개수 : 32,470개
- Test 데이터 개수 : 7,765개
 - 정답 라벨의 경우 blind = 100으로 임의로 표현함
- 데이터 유형 (Source)
 - wikipedia
 - wikitree
 - policy_briefing
- dict_label_to_num.pkl: 문자 label과 숫자 label로 표현된 dictionary (총 30개 classes)
- dict_num_to_label.pkl: 숫자 label과 문자 label로 표현된 dictionary (총 30개 classes)

```
'no_relation': 0, 'org:top_members/employees': 1,
'org:members': 2, 'org:product': 3, 'per:title': 4,
'org:alternate_names': 5, 'per:employee_of': 6, 'org:place_of_headquarters': 7,
'per:product': 8, 'org:number_of_employees/members': 9, 'per:children': 10,
'per:place_of_residence': 11, 'per:alternate_names': 12, 'per:other_family': 13,
'per:colleagues': 14, 'per:origin': 15, 'per:siblings': 16, 'per:spouse': 17,
'org:founded': 18, 'org:political/religious_affiliation': 19,
'org:member_of': 20, 'per:parents': 21, 'org:dissolved': 22,
'per:schools_attended': 23, 'per:date_of_death': 24, 'per:date_of_birth': 25,
'per:place_of_birth': 26, 'per:place_of_death': 27,
'org:founded_by': 28, 'per:religion': 29
```

Relation Class	Description
<i>no_relation</i>	No relation in between ($e_{\text{subj}}, e_{\text{obj}}$)
<i>org:dissolved</i>	The date when the specified organization was dissolved
<i>org:founded</i>	The date when the specified organization was founded
<i>org:place_of_headquarters</i>	The place which the headquarters of the specified organization are located in
<i>org:alternate_names</i>	Alternative names called instead of the official name to refer to the specified organization
<i>org:member_of</i>	Organizations to which the specified organization belongs
<i>org:members</i>	Organizations which belong to the specified organization
<i>org:political/religious_affiliation</i>	Political/religious groups which the specified organization is affiliated in
<i>org:product</i>	Products or merchandise produced by the specified organization
<i>org:founded_by</i>	The person or organization that founded the specified organization
<i>org:top_members/employees</i>	The representative(s) or members of the specified organization
<i>org:number_of_employees/members</i>	The total number of members that are affiliated in the specified organization
<i>per:date_of_birth</i>	The date when the specified person was born
<i>per:date_of_death</i>	The date when the specified person died
<i>per:place_of_birth</i>	The place where the specified person was born
<i>per:place_of_death</i>	The place where the specified person died
<i>per:place_of_residence</i>	The place where the specified person lives
<i>per:origin</i>	The origins or the nationality of the specified person
<i>per:employee_of</i>	The organization where the specified person works
<i>per:schools_attended</i>	A school where the specified person attended
<i>per:alternate_names</i>	Alternative names called instead of the official name to refer to the specified person
<i>per:parents</i>	The parents of the specified person
<i>per:children</i>	The children of the specified person
<i>per:siblings</i>	The brothers and sisters of the specified person
<i>per:spouse</i>	The spouse(s) of the specified person
<i>per:other_family</i>	Family members of the specified person other than parents, children, siblings, and spouse(s)
<i>per:colleagues</i>	People who work together with the specified person
<i>per:product</i>	Products or artworks produced by the specified person
<i>per:religion</i>	The religion in which the specified person believes
<i>per:title</i>	Official or unofficial names that represent the occupational position of the specified person

KLUE: Korean Language Understanding Evaluation RE 데이터셋 참고

2. 프로젝트 팀 구성 및 역할

- 김효연 : 데이터 분석, 단순 증강, val_dataset 예측 코드 작성
- 서유현 : 코드 리뷰어, 베이스 코드 작성 및 일반화 개선
- 손무현 : 코드 리뷰어, 데이터 분석 및 증강
- 이승진 : Loss 리서치, 편의 기능 제공, TAPT 연구
- 최규빈 : 데이터 튜닝, 모델 서치, 양상블
- 황지원 : PM, Git 코드 버전 관리, 모델링 및 임베딩 작업 진행

3. 프로젝트 수행 절차 및 방법

A. 팀 목표 설정

(1주차) 대회 간 팀 규칙을 설정하였고 Git Flow에 맞게 브랜치를 설정, PR(Pull-Request)에 대한 규칙을 정하였습니다. 팀 내에서 PM, 코드 리뷰어, 모델팀/데이터 팀 등 각각 역할을 분담하여 작업을 진행하였습니다. 전체적인 데이터 분포와 라벨을 확인하고 분석을 진행하였습니다. 또한 Focal Loss와 Special Entity Marker에 대한 탐색을 진행하였습니다.

(2주차) 이전 주차에 이어서 추가로 파라미터와 Loss에 대해 테스트를 진행하였습니다. 또한 데이터 부분에서 데이터 증강을 진행하였고 그와 함께 일부 학습 데이터에 대해 직접 데이터를 확인하고 분석하고 라벨링을 다시 하는 작업을 진행하였습니다.

(3주차) 마지막 주차인 만큼 다양한 테스트를 진행하기보다는 성능을 높일 수 있을 만한 세부 테스트를 진행하였습니다. TAPT와 Entity Marker, 프롬프트(Prompt)에 대해 코드 작성 후 테스트를 진행하였습니다. 그리고 Voting 방식을 채택해 soft/hard voting을 적용한 모델 양상을 테스트를 마지막으로 진행하고 대회를 마무리하였습니다.

B. 프로젝트 구성

```
level2_klue-nlp-01
├── code
│   ├── config      # 학습/테스트를 진행하는 config 파일
│   │   └── config.yaml    # 기본 Task를 위한 config 파일
│   ├── constant    # 코드 내에서 사용하는 상수
│   │   └── CONFIG.py
│   ├── custom
│   │   ├── custom_dataset.py
│   │   ├── custom_model.py
│   │   ├── custom_trainer.py
│   │   └── entity_special_token.txt
│   ├── log          # 모델 학습을 진행한 결과물
│   │   └── #
│   ├── prediction   # Inference를 진행한 검증 결과물
│   │   └── #
│   ├── utils
│   │   ├── config.py    # config 전처리 작업
│   │   └── log.py       # log 전처리 작업
│   ├── dict_label_to_num.pkl
│   ├── dict_num_to_label.pkl
│   ├── load_data.py
│   ├── train.py
│   ├── inference.py
│   └── run.py
└── requirements.txt
dataset (.gitignore로 공유되지 않도록 설정)
├── train.csv
└── test_data.csv
.gitignore
README.md
```

C. 프로젝트 타임라인

구분	내용	일정																
		5/2	5/3	5/4	5/5	5/6	5/7	5/8	5/9	5/10	5/11	5/12	5/13	5/14	5/15	5/16	5/17	5/18
계획	팀 규칙 회의																	
	Git Flow 회의																	
	코딩 컨벤션 작업																	
분석	데이터 분석																	
	모델 테스트 진행																	
개발	베이스 코드 작성																	
	Focal Loss 테스트																	
	Special Entity Token 테스트																	
	데이터 라벨링																	
	프롬프트, Multi-sentence 테스트																	
	TAPT 테스트																	
	Entity Marker 테스트																	
	모델 양상을 진행																	
테스트	모델링 최종 테스트 진행																	
	최종 결과 종합																	
	보고서 및 회고 작성																	

4. 프로젝트 수행 결과

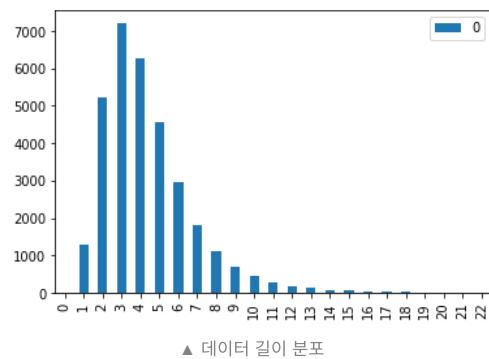
A. 프로젝트 실험 및 결과

1. 데이터 분석 & 증강 & 전처리

- 데이터 source와 특징
 - 이번 데이터 source는 3가지(wikipedia, wikitree, policy_briefing)이고, 백과사전과 뉴스에 있는 자료를 바탕으로 만든 데이터 셋입니다.

- 전처리를 위해 데이터를 탐색한 결과 XXX기자, XXX신문 등 뉴스에 있을법한 필요 없는 불용어는 없었고, 맞춤법, 띄어쓰기도 정교하기 때문에, UNK토큰을 탐색한 후 전처리를 진행했습니다.
- **raw 데이터 확인**
 - 팀원들과 2천개의 train데이터를 확인해본 결과, 약 100개 정도의 **라벨이 애매한 데이터셋**을 확인했고, 모델을 분석하면서 **entity의 타입에도 오류**가 있음을 확인했습니다.
 - 가족시리즈(per:spouse, per:children, per:siblings, per:other_family, per:parents)는 주어진 문장만으로는 관계를 정확하게 예측할 수 없는 것들이 많았고, 사람이 봐도 판단하기 어려웠습니다
 - employee_of와 per:product는 중복으로 해석할 가능성이 있었습니다. 또한, 라벨의 설명과는 다르게, 단체나 집단을 만드는 경우도 있었습니다.
 - 팀원들끼리 모여서, 확실하게 틀렸다고 생각되는 라벨들을 수정하는 작업을 진행했습니다.
- **데이터 중복**
 - 데이터 중복 검사 결과, **42개의 데이터 중복**을 확인했고, **5개의 라벨만 다른 데이터(오류)**를 확인했습니다.
- **텍스트 길이**
 - 텍스트 데이터의 분포를 확인한 결과 **가장 긴 텍스트의 길이가 455자, 평균 97자임**을 확인했고, 토크나이저의 최대 길이가 512이므로, 길이 때문에 문제가 발생하지는 않겠다고 판단했습니다.

	0
count	32470.000000
mean	97.083954
std	47.939902
min	14.000000
25%	64.000000
50%	87.000000
75%	118.000000
max	455.000000

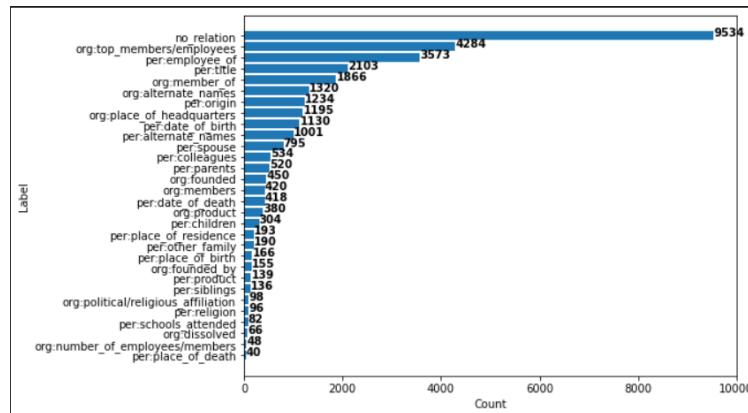


- **데이터 전처리 & UNK 토큰**
 - 특수문자가 연속된 문장이 존재했습니다. → ex) ""안녕하신가""
 - ～～ () ? 《》「」 등, 비슷하지만 다른 특수문자들이 존재했습니다.
 - », «, ←, € 등의 특이한 특수문자가 존재했습니다.
 - 한자, 일본어, 그리스어, 아랍어 등 여러 나라의 언어가 등장했습니다.
 - UNK토큰 분석 결과 → ～～, 특이한 특수문자들, 특이한 한글, 여러나라의 언어가 UNK토큰으로 바뀌는 것을 확인했습니다.



1~5의 분석 결과를 바탕으로 실험을 진행한 결과, UNK토큰으로 바뀌는 비슷하지만 다른 특수문자들과 비교적 많이 등장하는 », «, €만 변경했습니다.

- **데이터 불균형**



▲ no_relation이 압도적으로 많은 데이터가 불균형한 모습을 보이고 있다.

- 데이터의 수를 어떻게 맞추는가에 대한 고민이 많았지만, 무작정 데이터의 비율을 맞추려고 하지 않고 모델을 분석하면서 데이터의 비율과 개수를 조정하면서 프로젝트를 진행하였습니다.

• Validation dataset

- 주어진 train.csv 데이터 중 0.2 비율 만큼의 데이터를 분리하여 validation data로 사용하였는데, 분리할 때, 0.2 비율의 validation data는 기존 train.csv와 동일한 class 비율이 유지되도록 하였습니다. 모든 팀원들이 공통된 validation data를 사용하였습니다.

• best-model 분석과 데이터 단순 증강

- confusion matrix로 제출 스코어가 가장 높은 단일 best-model을 분석한 결과
 - per:product
 - per:place_of_residence
 - per:other_family
- 데이터를 많이 틀린것을 확인했습니다. 공통점은, 모두 데이터의 수가 적었고, 사람이 봐도 판단하기 어려운 데이터 였습니다. 특히, per:place_of_residence는 origin으로 예측한 비율이 0.4를 넘은것을 확인했습니다.
- 그래서 증강실험을 진행했고, 결과적으로 3개의 라벨을 가진 훈련 데이터를 2배 증강시킨 결과, 전체적인 스코어가 상승하는 모습을 확인할 수 있었습니다. 하지만 정작, 데이터를 늘린 3개의 라벨의 정답률은 낮아졌는데, 데이터 라벨에 오류가 있기 때문이라고 판단했습니다.
- 그리고 epoch이 늘어날수록, loss가 증가하는 현상이 발생했는데, confusion matrix를 확인해 본 결과, 위 3개의 라벨을 가진 데이터들은 학습할수록 정답률이 떨어졌습니다. 특히 no_relation에 과적합 되는 모습을 보였습니다.

2. Entity Marker Representation

Method	Input Example	Token
Entity mask	[SUBJ-PERSON] was born in [OBJ-CITY]	[SUBJ-TYPE], [OBJ_TYPE]
Entity marker	[E1] Bill [/E1] was born in [E2] Seattle [/E2].	[E1], [/E1], [E2], [/E2]
Entity marker (punct)	@ Bill @ was born in # Seattle #.	@, #
Typed entity marker	<S:PERSON> Bill </S:PERSON> was born in <O:CITY> Seattle </O:CITY>.	<S:TYPE>, </S:TYPE>, <O:TYPE>, </O:TYPE>
Typed entity marker (punct)	@ * person * Bill @ was born in # ^ city ^ Seattle #.	@, #
Kor Typed entity marker	<S:사람> Bill </S:사람> was born in <O:도시> Seattle </O:도시>.	<S:타입>, </S:타입>, <O:타입>, </O:타입>
Kor Typed entity marker (punct)	@ * 사람 * Bill @ was born in # ^ 도시 ^ Seattle #.	@, #, 타입

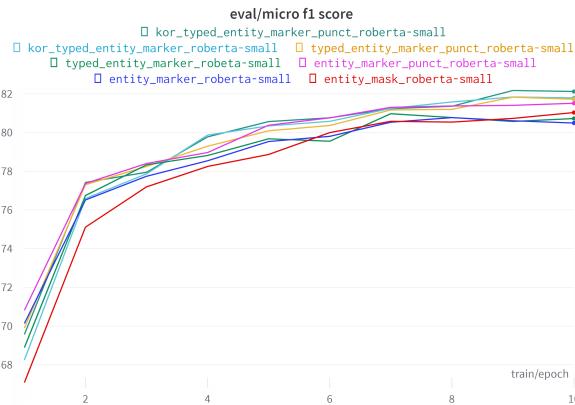
• 가설

- Entity를 나타내는 부분에 Special Token으로 표시를 한다면 좀더 Entity에 집중하여 학습이 이루어지고 성능향상이 있을 것이다.

• 내용

- Subject Entity Token과 Object Entity Token을 찾아서 Special Marker로 해당 토큰들을 감싸 사용
- Kor Marker의 경우, KLUE 한국어 데이터셋에 맞추어 기존의 Marker에서 Type 부분을 한국어로 변환해 토큰으로 사용하였습니다.

- 결과



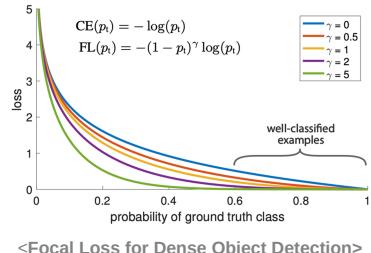
Rank	Method	Score
1	kor_typed_entity_marker_punct	82.121
2	kor_typed_entity_marker	81.789
3	typed_entity_marker_punct	81.719
4	entity_marker_punct	81.511
5	entity_mask	81.031
6	typed_entity_marker	80.723
7	entity_marker	80.491

- KLUE Dataset에 각 Entity Marker를 사용하여 베이스 모델을 통해 테스트를 진행하였습니다. 결과를 보면 한국어 데이터셋인 만큼 한국어로 Marker를 사용한 것이 좋은 점수를 보여주었습니다. 위 결과를 토대로 데이터의 전처리를 진행해 모델의 성능을 더욱 높이고자 합니다.

3. Loss 최적화 실험

1) 손실함수(Focal)

- 현실의 데이터는 불균형한 분포를 띠고 있는 것이 대다수이며 저희에게 주어진 데이터 또한 마찬가지였습니다. 따라서, 모든 샘플에 동일한 가중치를 주어서 손실을 구성하는 기존 Cross Entropy 손실함수는 효과적으로 학습이 되지 않을 것이라는 추측하였습니다.



- 불균형에는 빈도의 불균형도 있지만 모델의 예측확률로 접근하는 Easy/Hard 샘플의 불균형도 존재하였습니다. confusion matrix를 확인해보면 100%에 수렴하는 확률로 학습이 잘되는 클래스가 있는 반면에 50% 확률도 채 넘지 못하는 클래스도 존재함을 확인할 수 있고, 모델이 예측하기 어려운 클래스에 더욱 상대적으로 높은 가중치를 부여하게 되는 Focal Loss를 Cross Entropy Loss와 비교실험 하였습니다.
- 기본 베이스라인에서 주어진 파라미터를 기준과 약식 튜닝 후 비교 결과 다음과 같았습니다.

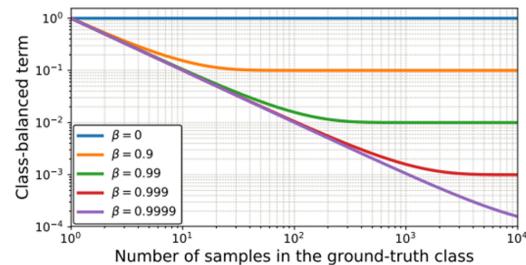
	Base parameter	after tuning(short form)		
	CE (5e-5)	Focal (5e-5)	CE (5e-6)	Focal (1.3e-5)
micro F1	57.54	61.98	68.07	71.73
AUPRC	62.35	66.57	71.96	76.70

- 파라미터를 통제시키고 같은 조건에서 비교를 원하였지만, CE에서 훌륭하게 작동하는 학습률이 Focal에 적용하니 전혀 수렴하지 못하는 등 CE와 어울리는 파라미터 조합과 Focal과 어울리는 파라미터 조합은 예상보다 많은 차이가 났습니다. 동등한 조건의 비교는 이 실험에서는 적합하지 않았고 sweep를 통한 튜닝을 약식으로 10회 진행하여 찾은 조합으로 비교하였습니다. 시간의 제약이 있었기 때문에 이 둘

의 튜닝에만 많은 시간을 투자할 수는 없었고, 따라서 이 실험의 신뢰도는 높지 않습니다. 하지만 이 실험을 바탕으로는 Focal Loss가 두 지표에서 모두 향상된 성능을 보여주었으므로 Focal Loss가 우리 태스크에서 불균형 문제를 해결해줌으로써 성능 향상을 야기할 것이라는 대립 가설을 채택하게 되었습니다.

2) 손실함수(LDAM)

- 위에서 언급한 Focal Loss는 모델의 예측 확률을 토대로 가중치를 re-weighting 하여 모델이 어려워하는 클래스에 대해 상대적 가중치를 높이는 방식이었다면 이와는 수직적인 방식인 'Label-Distribution-Aware Margin Loss' 이하 LDAM Loss는 Margin을 조정하여 불균형을 해소하려는 모티베이션으로부터 유도된 방법이었습니다.
- LDAM을 실험하게 된 배경으로는 본 논문에서 수학적으로 탄탄하게 유도된 최적의 트레이드오프 Margin이 있었기 때문에 Focal Loss와 비교하여 얼마나 차이가 있을지에 대한 호기심으로부터 출발하였습니다.
- 논문에서 제시한 클래스 빈도의 역수를 스케줄링에 반영하는 DRW 스케줄링 방법을 따랐으며 추가로 같은 해 나왔던 논문 중 빈도의 역수인 alpha 항을 정보의 중복을 고려하여 성능을 향상 시킨 Effective Number 방식으로 대체하여 실험하였습니다. 이는 단순 클래스 빈도의 역수를 하는 대신 정보의 중복(또는 기대 부피) 차원에서 가중치를 고려한 방법으로 수식에서 beta 값이 1에 수렴하면 클래스 빈도의 역수인 단순 alpha와 같아지고 0에 수렴할수록 클래스 간 가중치는 동일하게 되는 구조를 갖고 있습니다(오른쪽 그림 참조).



<Class-Balanced Loss Based on Effective Number of Samples>

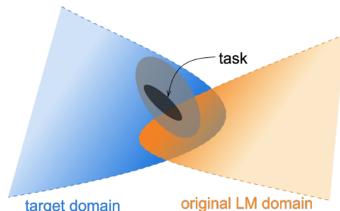
- Focal Loss와 LDAM Loss의 비교 실험 결과는 다음과 같습니다.

	Focal (1.3e-5)	LDAM + DRW (1.3e-5)
micro F1	71.73	71.81
AUPRC	76.70	74.54

- micro F1은 소폭 상승하였고, AUPRC는 2점 가량 하락하였습니다. 이에 대한 해석으로는 여러가지가 있을 수 있겠지만, AUPRC가 낮아져서 recall과 precision에 대한 트레이드오프 곡선에서 조금 손해를 보았어도 micro F1이 소폭이라도 늘어서 오분류 케이스에 대한 개수 자체가 줄었다는 점을 긍정적으로 평가하여 Focal을 기각하지는 않고 둘 다 사용할 수 있도록 하여 연구하는 방향으로 진행토록 하였습니다.

4. TAPT

- 사전학습 모델은 우리의 태스크와는 아무래도 동떨어진 데이터라고 할 수 있습니다. 따라서 'Don't Stop Pretraining: Adapt Language Models to Domains and Tasks' 의 논문 내용을 바탕으로 우리의 태스크에 더 잘 적응할 수 있도록 우리가 갖고 있는 일부 데이터로 further pretraining을 진행하는 방식을 시도하였습니다.

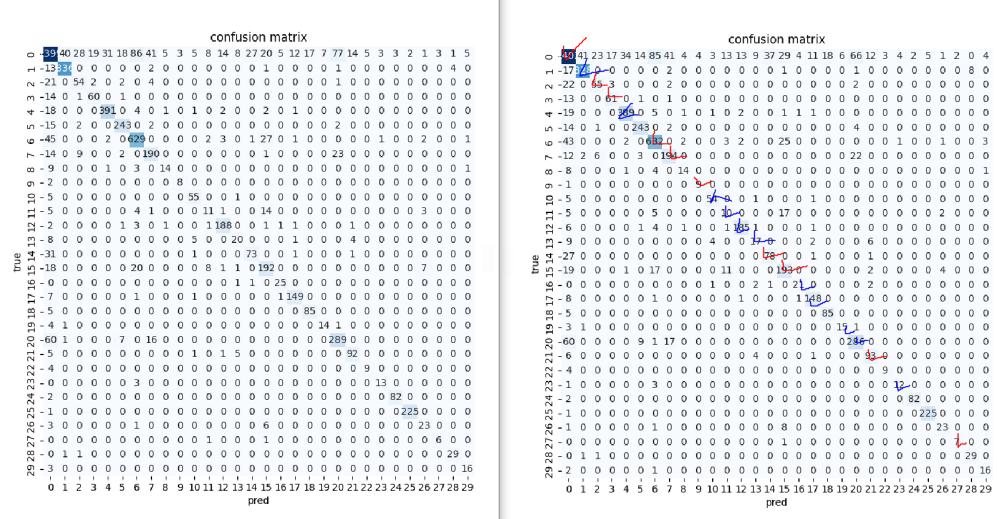


<Don't Stop Pretraining: Adapt Language Models to Domains and Tasks>

- 방법은 간단하였으며 **klue/RoBERTa-large** 모델을 라벨이 없는 test 데이터로 adaptive pretrain과정을 진행하였습니다. 논문에서는 **DAPT**, **TAPT**, **DAPT+TAPT**라는 세 가지 방식을 비교 실험하였지만, 우리의 상황에서는 DAPT를 할 만한 데이터 량이 확보되지 못했기 때문에 TAPT만 진행하였습니다.
- 논문 내용 중 TAPT는 같은 task일 경우에만 성능 향상을 보였고 cross-task transfer 관점에서 TAPT는 오히려 성능 저하를 야기하였다고 명시되어 있습니다. 저희가 진행하려는 TAPT는 test.csv 데이터를 쓰기 때문에 같은 task이므로 성능 향상을 꾀할 것으로 기대하였습니다. 결과는 오른쪽과 같습니다.

	3epoch TAPT
micro F1	-0.5
AUPRC	+2.0

- 3epoch만 진행하여 비교해본 결과 AUPRC는 2점가량 늘었으나, micro F1점수에서는 오히려 감소한 국면을 보여주며 이후 10epoch futher pretraining을 진행해보았어도 성능에는 유의미한 차이는 없었음을 관찰하였습니다.



- 좌(Before TAPT), 우(After TAPT)의 **confusion matrix**를 비교한 것이며 빨간색(+)과 파란색(-)로 얼마나 변했나 직접 비교해본 결과 증가한만큼 감소하는 즉, 기대와 달리 전체 성능상으로 크게 유의미한 변화를 야기하지는 않았습니다.
- 결론적으로 TAPT한 모델은 성능의 변화가 있다고 하기 어려웠지만 다양성이 필요한 양상을 모델로 써는 활용할 가치가 있다고 판단하여 최종 제출시 양상을 모델 중 일부로 반영하였습니다.
- epoch를 늘리거나, Knn-TAPT 방식의 실험은 시간 관계상 진행하지 못하였고 미래 프로젝트 TODO List로 남겼습니다. 또한 UDA, UST 방법으로 레이블링 되지 않은 데이터를 활용하는 방법도 연구해볼 계획으로 남겼습니다.

5. P-tuning 실험

- [GPT Understands, Too, 2021] 논문의 continuous space(내용을 인간이 직접 작성하는 discrete space와는 상반되는 개념) 기반 P-tuning 활용, [Context Aware Named Entity Recognition and Relation Extraction, 2022] 논문의 주변 text를 활용하였을 때 RE task performance가 증가하는 점에서 착안해 우리가 가진 continuous space 상의 entity 정보(word,type)를 prompt에 추가하여 학습을 시도하였습니다.

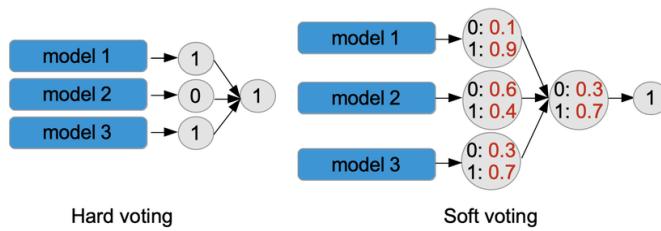
No.	Type	Example	micro_f1_score
1	이 문장에서 {obj_word}는 {sbj_word}의 {trans[obj_type]}이다. 이 때, 이 둘의 관계는	이 문장에서 SM C&C는 전현무의 단체이다. 이 때, 이 둘의 관계는?	86.37
2	이 문장에서 {sbj_word}는 {obj_word}의 {trans[sbj_type]}[{sbj_type}]이다. 이 때, 이 둘의 관계는	이 문장에서 전현무는 SM C&C의 사람[PER]이다. 이 때, 이 둘의 관계는?	86.06
3	x	아무것도 적용 x (type entity marker만 사용)	85.76
4	이 문장에서 {obj_word}와 {sub_word}의 관계는?	이 문장에서 SM S&C와 전현무의 관계는?	86.03
5	'이 문장에서 {obj_word}는 {trans[sbj_type]}인 {sbj_word}의 {trans[obj_type]}이다. {obj_word}와 {sbj_word}의 관계는?'	이 문장에서 SM C&C는 사람인 전현무의 단체이다. SM C&C 와 전현무의 관계는?	85.66

- 결과



- 최종적으로 `<kor_typed_entity_marker_punct + "이 문장에서 {obj_word}는 {sbj_word}의 {trans[obj_type]}이다. 이 때, 이 둘의 관계는">`의 형태로 입력 데이터로 하였을 때 가장 우수한 성능을 보여주었습니다.

6. 양상별



- 지금까지 실험했던 모델들의 결과를 바탕으로 다중 학습 모델을 기반으로 한 하나의 모델로 만들고자 하였습니다. 각 개별 모델이 전체 데이터에 대한 성능은 조금 떨어지더라도 특정 분야에 대해 뛰어나다면, 이들을 결합함으로써 예리는 줄이고 전체 성능을 향상시킬 수 있다는 가정하에 진행하게 되었습니다.

• Hard Voting

- 모델들의 예측 결과를 단순히 종합하여 가장 많은 예측 결과를 선택합니다.
- 제출 기준으로 micro_f1 score가 높은 3개를 선택해서 hard voting을 진행한 결과, 리더보드 기준 **micro_f1 score**가 **75.1620** 달성했습니다
- micro_f1 score가 높은 3개의 모델과, auprc가 높은 상위 2개의 모델을 hard voting을 진행한 결과 **micro_f1 score** **74.4427** 달성했습니다

• Soft Voting

- 모델들의 예측 확률을 합산해서 가장 높은 class를 선택합니다.
- 상위 10개 모델들을 soft voting한 결과 **micro_f1 score** **73.8554**, **auprc** **74.3248** 달성했습니다

B. 최종 결과

평가 지표는 **micro F1 score**로 진행되었습니다.

public : 75.1620 ⇒ **private** : 72.9367

9 (-)	NLP-01조		75.1620	76.0884	56
----------	---------	--	---------	---------	----

11
(2 ▾)

NLP-01조



72.9367

72.9288

56

최종 결과물 A (75.1620)

Model	KLUE/RoBERTa-Large
Optimizer	AdamW
Ensemble	Hard voting(3 Model)

최종 결과물 B (73.8554)

Model	KLUE/RoBERTa-Large
Optimizer	AdamW
Ensemble	Soft voting(10 Model)

5. 자체 평가 의견

A. 잘한 점들

- 대회 초기부터 협업을 위해 정한 Git Flow와 브랜치/PR 규칙이 협업을 진행하는데 많은 도움을 주었고 지속적인 코드 리뷰가 가능했습니다.
- 모델 학습 외로 Loss, Parameter와 같은 세부 테스트를 진행한 것도 좋은 성적을 거두는데 기여했다고 생각합니다.
- SOTA 모델만 서칭하는데서 멈추지 않고 가설로 세운 내용에 대해 서칭하고 연구를 진행한 것에 대해 많은 발전을 할 수 있었습니다.
- 프로젝트를 진행하면서 주저하지 않고 팀원 간 날카로운 지적을 해준 것으로 많은 성장이 이루어 졌다고 생각합니다.

B. 시도 했으나 잘 되지 않았던 것들

- Dropout
- Hidden Layer Concatenation
- Bi-label Classification Model
- class imbalance를 해결하기 위한 데이터 증강 (전체적인 성능 향상은 없었습니다)

C. 아쉬웠던 점들

- 우리만의 연구를 진행하기보다는 이미 있는 SOTA 모델과 관련된 연구를 따라가는 것이 조금 아쉽게 느껴졌습니다. 좀 더 자율적인 테스트를 진행하면 좋겠다는 의견이 나왔습니다.
- 코드리뷰를 진행했지만 팀원 간 코드에 대해 이해하지 못한 부분이 있던 점에서 바람직한 코드리뷰가 진행되지 않았다는 생각이 들었습니다.
- 코드의 흐름에 대해 전반적인 테스트를 진행하지 않고 코드를 작성한 경향이 있었습니다.
- 팀원 모두 계획한 대로 테스트를 진행하기 위해서 데드라인을 지정하여 관리해야겠다는 생각을 하였습니다.
- 데이터적인 측면으로 많이 테스트를 진행하지 못한 부분에 대해 아쉬움이 있었습니다.

D. 프로젝트를 통해 배운 점 또는 시사점

- Git Flow 방법론과 코드 리뷰
- 데이터 라벨링에 많은 시간이 요구됨을 확인하였고, 이후 대회에서 좀더 여유를 가지고 진행해야 함을 느꼈습니다.
- huggingface Trainer, Transformers 코드 흐름에 대해서 다시 한번 확인해볼 수 있었습니다.
- confusion matrix를 통한 예측 결과를 바탕으로 모델을 평가하는 방법을 경험하였습니다.
- 양상을 기법중 하나인 Voting 방식에 대해서 배웠고, 언제 양상을 기법을 사용해야 하는지에 대해 배웠습니다.
- private 점수가 많이 하락한 것을 보고 일반화 성능의 중요성을 다시 한번 인식했습니다.

Wrap-up Report

Boostcamp AI Tech NLP-01 황지원(T5231)

A. Overview of the Project

- 관계 추출 (RE : Relation Extraction)란 문장의 단어(Entity)에 대한 속성과 관계를 예측하는 문제이며, 비구조적인 자연어 문장에서 구조적인 triple을 추출해 정보를 요약하고, 중요한 성분을 핵심적으로 파악할 수 있습니다. 결과물로 Relation 30개 중 하나를 예측한 **pred_label**로 평가합니다.
- 평가지수로는 **micro F1 score**를 사용하여 평가를 진행합니다.

B. Goals of the Project

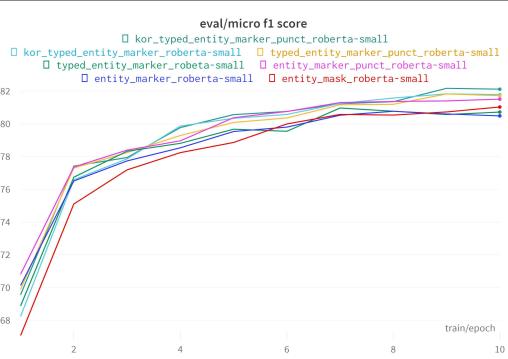
이번 대회는 크게 3가지 키워드를 통해 목표를 이루고자 하였습니다.
첫 번째로 **PM으로서의 경험**입니다. 프로젝트 매니저(PM)를 통해 전반적인 대회 프로젝트를 계획하고 설계하여 최상의 모델을 얻고자 하였습니다. 또한 팀원들과의 피드백을 통해 공통의 목표를 가지고 실험 및 연구를 진행할 수 있도록 도움을 주고자 하였습니다.
두 번째로 **협업 경험**입니다. 지난 대회에 이어서 Git Flow 및 브랜치와 커밋, Pull Request 규칙들에 대해서 정하고 이를 바탕으로 프로젝트를 진행하며 그와 함께 적극적인 코드 리뷰를 진행하는 등 팀원 간 최상의 결과를 얻기 위한 코드 작업/관리를 진행하고자 하였습니다.
마지막으로 **논문에 기반을 둔 모델링 작업**입니다. RE Task의 경우 많은 참고 자료가 많은 만큼 이를 참고하여 SOTA 모델과 임베딩, 라벨링 방식에 대해 연구를 진행하고자 하였습니다.

C. Issues of the Project

1. Git 팀 규칙 설정 (Setting Team Git Rule)

상황	이전 대회에 이어 팀 프로젝트를 시작하기 전에 전반적인 Git Rule을 설정하여 효율적인 업무 수행을 하고자 하였다.
과정 및 결과	<ol style="list-style-type: none">Git Flow 설정 : 기존의 Git Flow 방법론을 베이스로 하되 main, development, fix, feature로 브랜치를 사용해서 작업 하도록 규칙을 정함Git Branch Name Rule 설정 : 각 브랜치마다 어떤 작업을 하는지 확인할 수 있도록 feature, fix와 같은 작업을 포함한 브랜치 명으로 설정하여 작업을 진행할 수 있도록 설정PR Template 설정 : 작업에 대한 내용을 공유할 때, 일정한 템플릿을 통해 효율적인 업무 수행을 할 수 있도록 설정코드 리뷰 : PR을 진행하면서 각 Merge할 코드에 대해 팀원간 코드 리뷰를 통해 더 좋은 코드를 만들 수 있도록 노력하였습니다.

2. 엔티티 마커 전처리 (Entity Marker Representation)

상황	Entity의 관계를 더 잘 구분하지 못하는 것 같은 결과를 확인하였다.																								
가설	Sentence-level RE를 참고하여 Entity에 Special Marker를 추가하면 성능이 개선될 것이다.																								
과정 및 결과	<ul style="list-style-type: none">- 데이터 전처리 단계에서 Train 데이터셋을 Subject/Object Entity를 확인하여 Marker로 수정하고 이를 input으로 모델에 학습을 시키고자 하였습니다.- Special Token Method<ul style="list-style-type: none">- Entity mask, Entity marker, Entity marker (punct), Typed entity marker, Typed entity marker (punct), Kor Typed entity marker, Kor Typed entity marker (punct)- KLUE Dataset을 사용해 모델 학습을 진행하였고, 한국어로 이루어진 데이터 셋인 만큼 한국어로 Marker를 사용한 것이 좋은 점수를 보여주었습니다. 위 결과를 토대로 데이터의 전처리를 진행해 모델의 성능을 더욱 높이고자 합니다.																								
참고 자료	<table border="1"><thead><tr><th>Rank</th><th>Method</th><th>Score</th></tr></thead><tbody><tr><td>1</td><td>kor_typed_entity_marker_punct</td><td>82.121</td></tr><tr><td>2</td><td>kor_typed_entity_marker</td><td>81.789</td></tr><tr><td>3</td><td>typed_entity_marker_punct</td><td>81.719</td></tr><tr><td>4</td><td>entity_marker_punct</td><td>81.511</td></tr><tr><td>5</td><td>entity_mask</td><td>81.031</td></tr><tr><td>6</td><td>typed_entity_marker</td><td>80.723</td></tr><tr><td>7</td><td>entity_marker</td><td>80.491</td></tr></tbody></table> 	Rank	Method	Score	1	kor_typed_entity_marker_punct	82.121	2	kor_typed_entity_marker	81.789	3	typed_entity_marker_punct	81.719	4	entity_marker_punct	81.511	5	entity_mask	81.031	6	typed_entity_marker	80.723	7	entity_marker	80.491
Rank	Method	Score																							
1	kor_typed_entity_marker_punct	82.121																							
2	kor_typed_entity_marker	81.789																							
3	typed_entity_marker_punct	81.719																							
4	entity_marker_punct	81.511																							
5	entity_mask	81.031																							
6	typed_entity_marker	80.723																							
7	entity_marker	80.491																							

3. Hidden-layer Concatenation (은닉층 정규화)

상황	R-BERT 논문을 참고하여 Hidden-layer의 정규화를 진행하고자 하였다.
가설	각 Entity에 대한 Hidden-layer를 평균 내어 Layer를 거쳐 학습을 진행하면 정규화를 통해 성능 향상이 이루어질 것이다.

과정 및 결과	<ul style="list-style-type: none"> Custom Model로 하여금 PTM을 Fine-tuning하여 hidden state output에 추가적인 Task를 진행하였습니다. Subject Entity Token과 Object Entity Token에 대한 Hidden State을 평균내어 계산하고 각각 tanh activation function과 fully connected layer로 output을 얻음 h1(Subject Entity Token)과 h2(Object Entity Token)에 대해서는 같은 가중치 공유하여 사용하게 됨 이를 다시 각각을 Concatenation + softmax layer를 거쳐 output을 확보함 테스트를 진행해봤을 때 실제 논문과 유사한 결과는 확인할 수 없었지만, 일반화가 진행되어 성능 향상이 이뤄질 수 있었다는 결과를 확인할 수 있었다.
참고 자료	<p>The figure consists of three line charts side-by-side, each with 'train/global_step' on the x-axis (ranging from 1.2k to 4k) and a vertical dashed line at approximately 2.1k steps.</p> <ul style="list-style-type: none"> eval/loss: The y-axis ranges from 0.34 to 0.42. The green line (small-apply-hidden) starts at ~0.38 and rises to ~0.41. The blue line (small-test-ep10) starts at ~0.34 and rises to ~0.40. The purple line (small-test-ep10) starts at ~0.36 and rises to ~0.42. eval/accuracy: The y-axis ranges from 0.792 to 0.802. The green line (small-apply-hidden) starts at ~0.796 and rises to ~0.801. The blue line (small-test-ep10) starts at ~0.792 and rises to ~0.794. The purple line (small-test-ep10) starts at ~0.794 and rises to ~0.798. eval/micro f1 score: The y-axis ranges from 76 to 82. The green line (small-apply-hidden) starts at ~78.5 and rises to ~81. The blue line (small-test-ep10) starts at ~77.5 and rises to ~80. The purple line (small-test-ep10) starts at ~76.5 and rises to ~80.5.

4. Num-labels 내용 수정

상황	30개의 label을 분류하기 이전에 미리 크게 분류를 진행하고 세부적인 Label로 식별하는 모델을 구성하면 좀 더 성능이 오르지 않을까 고민
가설	데이터 불균형을 가지고 있는 학습 데이터에 대해 no_relation과 아닌 경우를 1차적으로 나누고 세부적으로 학습을 거치면 성능 향상이 이루어질 것이다.
과정 및 결과	<ul style="list-style-type: none"> 데이터 전처리 과정에서 label을 확인하고 no_relation에 해당하는 경우와 아닌 경우를 다시 라벨링을 진행 no_relation과 아닌 경우를 처음으로 no_relation/Person/Organization의 3가지 Label을 구분하는 모델 학습 진행 처음에 생각했을 때에는 1차적으로 분류를 하려면 85~90% 정도의 정확도는 나와야한다고 생각했는데 그것보다 높지 않은 결과를 확인하였습니다. 이러한 결과를 바탕으로 해당 Task는 모델에 적용할 수 없을 것이라 판단하였습니다.

D. Evaluation of the Project

- Qualifications of the Project (PM, Git 코드 버전 관리, 모델링 및 임베딩 작업 진행)

- 프로젝트 진행 방향 설계 및 모델링 전략 수립
- 모델링 코드 구현 및 공유, 코드 버전 관리 (Github)
- 모델 탐색 및 테스트 진행 (R-BERT)
- 모델 학습 전략 수립 (Hidden-layer Concatenation, Entity Marker Representation 적용)

만족스러웠던 점	<ul style="list-style-type: none"> - PM으로서 처음 프로젝트를 진행하는데, 팀원 각각의 진행 속도와 방향성에 맞게 계획을 조정하고 설계하는 경험 이 이후 프로젝트 계획을 세우고 진행하는데 많은 도움을 줄 것이라 생각합니다. - 이전보다 Git Flow 방법론에 맞게 더 효율적으로 팀 프로젝트를 진행할 수 있었습니다. - 적극적인 코드리뷰를 권장하였고, 이를 바탕으로 좀 더 좋은 퀄리티의 코드를 작성할 수 있었다고 생각합니다. - 모델 및 학습 방법에 대한 서칭(Searching)을 진행하였고, 이를 바탕으로 최상의 결과를 얻기 위한 연구를 진행할 수 있었습니다. - 매일 프로젝트 진행상황을 공유하면서 서로 피드백을 하고 다음 공동의 목표를 설계하는 일련의 과정이 매우 값진 경험이었다고 생각합니다.
아쉬웠던 점	<ul style="list-style-type: none"> - 데이터를 이번 프로젝트 때 많이 확인해보지 못했는데, 다음 대회에는 좀 더 집중해서 볼 수 있도록 진행하고 싶습니다. - PM으로서 중간에 전반적인 프로젝트를 진행하는데 어려움을 주는 요소나 작업은 미리 확인을 했어야 했는데, 이를 고려하지 못했던 점이 아쉽게 느껴졌습니다. - Custom Model을 설계하면서 Transformers 라이브러리에 대해 이해하지 못한 부분이 있어 중간에 포기도 했었는데, 끝까지 디버깅을 통해 코드를 이해하려고 하지 않았음이 아쉽게 느껴졌습니다. - 전체적인 베이스 코드를 작성했으면 했지만 해당 작업을 빠르게 진행하지 못해 모델 학습과 그 결과를 분석하는데 지장이 있던 부분이 아쉬웠습니다.

E. Conclusion

대회를 성공적으로 마무리 지을 수 있었습니다. 성적으로 전반적인 결과를 보자면 이전 대회 때에는 13등이라는 하위권 성적을 거뒀던 것에 비해 이번 대회에서는 9등이라는 성적으로 전보다 발전된 결과가 있었음을 알 수 있었습니다.

이번 대회는 모델링 작업을 맡아 프로젝트를 진행했습니다. 이전 대회에서 아쉬웠던 부분이었던 논문을 바탕으로 SOTA 모델을 연구하고 테스트하는 경험을 이번에는 충분히 진행할 수 있었던거 같습니다. 물론 모든 결과를 확인할 수 있었던 것은 아니었지만 테스트한 결과를 팀원간 공유하고 피드백을 하면서 최상의 모델을 얻기 위해서 노력했던거 같습니다.

하지만 이번 대회는 모델링 작업보다도 최종 프로젝트에 대비해 협업을 좀 더 중시하고 팀원들에게 더 피드백 하는 시간을 더 가졌던 거 같습니다. 각자 가설을 세우고 서칭(Searching)을 진행하며 SOTA 모델로서 연구를 진행하는 것도 물론 중요한 요인 중 하나지만 6명이라는 팀원이 하나의 공동의 목표를 가지고 프로젝트를 임하는 만큼 최고의 결과를 얻기 위해서는 협업이 우선시 되어야한다는 생각을 가졌던 거 같습니다. 그래서 PM이라는 역할로 이번 프로젝트에 임하면서 개인의 연구와 테스트에 집중하였지만 틈틈이 코드리뷰, Branch 관리 등 협업 부분에서도 많은 관심을 기울였던 거 같습니다.

대회는 중위권의 성적을 거뒀지만 이보다 값진 경험을 팀원과 함께 했다고 생각하고 위 경험을 발판삼아 더 좋은 모델링으로 최고의 결과를 얻고, 더 체계적인 프로젝트를 진행할 수 있도록 노력할 생각입니다.