



# KLUE\_NLP\_팀 리포트(10조)

## ○ 프로젝트 개요

### • 프로젝트 주제

문장, 단어에 대한 정보를 통해 ,문장 속에서 단어 사이의 관계를 추론하는 Task

### • 프로젝트 개요

#### ○ 프로젝트 목표

비구조적인 자연어 문장에서 구조적인 triple을 추출해 정보를 요약하고, 중요한 성분(관계 추출:Relation Extraction)을 파악하는 것.

#### ○ 구현 내용

##### ▪ EDA

- 데이터 특성 파악을 위한 EDA
- 중복 및 오류 데이터 검출 및 수정 작업
- Backtranslation을 통한 데이터 augmentation 시도

##### ▪ Data

- 학습의 효율을 높이기 위해 논문(An Improved Baseline for Sentence-level Relation Extraction)의 내용을 적용하여 input 데이터의 Entity Representation에 Typed entity marker (punct) 적용

##### ▪ Model

- "klue/roberta-large" 모델과 GRU, LSTM layer를 연결한 custom 모델을 생성
- 생성 모델을 중 상위 모델을 이용하여 soft ensemble 구현

#### ○ 교육 내용의 응용

요약된 정보를 사용해 QA 시스템 구축 및 효율적인 시스템 및 서비스 구성이 가능

### • 활용 장비 및 재료(개발 환경, 협업 tool 등)

- 서버환경 : Ubuntu 18.04.5 LTS , GPUv100
- 개발툴 : vscode, jupyter notebook
- 협업툴 : Git, Github Project Slack, Zoom

- 프로젝트 File tree 및 Workflow

## ○ 프로젝트 팀 구성 및 역할

- 김남현(T3021) : EDA, 데이터 전처리, 데이터 교정을 위한 dataframe editing
- 민원식(T3079) : 데이터 분석(EDA) , 모델 성능 향상을 위한 튜닝 시도, 하이퍼 파라미터 튜닝을 위한 wandb 모니터링 툴을 이용.
- 전태양(T3194) : EDA, 성능 향상을 위한 모델 개선 강구
- 정기원(T3195) : EDA, 전처리, Data Augmentation, Baseline 코드 수정, 팀원 코드 병합
- 주정호(T3211) : 데이터 전처리, 모델 구현, 앙상블 구현
- 최지민(T3223) : EDA, Data Augmentation

## ○ 프로젝트 수행 절차 및 방법

### • 프로젝트 개발 Process

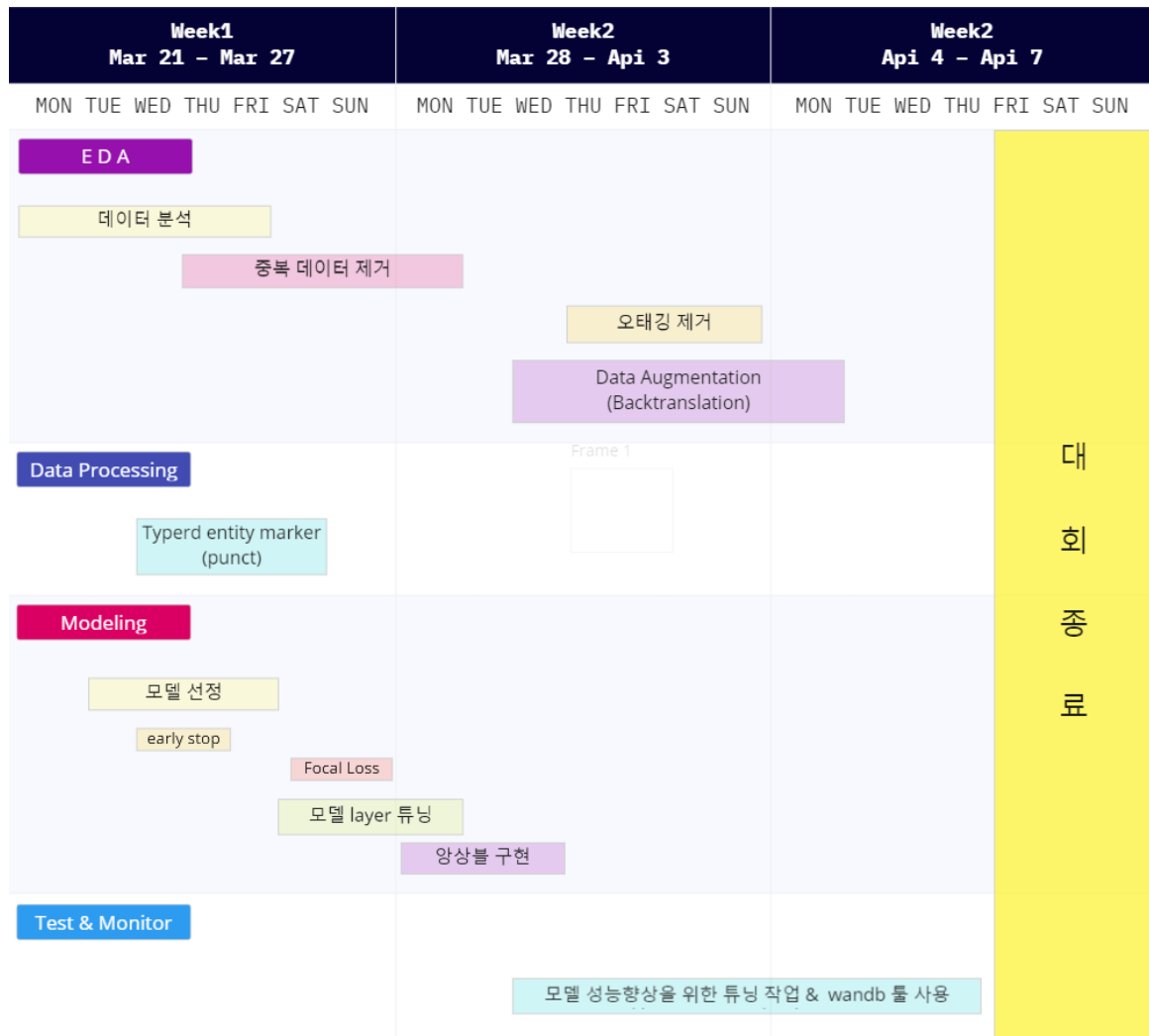
개발 과정을 아래와 같이 크게 5가지 파트로 분류함.

- EDA : Jupyter Notebook을 이용하여 데이터 특성 및 이상치 분석
- Data Processing : 모델 학습에 유용한 형태로 데이터를 처리
- Modeling : 모델을 구현하고 성능 향상을 위해 Parameter Tunning 및 다양한 기능 추가
- Model Test & Monitor : Monitoring Tool을 이용하여 모델을 다양한 환경에서 테스트
- 협업 Tool 관리 및 기타(문서 정리) : Git Flow 적용

### • 프로젝트 역할분담

모든 Process를 경험하고 싶다는 팀원들의 의견에 따라 팀원 별로 파트를 나누지 않고 모든 파트에 모든 팀원이 언제든지 참여할 수 있도록 자유롭게 진행

### • 프로젝트 수행 및 완료 과정(Work Breakdown Structure)

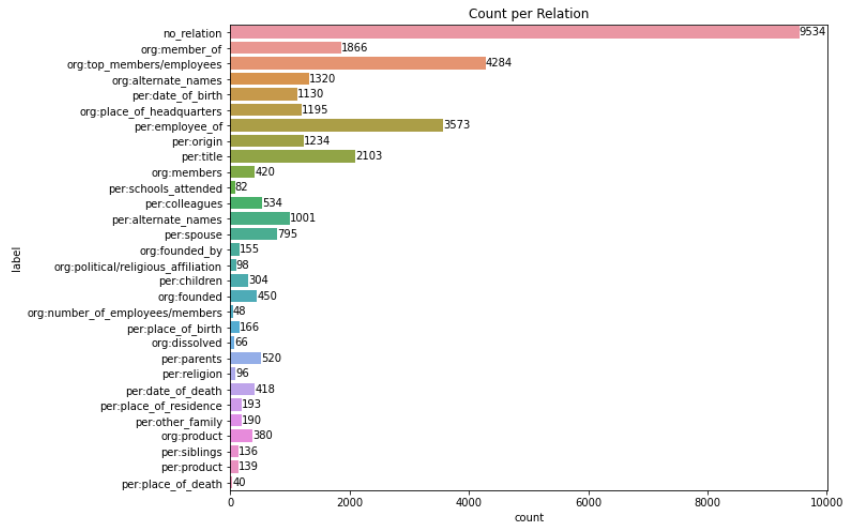


대  
의  
중  
요

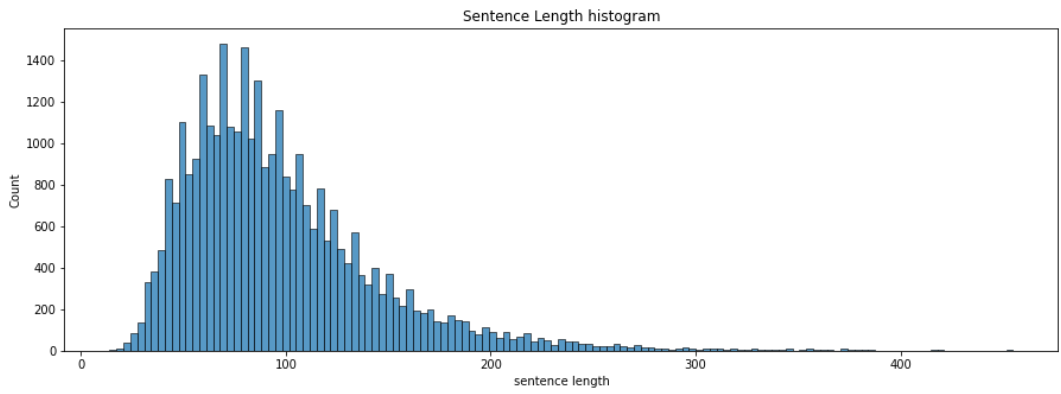
## ○ 프로젝트 수행 결과

### 1. EDA 및 전처리 (학습 데이터 소개)

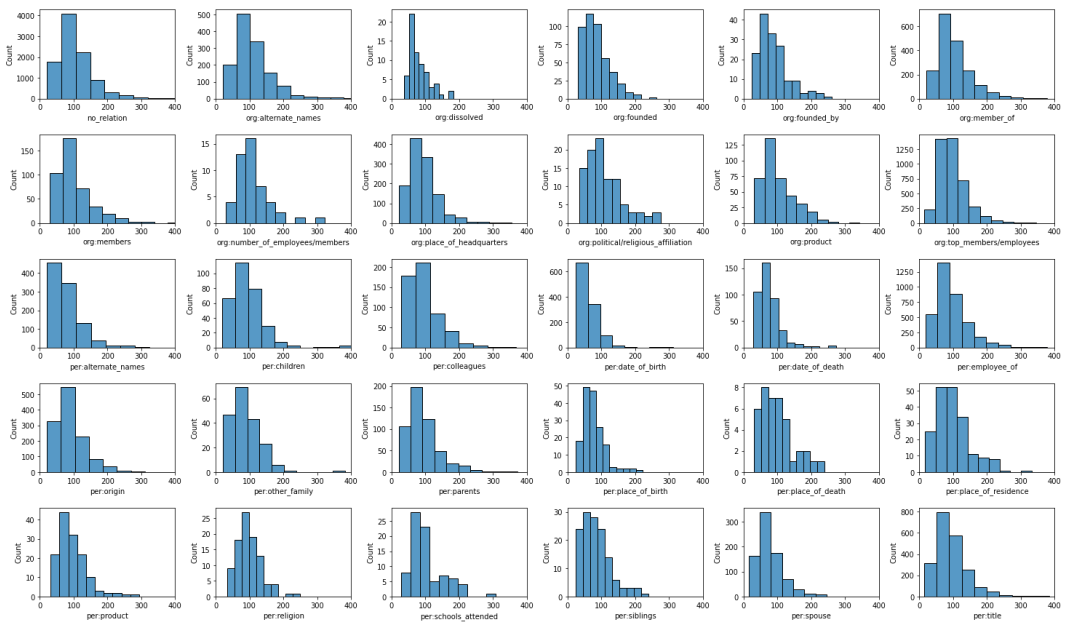
- EDA
  - label



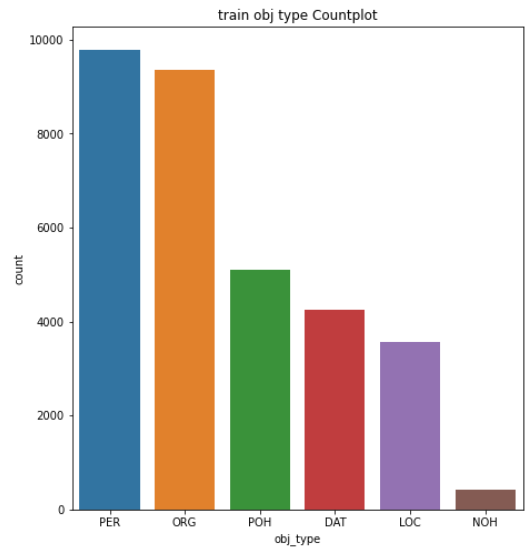
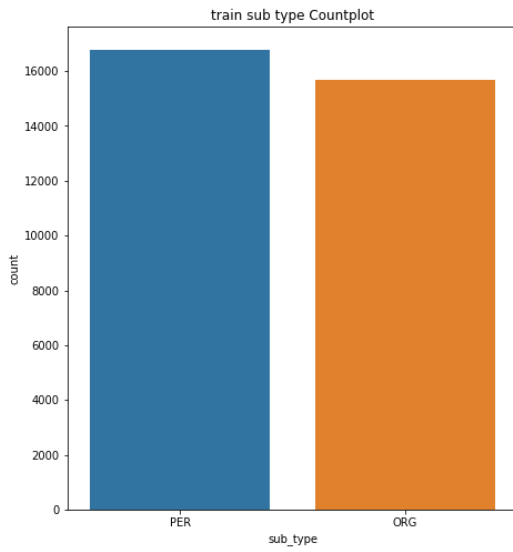
○ 문장 분포 확인



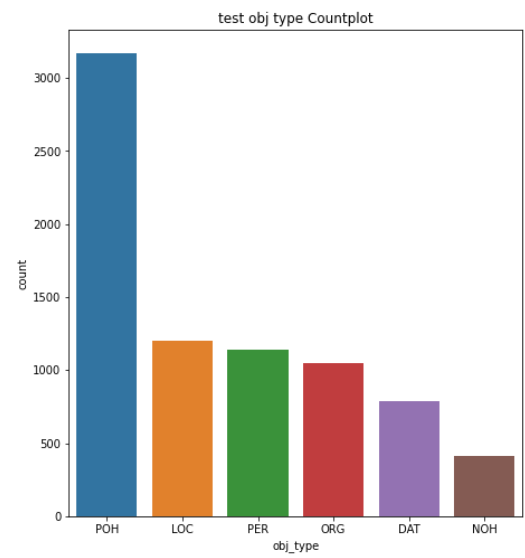
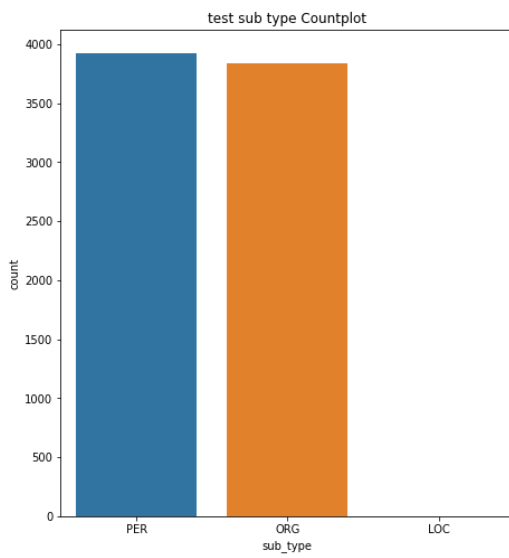
○ label별 문장 분포 확인



- 학습 데이터의 Subject, Object 단어의 type 분포 확인



- 테스트 데이터의 Subject, Object 단어의 type 분포 확인



- 학습 데이터 : id-subj\_info-obj\_info-sen-label-label\_num 순서로 구조 재구성

id	subj_type	subj_word	subj_start	subj_end	obj_type	obj_word	obj_start	obj_end	sentence	label	label_num
29086	ORG	한국남동발전	0	5	PER	유광열	10	12	한국남동발전(사장 유광열)은 22일 전주시 수곡면에서 농촌을 기반으로 하는 지속적인 일자리 생태계 조성을 위한 'KOEN 농촌복지센터(사관그림소) 개소식을 가졌다.	org.top_members/employees	1
16251	ORG	한국국제협력단	87	93	PER	이미경	106	108	한국건강관리협회(회장 채종일, 이하 '건협')와 국제구호개발 NGO 사단법인 코네이비스 인터내셔널(이사장 이일하, 이하 '코네이비스')은 지난 3월 13일 한국국제협력단(KOICA) 이사장 이미경과 "탄자니아 고매성 지역사회 보건환경개선을 통한 초중등생 건강증진 사업수행에 위한 약정을 체결했다.	org.top_members/employees	1
20763	ORG	전주고구	125	128	LOC	중앙성당	93	96	1800년 5.18 광주 민주화 운동 당시 구제 전은사에 머물면서 광주에서 탈출한 사람들의 이야기를 정리해 <전두환 광주살육직전>이라는 유인물을 만들어 전주고구 전주 중앙성당을 맡고 있던 문정현 신부를 찾아왔으며, 문정현은 전주고구청 윤진기로 1만장을 복사해서 부산 대구 서울로 보내고 미사 전후로 유인물을 주보에 붙여서 나눠주었다.	org.members	2
1965	PER	마사오원	24	27	POH	아버지	20	22	프랑스 파리에서 지휘자이자 작곡가인 아버지 마사오원(馬孝勳)과 성악가인 어머니 마리나 루(盧雅文) 사이에서 태어났다.	no_relation	0
13188	PER	유재국	11	13	POH	2001년 아마주어 자유계약	37	51	덕수고등학교 출신인 유재국은 졸업 후 국내 프로야구를 가지지 않고 2001년 아마주어 자유계약으로 시카고컵스에 입단했다.	no_relation	0
27748	PER	박진영	13	15	ORG	JYP	0	2	JYP엔터테인먼트 대표 박진영 씨가 '태양을 피어는 방법' 작곡에 원한 뒷배기를 공개했다.	per.employee_of	6
14089	ORG	연산 OK저축은행 리얼캐시	29	43	DAT	2015	5	8	2014-2015 시즌 정규리그 우승을 차지했으나, 연산 OK저축은행 리시(현재에게 3배를 당한 후 2015년 5월에 대한 상영 불우영상 프로젝트위원의 결정으로 선임되어 감독직에서 물러났으며,채일기와 스포츠구단 운영담당 임원(부사장)을 겸직했다.	no_relation	0
14047	ORG	KB증권	20	23	ORG	KB금융그룹	62	67	신대증권과 KB증권 합병 이후 통합 KB증권에서 경영관리부는 부서장직을 2018년까지 역임했으며 올해 1월부터 KB금융그룹 계열사인 KB저축은행에서 영업총괄 부서장으로 재직했다.	no_relation	0
13995	PER	펠라페 4세	50	55	PER	마르가리타 데레사	0	8	마르가리타 데레사는 태어난 지 얼마 되지 않아 외숙부 레오폴트 1세와의 결혼이 결정되었고 펠라페 4세는 궁정 화가인 벨라스케스에게 오스트리아 궁정에 보낼 딸의 초상화를 그려주도록 명했다.	per.children	10

- 중복 데이터 : 84개의 중복 데이터 확인 후 제거
- 오타깅 데이터 제거 : 5개의 오타깅 데이터 확인 후 제거
- 데이터 교정 : 학습 데이터 중 `subj_type`, `obj_type`, `label` 이 잘못된 데이터를 교정 → 오히려 성능이 떨어져 원래의 학습 데이터를 사용
- Easy Data Augmentation : KoEDA 라이브러리를 사용하여 Random Insertion, Random Deletion, Random Swap, Synonym Replacement 적용 → 성능 개선 효과 없었음
  - 논문 "1901.11196.pdf (arxiv.org)" 참고

• Preprocess

- Typed Entity marker(punct)
  - 논문 "An Improved Baseline for Sentence-level Relation Extraction" 참고

Method	Input Example	BERT <sub>BASE</sub>	BERT <sub>LARGE</sub>	RoBERTa <sub>LARGE</sub>
Entity mask	[SUBJ-PERSON] was born in [OBJ-CITY].	69.6	70.6	60.9
Entity marker	[E1] Bill [/E1] was born in [E2] Seattle [/E2].	68.4	69.7	70.7
Entity marker (punct)	@ Bill @ was born in # Seattle #.	68.7	69.8	71.4
Typed entity marker	(S:PERSON) Bill (/S:PERSON) was born in (O:CITY) Seattle (/O:CITY).	71.5	72.9	71.0
Typed entity marker (punct)	@ * person * Bill @ was born in # ^ city ^ Seattle #.	70.9	72.7	74.6

원본 : <Something> 는 조지 해리슨이 쓰고 비틀즈가 1969년 앨범 《Abbey Road》에 담은 노래다.

→ Typed Entity marker(punct) : <Something> 는 # ^ [PER] ^ 조지 해리슨 # 이 쓰고 @ \* [PER] \* 비틀즈 @ 가 1969년 앨범 《Abbey Road》에 담은 노래다.

⇒ [CLS] <Something> 는 # ^ [PER] ^ 조지 해리슨 # 이 쓰고 @ \* [PER] \* 비틀즈 @ 가 1969년 앨범 《Abbey Road》에 담은 노래다. [SEP]

- Typed Entity marker(punct) + Query
  - 논문 "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" 참고



⇒ 기존 BERT의 Pretrain 방식과 유사한 input으로 만들어줌

원본 : 〈Something〉 는 조지 해리슨이 쓰고 비틀즈가 1969년 앨범 《Abbey Road》 에 담은 노래다.

→ Typed Entity marker(punct) : 〈Something〉 는 # ^ [PER] ^ 조지 해리슨 # 이 쓰고 @ \* [PER] \* 비틀즈 @ 가 1969년 앨범 《Abbey Road》 에 담은 노래다.

→ Query : @ \* [PER] \* 비틀즈 @ 와 # ^ [PER] ^ 조지 해리슨 # 의 관계

⇒ [CLS] @ \* [PER] \* 비틀즈 @ 와 # ^ [PER] ^ 조지 해리슨 # 의 관계 [SEP] 〈Something〉 는 # ^ [PER] ^ 조지 해리슨 # 이 쓰고 @ \* [PER] \* 비틀즈 @ 가 1969년 앨범 《Abbey Road》 에 담은 노래다. [SEP]

○ Standard with Entity Location Token

- 논문 “엔티티 위치 정보를 활용한 한국어 관계 추출 모델 비교 및 분석” 참고



원본 : 〈Something〉 는 조지 해리슨이 쓰고 비틀즈가 1969년 앨범 《Abbey Road》 에 담은 노래다.

→ Standard with Entity Location Token : 〈Something〉 는 [OBJ] 조지 해리슨 [/OBJ] 이 쓰고 [SUB] 비틀즈 [/SUB] 가 1969년 앨범 《Abbey Road》 에 담은 노래다.

⇒ [CLS] 〈Something〉 는 [OBJ] 조지 해리슨 [/OBJ] 이 쓰고 [SUB] 비틀즈 [/SUB] 가 1969년 앨범 《Abbey Road》 에 담은 노래다. [SEP]

○ Backtranslation : Selenium을 활용한 크롤링을 통해 한국어 → 영어 → 한국어 번역

원본 : 〈Something〉 는 조지 해리슨이 쓰고 비틀즈가 1969년 앨범 《Abbey Road》 에 담은 노래다.

→ "Something" is a song written by George Harrison and included by the Beatles on their 1969 album Abbey Road.

⇒ "Something"은 조지 해리슨이 작곡하고 비틀즈가 1969년 앨범 Abbey Road에 포함시킨 노래입니다.

○ 성능 비교(micro f1)

(AutoModelForSequenceClassification.from\_pretrained("klue/roberta-large"))

- Typed Entity marker(punct) : 71%
- **Typed Entity marker(punct) + Query : 73%**
- Standard with Entity Location Token : 70%
- Backtranslation : 72% → 생성 문장을 살펴보면 저품질 문장이 많음

## 2. 모델 개요

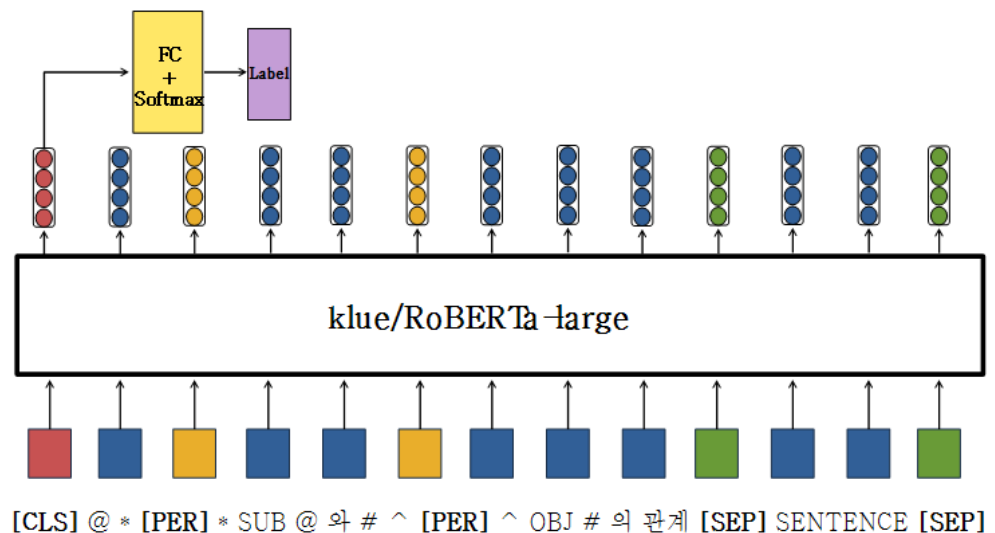
• Model

○ Pretrained Model

- klue/bert-base
- **klue/roberta-large**

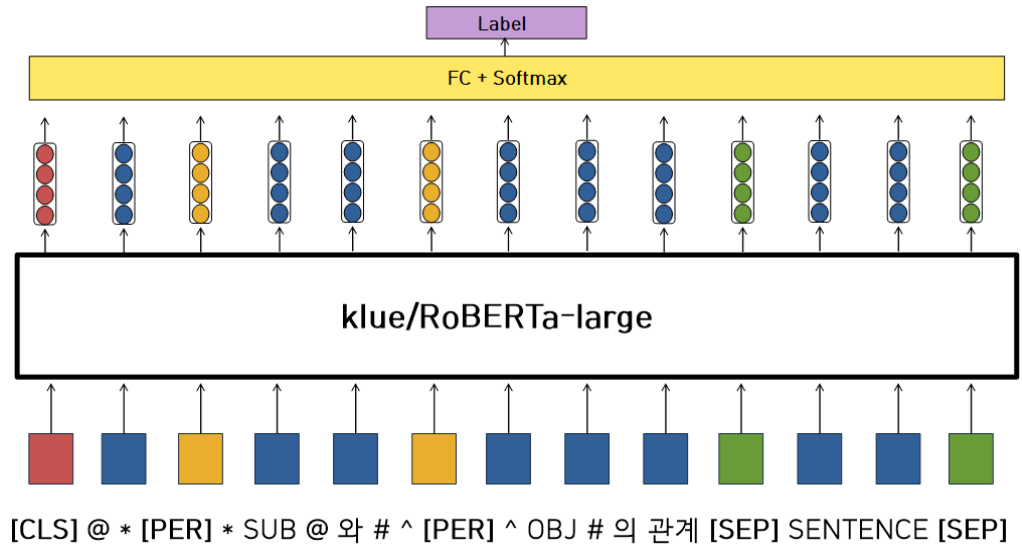
○ Additional Layer

- AutoModelForSequenceClassification

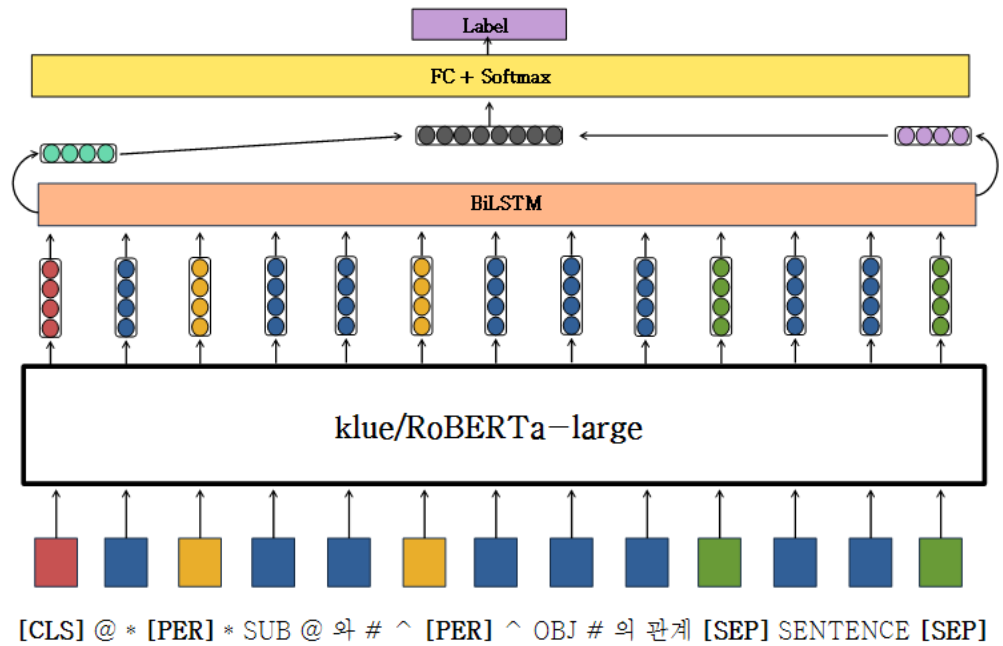


- FC

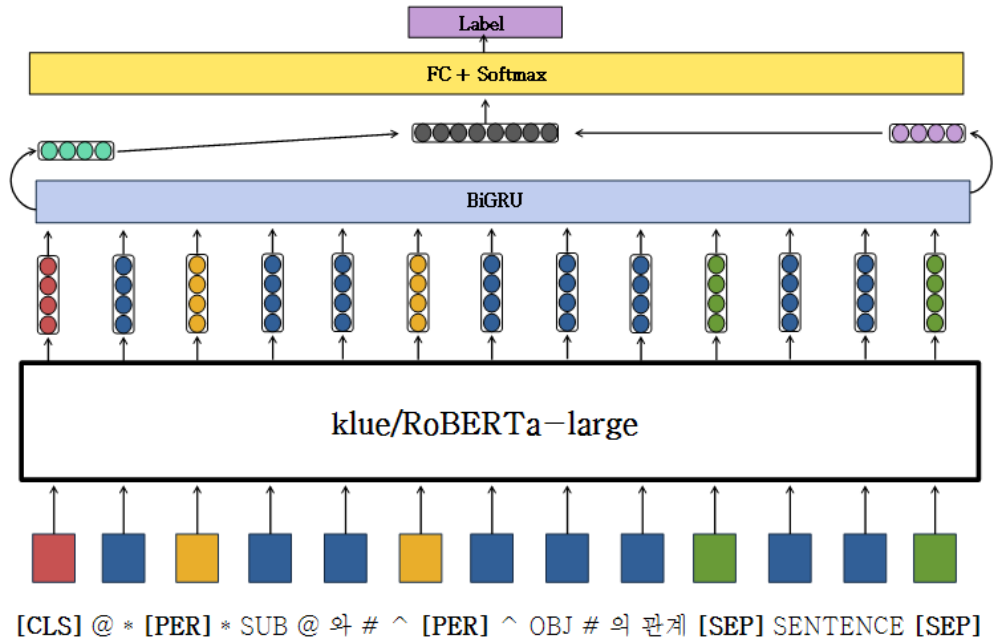




- BiLSTM

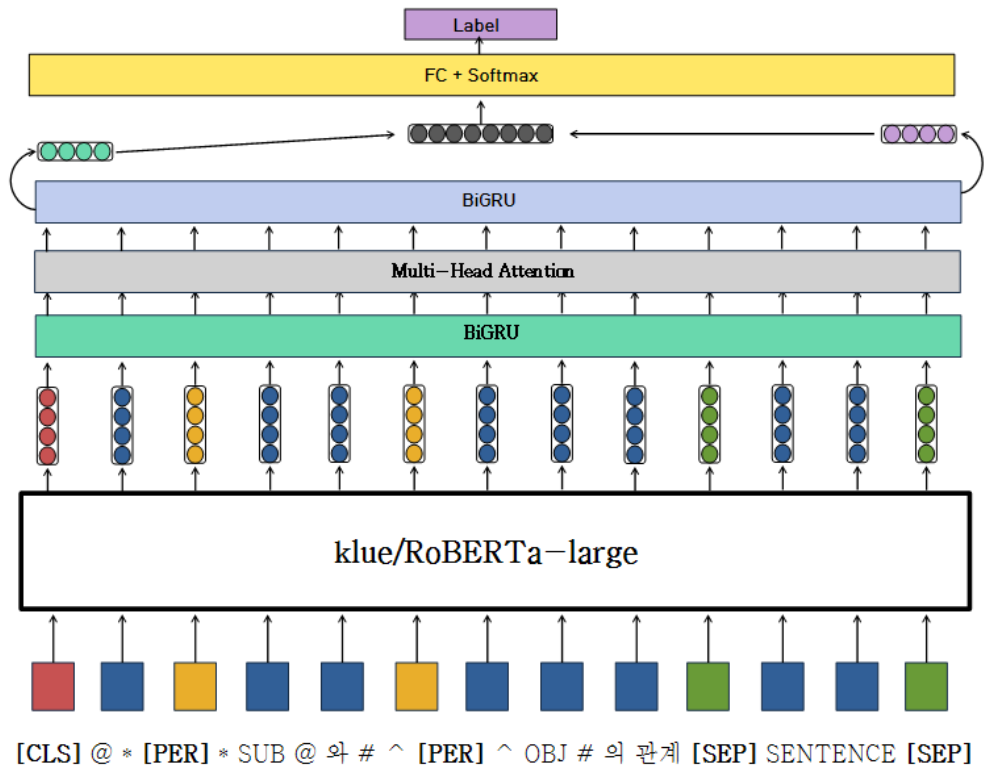


- BiGRU



■ BiGRU + Multi-Head-Attention + BiGRU

- 논문 "UO UP V2 at Haha 2019: BiGRU Neural Network Informed with Linguistic Features for Humor Recognition" 참고



- 성능 비교(micro f1)  
(Typed Entity marker(punct) + Query)
  - AutoModelForSequenceClassification : 74.9%
  - FC : 74.3%
  - **BiLSTM : 75.6%**
  - BiGRU : 75.1%
  - BiGRU + Multi-Head-Attention + BiGRU : 74.4%

- Hyper Parameter

- Learning Rate
  - **3e - 5**
  - **warmup\_ratio : 0.1**
  - **decay to 0**
  - **weight decay : 0.01** → overfitting 방지

→ 논문 “An Improved Baseline for Sentence-level Relation Extraction” 참고



- Batch Size
  - **64**
  - 32

→ **Out Of Memory**가 일어나지 않는 선에서 **Batch Size**는 클수록 성능이 좋았음
- max\_len : 입력 Sentence의 최대 길이
  - **160**
  - 256

- 성능은 비슷했지만, 256은 batch size를 64로 했을 때, Out Of Memory가 발생해 160을 사용
- Epoch
  - 5
  - 10
 → 똑같은 조건에서 Epoch가 10일 때, 성능이 더 떨어졌고, Overfitting이 발생했다고 판단
- Loss Function
  - **Cross Entropy** : Transformer의 Default
  - Focal Loss : Class Imbalance를 개선하지만 CE와 성능 차이가 없었음
  - **Label Smoothing : 0.1** → Class Imbalance 개선
- Optimizer
  - **AdamW** : Transformer의 Default
- Train
  - StratifiedKFold : 성능 개선 효과 없었음
- SOTA 모델
  - Preprocess : Typed Entity marker(punct) + Query
  - Model
    - Pretrained Model : "klue/roberta-large"
    - Additional Layer : BiLSTM
  - Hyper Parameter
    - Learning Rate
      - $3e - 5$
      - warmup\_ratio : 0.1
      - decay to 0
      - weight decay : 0.01
    - Batch Size : 64
    - max\_len : 160
    - Epoch : 5

- Loss Function
      - Cross Entropy
      - Label Smoothing : 0.1
    - Optimizer : AdamW
  - Ensemble
    - **Soft Voting**
      - 리더보드 상위 4개 모델 : **76.7338% (SOTA + 1%)**
      - 5개의 모델 : 75.5%
        - AutoModelForSequenceClassification : 74.9%
        - FC : 74.3%
        - BiLSTM : 75.6%
        - BiGRU : 75.1%
        - BiGRU + Multi-Head-Attention + BiGRU : 74.4%
      - 상위 3개의 모델 : 75.7%
        - AutoModelForSequenceClassification : 74.9%
        - BiLSTM : 75.6%
        - BiGRU : 75.1%


## ○ 자체 평가 의견

### • 프로젝트의 의도 및 달성 정도

- 프로젝트의 의도 : Sentence를 근거로 Subject 단어를 중심으로 Object 단어가 어떤 관계를 가지고 있는지 예측
- 달성 정도

#### ▪ Public


- micro\_f1 : 76.7383
- auprc : 82.4006

순위	팀 이름	팀 멤버	micro_f1 ↕	auprc ↕	제출 횟수	최종 제출
2 (-)	NLP_10조		76.7383	82.4006	93	2d

#### ▪ Private

- micro\_f1 : 74.8357

- auprc : 81.6977

순위	팀 이름	팀 멤버	micro_f1 ↓	auprc ↓	제출 횟수	최종 제출
3 (1 ↓)	NLP_10조		74.8357	81.6977	93	2d

- **계획 대비 달성도, 완성도 등**(자체적인 평가 의견과 느낀 점)
  - 대회 기간 내 리더보드 순위 1위였으나 최종 순위 3위로 마감함.
  - Data augmentation, Ensemble 등 원하던 기능을 모두 구현할 수 있었음.
- **잘한 점과 아쉬운 점**(팀 별 공통 의견 중심으로 작성하며, 2~3장 분량을 고려하여 개인적인 의견은 개인 회고 부분에서 작성할 수 있도록 합니다.)
  - **잘한 점들**
    - 분업을 통해 데이터 전수 조사 및 Backtranslation 등 대규모 작업을 성공적으로 완료함.
    - 대회 Data와 비슷한 TACRED 관련 논문을 다같이 읽어보고 발표하는 시간을 가짐.
  - **시도 했으나 잘 되지 않았던 것들**
    - 데이터 태깅 오류 수정 작업을 하였으나 성능 향상이 없었음.
    - Data augmentation 작업을 하였으나 성능 향상이 없었음.
  - **아쉬웠던 점들**
    - 새로 작성한 코드에 주석을 다 달았더라면 팀원들이 더 빠르고 쉽게 이해할 수 있었을 거라는 생각이 들어서 다음 대회 때 이 부분을 개선해야겠다.
  - **프로젝트를 통해 배운 점 또는 시사점**
    - 평가 데이터도 완벽하지 않을 수 있다는 점과 그럼에도 불구하고 기타 다른 방법으로 성능을 향상 시킬 수 있다는 점을 알 수 있었다.

개인회고록(민원식)

개인회고록(김남현)

개인회고록(전태양)

개인회고록(정기원)

개인회고록(최지민)

개인회고록(주정호)

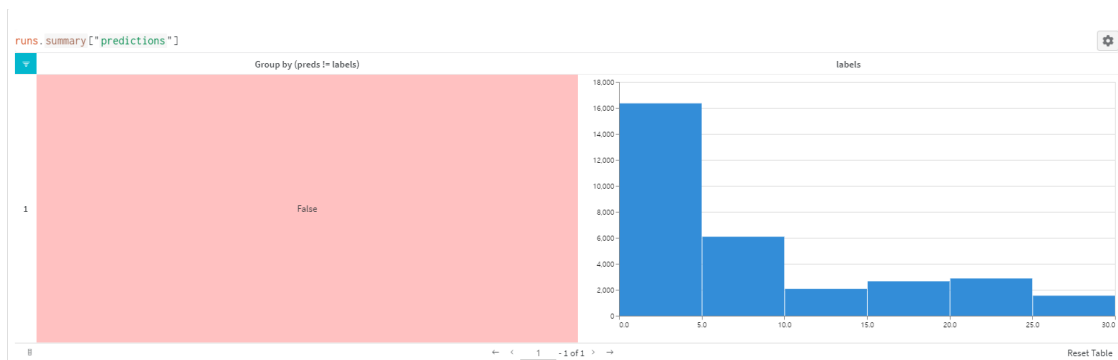
# 개인회고록(민원식)

○ 이번 프로젝트에서 나의 목표는 무엇이었는가?

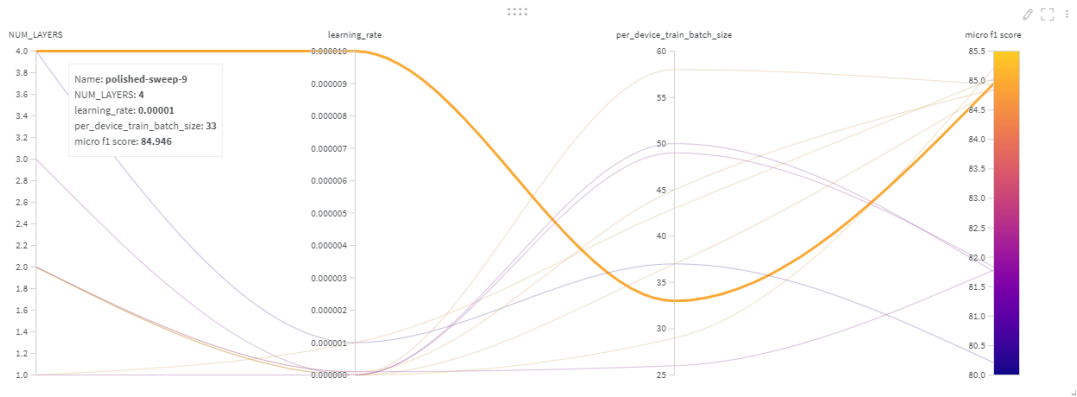
- pandas, matplotlib 라이브러리를 최대한 이용하여 EDA 해볼 것
- huggingface 라이브러리의 다양한 기능을 활용해 보는 것
- wandb 툴을 이용하여 모델 실험결과를 모니터링 및 하이퍼파라미터 튜닝하기
- 팀원간 최대의 시너지를 발휘하기 위해 원할히 협업하기

○ 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

- pandas, matplotlib 라이브러리를 이용하여
  - 데이터 내 중복 문장과 중복 문장 중 entity까지 완전히 동일한 중복 데이터 필터링 수행
  - 라벨별 기준으로 (subject\_entity, object\_entity)의 분포를 확인하고 entity 태깅 오류 파악
- huggingface 라이브러리를 이용하여
  - TrainingArguments의 다양한 parameter(warmup\_ratio, weight\_decay, label\_smoothing\_factor, fp16, report\_to)를 이용하여 모델 학습 및 실험결과를 wandb와 연동작업
  - Trainer의 callbacks parameter를 통한 earlystopping 기능 활용 및 세부적인 훈련과정에서 custom 으로 추가 작업이 가능함을 이해함
- wandb 라이브러리를 이용하여
  - 실험결과에 대한 분석



○ Sweep 기능을 이용하여 다양한 hyperparameter에 대한 튜닝시도



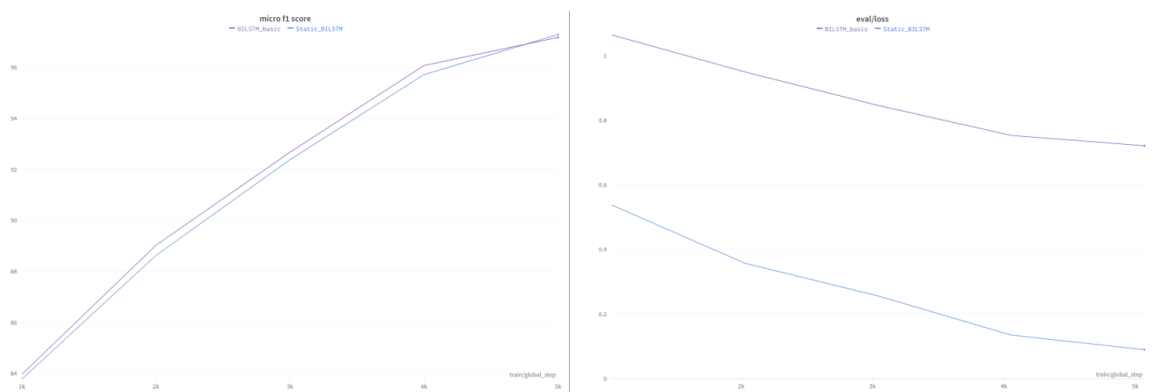
- 기본적으로 슬랙, Git, Zoom 등 협업 툴을 이용하여 의사소통을 원활히 하였으며,
  - 성능향상에 도움이 될만한 논문이 있어 팀원간 원활한 이해 향상을 위해 논문을 한글로번역작업을 함
  - 모델의 예측값과 실제 라벨의 값을 편리하게 분석할 수 있는 기능이 Wandb에 있어 이를 팀원들과 공유하고자 Wandb사이트에 올라온 사용법을 한글로 번역하여 요약함
- 나는 어떤 방식으로 모델을 개선했는가?
  - 우선 EDA를 통해 데이터에 오류 데이터(entity 태깅 오류)가 있음 확인하고 수정 작업을 수행함. 하지만 모델의 성능 향상이 전혀 없었으며 심지어 하락함. 혹시나 싶은 마음에 테스트 데이터의 태깅을 확인하였으나 훈련 데이터와 동일한 오류가 있음을 확인함.
  - 위의 오류 데이터 수정 작업을 통한 성능향상은 어려울 것 같다는 팀원들의 의견에 따라 backtranslation을 통한 데이터 augmentation 작업에 참여하였으나 성능 향상이 되지 않음.
  - 마지막으로 현재 SOTA모델 성능향상을 위해 모델이 예측한 라벨과 실제 정답 라벨의 확률분포를 반영하여 잘못 예측한 값에 대해 더욱 빠르게 학습할 수 있도록 모델 튜닝을 실시함.
- 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?
  - 관계 추출(Relation Extraction)의 경우 주관에 따라 해석이 달라질 수 있어 데이터를 제작하는 사람에 따라 정답이 달라 질수 있으며 따라서 데이터를 수정한다고 무조건 성능이 향상되는 것은 아니라는 점을 알 수 있었음
  - 모델을 잘 만들어 성능을 향상 시키는 것도 중요하지만 pretrained 모델을 잘 활용하는 것이 훨씬 더 효율적임을 체험할 수 있었고 평소에 여러 모델에 대한 학습이 필요함을 느낌
- 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?
  - EDA를 비중있게 생각하지 않아서 pandas, matplotlib라이브러리 등을 잘 사용하지 않았는데 막상 사용하려니 생각대로 자유롭게 다루기 어려웠는데 이번에 많이 사용해 보면서 익숙해 지게 되었음
  - 데이터 오류 수정작업에 총 3주간의 기간 중 1주일을 투자하였는데 노력대비 성능향상이 전혀 없어 실망스러웠지만 데이터 분석에 대한 기술적인면 이외에도 분석 결과에 대한 해석능력이 많이 향상됨을 느낄 수 있었음.
  - wandb라는 툴을 이용하여 코드를 크게 변형하지 않고도 모델 hyperparameter 튜닝 및 모델 예측결과를 편리하게 분석할 수 있게 되어 전보다 효율적으로 작업을 할 수 있게 되었음



- pretrained 모델에 fc layer를 추가하거나 교체하는 간단한 튜닝은 전에도 할 수 있었지만 새로운 아이디어를 추가하여 모델을 튜닝하는 작업을 통해 조금더 자유롭게 모델을 다룰 수 있었음.

○ 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 짧은 기간 내에 많은 내용을 학습하다보니 학습한 내용을 완전히 활용하지 못하였음.
- Git을 기능을 적극적으로 활용해 보지 못함.
- EDA에 너무 많은 시간을 할애하여 모델 성능향상에 투자하는 시간이 부족하였고 대회종료가 임박해서야 모델을 튜닝을 완성하였는데 최종제출까지 해보지 못하였음. 이 후 최종 제출 모델과 튜닝모델을 동일조건에서 micro f1 score와 loss 성능을 확인해 보았는데 micro f1 score는 거의 비슷한 하였지만 loss감소와 관련하여 튜닝모델의 성능향상을 확인할 수 있었음.



○ 한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?

- 학습 내용 중 중요한 부분은 최대한 기록하여 다음에 최대한 활용할 수 있도록 하겠음
- 현재 수행 중인 작업에 대한 시간투자대비 효율성을 고려하는 것과 작업을 어느 정도 진행된 이후 계속 진행할지 여부에 대해 빠르게 판단하는 것이 필요함
- 개발이 도움이 되는 내용에 대해 팀원들과 더욱더 적극적으로 공유할 하고 도움이 필요할때 더 적극적으로 팀원들에게 요청하도록 하겠음

# 개인회고록(김남현)

○ 이번 프로젝트에서 나의 목표는 무엇이었는가?

- 깃허브
  - 지난 대회와 달리 vscode를 통해 V100서버에 원격 접속하고, 깃허브와 연동해서 작업하기.
  - baseline 코드로부터 branch를 만들고 다양한 실험 시도하기.
- huggingface의 다양한 모델 시도해보기.
- wandb 사용 및 sweep 사용하기.
- 문장의 단어 간 관계 추론 관련 논문 읽어보기.

○ 나는 어떤 방식으로 모델을 개선했는가?

- EDA를 통해 주어진 데이터가 가진 문제를 파악하고, 데이터프레임을 수정했다.
  - sentence, subj\_entity, obj\_entity가 전부 동일한 중복된 데이터를 식별한 후, 제거했다.
  - 각각의 데이터 튜플 속에서 subj\_type, obj\_type을 추출해서 각각 새로운 칼럼을 만들어서 추가했다.
  - label마다 번호를 부여해서 튜플이 가진 라벨에 해당하는 번호로 새로운 칼럼을 만들어서 추가했다.
- huggingface 모델 실험.
- wandb와 sweep을 이용한 모델 개선.
- 데이터 교정 작업 후 정기원 캠퍼님이 제안한 backtranslation 작업 참여.

○ 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?

- 처음에 klue/roberta-large 모델과 함께 baseline code에서 learning rate, batch size, 입력 sentence의 최대 길이만 조절했는데 micro\_f1 score가 약 71이나 나왔었다. 예상보다 좋은 결과를 얻어서 이후로 klue/roberta-large만 썼는데, 오히려 이 결과 때문에 다른 모델을 한 번도 시도하지 않았다는 게 아쉽다.
- wandb의 sweep 기능의 효과를 경험했다. 이후 대회에서도 하이퍼파라미터 튜닝 때 유용할 것 같다.

○ 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

- EDA와 데이터 검수 및 교정 작업에 공을 들였다. 하지만 테스트 데이터에도 똑같이 잘못된 데이터가 많았기에 데이터 교정 작업은 기대한만큼의 효과를 가지지 못했다.
- wandb를 사용해서 현재 실험하고 있는 모델의 진행 상황과 성능을 알 수 있었다. 이를 통해 실험을 거듭하면서 조금씩이나마 개선할 수 있었다.

○ 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 처음으로 서버에 원격 접속한 채로 vscode와 깃허브를 연동해서 작업했지만, 여전히 낯설게 느껴졌다. 팀원들에게 도움을 많이 받았으나 vscode를 통해 여러 모듈을 관리하는 것과 터미널 사용법이 어려워 한 것

이 아쉽다.

- 워낙 여러 작업을 병행하다보니 팀원들이 새로 작성한 코드뿐만 아니라 내가 작성한 코드도 시간이 지나고 보니 이해하기 어려웠다. 코드 주석의 중요성을 알게 되었다.
  - 관련 논문을 읽고, 코드로 직접 구현하는 게 어렵게 느껴져서 시도하지 못했다.
  - 잘못된 데이터가 많아서 팀원들과 전체 데이터를 검수하고 교정했지만, 테스트 데이터에도 문제가 많아서 노력한만큼의 모델 성능 향상은 없었다.
  - backtranslation 작업이 흥미롭고, 좋은 성능을 낼 거라고 기대해서 팀원들과 함께 파파고 번역기를 이용해서 시도했지만, 이 또한 모델 성능을 향상시키지 못했다. 구글 번역기를 사용한 팀은 모델 성능이 좋아졌다고 하니 아쉽다.
  - 대회 기간 동안 팀원들 간 소통은 잘 되었으나, PM 역할을 맡은 사람도 없었고 체계적으로 계획을 세워서 움직이지 못했다.
- 한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?
- EDA 작업을 빨리 마쳐서 데이터 문제 정의와 개선 방안을 조금 더 일찍 제시하고 싶다.
  - wandb의 sweep 기능을 더 정교하게 사용해서 하이퍼파라미터 튜닝 작업을 하고 싶다.
  - 시간과 일을 할당해서 체계적으로 프로젝트를 진행하고 싶다.
  - 작성한 코드마다 주석을 상세하게 써서 팀원들이 코드를 이해하기 쉽게 돕고 싶다.

# 개인회고록(전태양)

○ 이번 프로젝트에서 나의 목표는 무엇이었는가?

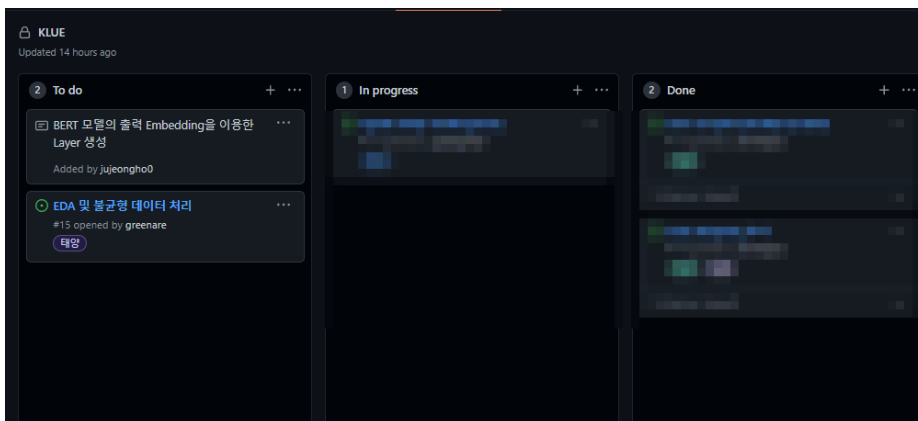
- 저번 프로젝트에서 개인적으로 아쉬웠던 점은 팀원들과 프로젝트 관련하여 다양한 의견 교류가 없었다는 것이었다. 특히 저번 프로젝트 1등 팀의 발표를 통해 활발한 협업에 대한 자극을 많이 받았다. 그렇기 때문에 이번 프로젝트에서 개인적인 목표는 “매번 진행한 실험을 기록하고 적극적으로 팀 협업에 참가하기”였다.

○ 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

- **팀원들과 Wandb 공유** : 팀원들과 실험 결과를 효과적으로 공유하기 위해 Wandb 사용 방법을 공유했다.



- **대회 시작 전 협업을 채택** : git issue를 활용하여 효율적인 협업을 이뤄내고자 했다.



- 팀원들과의 협업 진행 : 데이터 전수조사 같이 팀원들이 혼자하기 벅찬 부분이 있다면 최대한 도우려고 노력했다.

○ 나는 어떤 방식으로 모델을 개선했는가?

- EDA
  - 첫 주에 Pandas와 Matplotlib, Seaborn을 활용하여 EDA를 진행, 전반적인 데이터 분포를 파악했다.

- 이후 EDA를 기반으로 데이터의 문제점을 파악하여 모델을 개선하고자 했다.
  - Huggingface
    - 비교적 가벼운 koelectra 모델을 개선하며 좋은 성능을 내고자 했다.
    - Huggingface내의 Trainer를 Custom하여 사용했다.
      - Focal Loss 추가
  - StratifiedKFold를 활용하여 모델 학습
  - Entity type을 한국어로 치환하여 입력
    - ex) PER : 사람, ORG : 기관
  - 잘못 기입된 Label 수정
- 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?
- Train valid를 나누지 않고 Train에 Random seed를 부여하여 학습시키는 방식이 있다는 것을 깨달았다.
  - train 데이터가 올바르게 라벨링 되어 있더라도 Eval 데이터 셋이 잘못되어 있는 경우가 있을 수 있다는 것을 알았고, 그로인해 학습 점수가 하락할 수 있다는 것을 알았다.
  - 토큰라이저의 maxlen을 256으로 설정하고 모델의 배치 사이즈를 64개 이상 설정하여 학습할 경우 OOM이 발생한다. 그렇다 보니 배치 사이즈를 바꿔가며 실험을 진행하는 경우가 많았는데 learning rate를 3e-5로 고정해두는 것에 대한 의문을 가졌다. 이후 다음과 같은 자료들을 찾아보고 <https://inovation97.tistory.com/32>, <https://www.baeldung.com/cs/learning-rate-batch-size> 배치 사이즈와 learning rate간에 선형 관계가 있다는 것을 알게 되었다.
- 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?
- 이전 대회와 달리 할 일과 실험 결과를 하나하나 기록했다. 실험 결과를 체크해두니 실험을 시작할 때 이전에 실험했던 것과 겹치지 않는지 파악하기 용이했고, 할 일을 미리 적어두니 보다 효율적으로 필요한 자료를 찾을 수 있었다.
  - 팀원들과 프로젝트 진행 상황과 결과를 공유하려 노력했다. 각자 어떤 생각을 갖고 작업을 진행하는지 파악하기 용이했고, 같은 task에서 문제가 발생하더라도 팀원들과 공유하며 해결하는 속도가 훨씬 빨라졌음을 체감했다.
- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?
- EDA 과정에서 Embedding Visualization을 진행하여 군집별 이상치나 모델이 분류에 어려움을 겪는 sample 확인이 가능했을 텐데 시도할 생각을 뒤늦게 했다.
  - 초반에 koelectra 모델이 Roberta-Large에 비해 성능이 낮게 나오에도 불구하고 계속 잡고 있었다. koelectra 모델이 진행하고 있는 RE Task에 맞는 모델인지 고민해볼 필요가 있었다.
  - 아직까진 git으로 issue를 다루거나 branch 접근이 미숙하다는 생각이 들었다. 이후 프로젝트 시작 전에 git하고 더 친해질 필요가 있다.
  - 시도했던 것들이 성능 향상에 도움되지 않아 아쉬웠다. 생각해보면 구현하다가 성능이 좋지 않은 경우 끝까지 해보지 않고 도중에 포기하는 경향이 나타났던 것 같다. Focal Loss의 gamma값과 같은 hyperparameter를 더 조작해 볼 필요가 있었으나 그러지 않았다.

- 아직까지 내가 구현한 것에 대한 확신을 갖지 못한다는 느낌이 들었다. 이 역시 내가 더 배우고 성장해야한다는 것을 의미한다고 생각한다.
  
- 한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?
  - 하나의 Task에 대해 장인 정신을 담아 더 깊게 탐구하는 자세를 가져야겠다.
  - 구현 능력을 키우기 위해 내가 찾은 논문이나 자료를 적극적으로 구현해 볼 계획이다.

# 개인회고록(정기원)

## 개인회고록(정기원)

○ 이번 프로젝트에서 나의 목표는 무엇이었는가?

- 팀원과 협업을 원활히 할 것, 시각화 도구를 최대한 써볼 것, 성능 향상을 나만의 방법으로 해낼 것, 체계적으로 실험을 설계하고 수행할 것
  - 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?
- 버전관리도구, 시각화도구에 대해 소개하고 연습하는 시간을 팀원과 가졌음
- 본 대회 데이터셋과 유사한 TACRED RE task를 다룬 논문을 읽고 발표함
- Backtranslation 구현 게시물을 읽고 나만의 방식으로 응용하려 노력함
- 각각의 실험마다 실험명을 적어 팀원과 겹치는 실험이 없도록 하였음
  - 나는 어떤 방식으로 모델을 개선했는가?
- 베이스라인 코드를 수정하여 원하는 기능을 넣을 수 있도록 하였음
- Backtranslation을 수행하기 위해 웹 크롤링, 200자 제한 개선, Encoded Entity 도입을 시도함
- 데이터 전수 조사를 통해 잘못된 데이터를 수정하고자 함
  - 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?
- 베이스라인 코드를 깔끔하게 수정하고 Docstring을 작성해서 팀원이 해당 기능에 대해 쉽게 이해할 수 있도록 노력하였음. 협업의 즐거움을 알게 되었음.
- Backtranslation 및 데이터 전처리 결과가 좋지 못했음. EDA의 중요성에 대해 깨닫게 되었음.
- 실험 명을 적었지만, 비슷한 실험을 반복한 경우가 가끔 발생하였음. 다같이 공유하는 실험 테이블을 만들어 같은 실험을 반복하지 않도록 노력하는 것이 중요함.
  - 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?
- 최대한 작업물의 가독성을 높이기 위해 Jupyter notebook을 통해 결과물을 시각적으로 보여주고, Docstring 및 Notation을 적극적으로 활용하였음. 팀원이 내 코드를 잘 이해해줘서 매우 기분이 좋았음.
  - 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?
- Backtranslation 및 데이터 전처리 결과가 좋지 못했음. EDA를 좀 더 꼼꼼하게 하는 것이 필요했음.
  - 한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?
- 실험 테이블을 작성하여 현재 어떤 실험이 진행 중인지 체계적으로 정리하려고 함.
- EDA 방법론에 대해 공부해보기

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/13fb9778-6f97-4119-960c-e5dff96c0da0/0410\\_KLUE\\_개인회고록\\_\(정기원\).docx](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/13fb9778-6f97-4119-960c-e5dff96c0da0/0410_KLUE_개인회고록_(정기원).docx)

# 개인회고록(최지민)

- 이번 프로젝트에서 나의 목표 & 달성하기 위해 무엇을 어떻게 했는가?
  - huggingface의 transformers 라이브러리 사용법 익히고 customizing
    - huggingface의 여러 모델 시도 및 TrainingArguments 조정
    - Trainer에 EarlyStoppingCallback 추가하여 early stopping 사용
  - Github 활용
    - baseline 코드에서 각자 이름 / 실험 이름으로 branch 만들어 작업
    - project와 issue 활용
    - 팀원들과 공통으로 합의된 것만 Pull Request
  - wandb를 통한 실험 관리 및 hyperparameter tuning
    - 모든 실험을 logging하여 실험 간 차이를 분석
    - sweep 사용법을 팀원들에게 소개, hyperparameter(learning rate, batch size 등) tuning 시도
- 나는 어떤 방식으로 모델을 개선했는가?
  - pretrained 모델을 finetuning 하는 방식으로 여러 모델을 실험(bert-base, roberta-large)
  - 중복 데이터 처리 및 data augmentation
  - 입력 문장(query) 개선 시도
- 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?
  - overfitting을 우려하여 train-validation을 나누고 early stopping 기능을 구현하였다. 여러 실험을 통해 약 epoch 5 이후에 validation loss가 증가함을 확인하였다. 그러나 train-validation을 나누어 학습한 모델은 모든 데이터를 사용하여 학습한 것보다 성능이 떨어졌다.
  - data augmentation 작업을 하였으나, 단어와 어순이 중요한 NLP에서 쉽지 않은 작업임을 깨달았다.
    - EDA를 통해 발견한 insight( per:parents 의 S(subject)와 O(object)를 바꾸면 per:children 이 되는 등) 이용 → 미미한 성능 향상
    - backtranslation(한-영-한) 작업 참여 → 성능 향상 X
    - KoEDA 라이브러리 사용한 Easy Data Augmentation → 성능 향상 X
  - "S과 O의 관계"라는 query는 순서를 반영하지 못한다고 생각하여, 순서가 다르면 의미가 달라지는 query("From S To O", "S에게 O")를 사용하였다. CLS token으로만 classification할 땐 성능 향상 효과가 있었으나, 모든 token을 받아 BiLSTM을 태우는 모델에선 그렇지 않았다.
- 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?
  - notebook이 아닌 파이썬 파일로 모듈화하여 실험을 진행해서, 관리와 확장이 용이하였다.



- 모든 팀원들과 wandb로 실험 logging을 하였다. 이를 통해 다른 팀원들이 진행한 실험 결과를 쉽게 알 수 있었고, 결과 비교 및 다음 실험 설계에 도움이 되었다.
  - Github를 통해서 현재 팀 최고 성능을 낸 코드에 쉽게 접근 및 발전시킬 수 있었다.
- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?
- RE task에 적용할 수 있는 자료들을 찾는 데에 익숙지 못했다.
  - 앙상블에 다양한 모델을 사용하지 않고, 그저 f1 score 상위 5개 모델로 soft voting한 것이 아쉽다.
- 한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?
- 능동적으로 논문 등을 찾아보고 아이디어를 얻어와 정확하게 구현해보고 싶다.
  - 앙상블에 대해 더 공부하여 효과적인 모델을 만들고 싶다.

# 개인회고록(주정호)

- 이번 프로젝트에서 나의 목표 & 달성하기 위해 무엇을 어떻게 했는가?
  - 학습 데이터 전처리
    - 논문을 탐색하여 Typed Entity marker(punct) / Typed Entity marker(punct) + Query / Standard with Entity Location Token의 방법으로 학습 데이터를 전처리
  - huggingface의 transformers 사용
    - bert, roberta 모델 사용
    - TrainingArguments를 이용해 warmup, label smoothing, weight decay 등을 사용
    - FC, BiLSTM, BiGRU, MHA 등을 이용해 transformer를 customizing
  - 앙상블
    - 다수의 결과를 Soft Voting하여 점수 향상
  - github를 이용한 협업
    - 각자의 branch에서 작업
    - 구현한 추가 기능 등을 issue 및 pull request로 공유
  - wandb를 이용한 협업
    - 실험의 특징(하이퍼 파라미터, 모델 등) 기록
    - 실험 결과 공유
- 나는 어떤 방식으로 모델을 개선했는가?
  - Typed Entity marker(punct) + Query의 방법으로 학습 데이터를 전처리
  - 기학습 모델 "klue/roberta-large"에 FC, BiLSTM, BiGRU, MHA 등을 이용해 다양한 모델을 생성
  - TrainingArguments를 이용해 다양한 하이퍼 파라미터, 다양한 기능(warmup, label smoothing, weight decay 등)을 사용해 학습 및 결과 예측
  - 다양한 결과 중 상위 5개의 결과를 Soft Voting
- 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?
  - 논문을 따라 Typed Entity marker(punct) + Query의 방법으로 학습 데이터를 전처리를 하여 SOTA 모델을 달성하였지만, 왜 결과가 좋은지는 이해가 되지 않았다. 이를 통해, 코딩 실력도 중요하지만, 논문 조사의 중요성도 느낄 수 있었다.

- 앙상블을 이용해 점수를 1.5% 올렸고, 이를 통해 앙상블의 효과가 대단하다는 것을 느낄 수 있었다.
  
- 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?
  - github를 이용한 협업으로 팀원들끼리 다양한 시도의 코드를 쉽게 공유하고 접근할 수 있었다.
  - wandb를 이용한 협업으로 팀원들끼리 다양한 실험 결과를 쉽게 공유하고 접근할 수 있었다.
  
- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?
  - 실험의 특징과 결과를 자세하게 기록하지 못했다.
  - Backtranslation 대신 영어로 번역해, 영어 데이터를 이용해 Task를 수행하고 싶었는데, 시간이 부족해 하지 못해 아쉬웠다.
  
- 한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?
  - 학습 데이터를 영어로 번역해, 영어 데이터를 이용해 Task를 시도해볼 것이다.