

# NLP\_07조 랩업 리포트

## 1. 프로젝트 개요

- 한국어 모델의 성능을 평가하기 위한 데이터셋인 KLUE(Korean Language Understanding Evaluation)의 8가지의 대표적인 task중 하나인 관계 추출(RE, Relation Extraction)을 수행하는 모델 제작
- 관계 추출이란, 문장 내 두 단어(Entity)의 속성 및 서로에 대한 관계를 예측하는 task이다.

## 2. 프로젝트 팀 구성 및 역할

공통	모델 성능 테스트, 하이퍼파라미터 성능 실험, <b>fine-tuning</b>
양서현	데이터 클리닝, WandB sweep h-p 서치, 데이터 전처리(Input Format 변경)
이상경	데이터 분할, 하이퍼파라미터 튜닝
이승백	데이터 분석 및 <b>sampling</b>
이주용	데이터 분석 및 <b>sampling</b> , 앙상블, 모델 서치 및 성능 테스트
정종관	데이터 전처리(중복값제거), <b>focal loss</b> , 추론 후처리
정지영	데이터 전처리(Special Token)

## 3. 프로젝트 수행 절차 및 방법

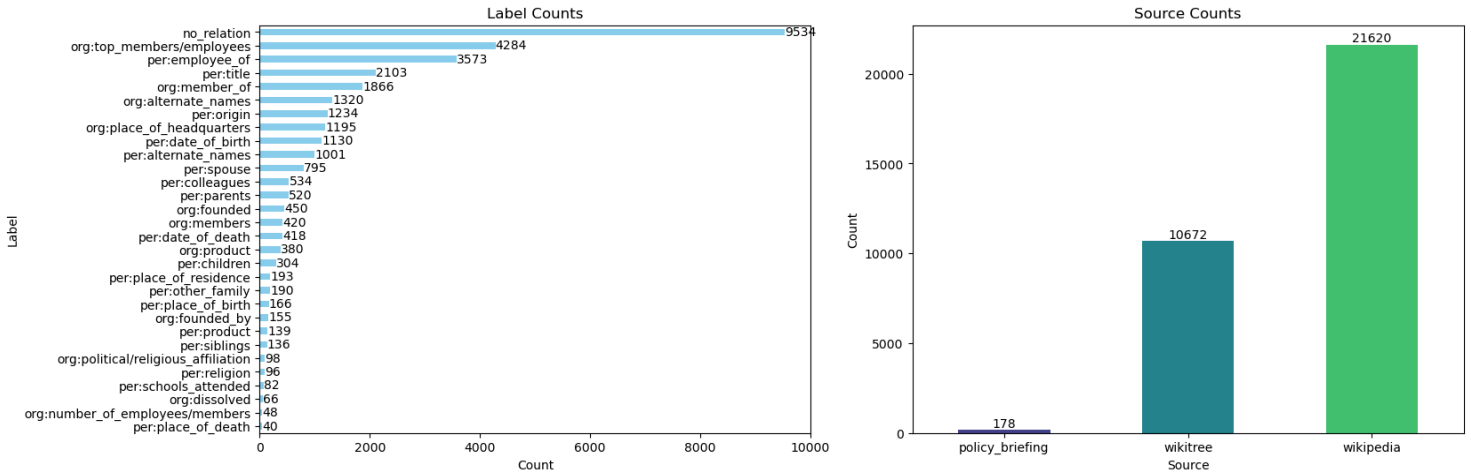
### 1) 환경 구축

- 서버 환경 구축
- Github**를 통한 협업 환경 구축
  - Github Issue / PR 컨벤션 수립, **projects** 자동 등록 및 운영
- WandB**를 활용한 모델 실험 모니터링
  - 실험이 진행되는 동안 주요 값(**loss**, **f1 score** 등) 추적
  - Sweep**을 활용하여 최적 모델의 주요 하이퍼 파라미터 최적값 탐색

## 2) 데이터

- 데이터 분석 - EDA

- Train : (32470,6), Test : (7765,6)
- Train 데이터는 가장 많은 라벨(no\_relation)이 9,534 개, 가장 적은 라벨(per:place\_of\_death)가 40개인 불균형 데이터



- 데이터 클리닝

- Sentence / subject\_entity / object\_entity 전부 동일한 train data 조회
- Label이 다른 데이터 drop

- 데이터 분할

- Train데이터에서 [Subject\_type, Object\_type] pair 별로 validation 데이터셋 추출

- 데이터 전처리

(3강-실습-1) Special Token 추가 방법론과 논문 [Unified Semantic Typing with Meaningful Label Inference](#) 와 [Unified Semantic Typing with Meaningful Label Inference](#) 을 참조하여 **Input Format**을 변경하였다.

\* sentence = <Something>는 조지 해리슨이 쓰고 비틀즈가 앨범에 담은 노래다

- **Special Token** 활용

### A. Entity Marker(Masked Entity)

Sentence에서의 두 Entity word 대신 각각 <S-type>, <O-type>으로 Masked 하여 학습의 input format을 변경하였다. 또한, <S-type> 와 <O-type>에 해당하는 Special token을 추가하고 Embedding Layer을 추가하였다.

e.g. <Something>는 <O-PER>이 쓰고 <S-ORG>가 앨범에 담은 노래다.

### B. Typed Entity Marker with Special Token

Sentence에서 두 Entity word를 각각 [S:Type] Subject word [/S:Type], [O:Type] Object word [/O:Type] 형태의 Special token으로 감싸주는 형태로 변경하였다.

e.g. <Something>는 [O:PER] 조지 해리슨 [/O:PER]이 쓰고 [S:ORG] 비틀즈 [/S:ORG]가 앨범에 담은 노래다.

### C. Typed Entity Marker with Punctuation

Special Token이나 Embedding Layer을 추가하지 않으면서 각 Entity word의 위치와 Type을 알리는 방법이다. 새로 학습시켜야하는 Special Token보다는 기학습된 “ @, \*, #, ^ “ 특수문자를 이용해 Entity의 정보를 표현하는 Input Format으로 변경해준다면 모델의 성능이 향상될 것이라는 가설을 세워 실험하였다.

e.g. <Something>는 #^PER^조지 해리슨#이 쓰고 @\*ORG\*비틀즈@가 앨범에 담은 노래다.

#### ○ Add Semantic Typing Query

Bert와 같은 사전 학습 모델의 학습 방식에 NSP(Next Sentence Prediction)이 있는 것에 착안하였다. 기존 **Subject Entity word [SEP] Object Entity word [SEP] Sentence** 형식의 format을 두 Entity word 대신 두 단어의 관계를 설명하는 Semantic Typing으로 Query를 생성해, C. Typed Entity Marker with Punctuation으로 변경된 sentence 형태와 함께 전달해주는 input format으로 변경하였다.

#### e.g.1. Semantic Query + C. Typed Entity Marker with Punctuation(ENG)

@\*ORG\*비틀즈@과 #^PER^조지 해리슨#는 ORG와 PER의 관계이다. [SEP]

<Something>는 #^PER^조지 해리슨#이 쓰고 @\*ORG\*비틀즈@가 앨범에 담은 노래다.

#### e.g.2. Semantic Query + C. Typed Entity Marker with Punctuation(KOR)

@\*기관\*비틀즈@과 #^사람^조지 해리슨#는 조직과 사람의 관계이다. [SEP]

<Something>는 #^사람^조지 해리슨#이 쓰고 @\*조직\*비틀즈@가 앨범에 담은 노래다.

#### e.g.3. punctuation + Semantic Query (순서변경)

<Something>는 #^PER^조지 해리슨#이 쓰고 @\*ORG\*비틀즈@가 앨범에 담은 노래다.

[SEP] @\*ORG\*비틀즈@과 #^PER^조지 해리슨#는 ORG와 PER의 관계이다.

3가지 경우, 모두 기존과는 확연한 성능 향상이 있었다. 그 중에서도 e.g.1.의 형태를 사용했을 때 가장 성능이 좋았다. (단일 모델 실험 중 SOTA)

## 3) 모델링

### ● Baseline 모델 실험

- klue/bert-base

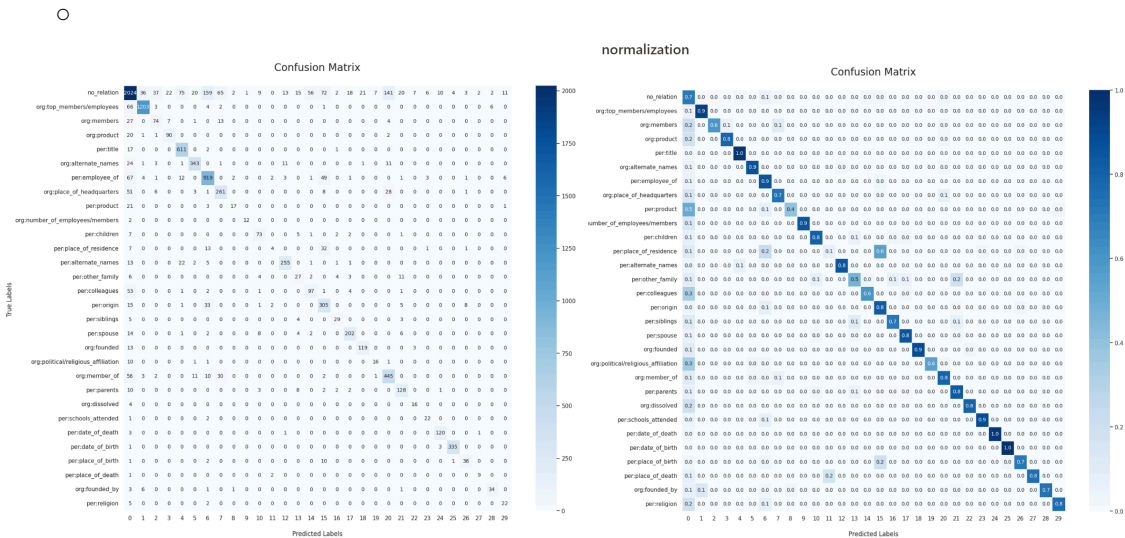
### ● Pre-trained 모델 탐색

: 한국어로 사전 학습된 모델들 중 RE에 많이 사용되는 모델들을 각각 4 epoch 훈련시킨 후, submit score 기준으로 가장 나은 성능을 보였던 klue/roberta-large 모델 선택

- klue/roberta-large
- snunlp/KR-SBERT-V40K-klueNLI-augSTS
- snunlp/KR-BERT-char16424
- paust/pko-t5-base, large
- paust/pko-flan-t5-large

## ● Confusion Matrix

학습의 성능평가를 위해 오답의 분포를 Confusion Matrix로 확인하였다. Valid Set이 따로 주어지지 않아 주어진 Train set의 데이터를 7:3 으로 Stratified하게 나눠 Valid Set을 생성하고, Train Set으로만 학습 후 Valid Set을 Test Set으로 Inference하여 실제 Valid Set의 True Label과 비교하여 예측정도를 시각화로 확인하였다. Label 수가 불균형하여 적은 Label의 수의 오답이 많을 것이라 예상하였으나, 예상과는 다르게 학습데이터가 가장 많은 “no\_relation”의 오답 비율이 꽤 높음을 확인할 수 있었다.



## ● Hyper Parameter tuning with Wandb Sweep

- epoch, batch, learning rate에 대해서 어떤 조합이 최적의 성능을 발휘하는지 객관적으로 평가할 수 있는 방법과 지표가 필요했다. WandB sweep을 활용해 다양한 조합을 자동으로 실행하고, 시각화를 통해 직접 그 결과를 관찰할 수 있도록 하고자 했다. 가장 높은 f1 score을 보이는지 근거해 최적의 조합을 판단했다.

## ● 실험 내용

- batch\_size = [8, 16, 32]
- Num\_train\_epoch : {"min": 3, "max": 6}
- leraning\_rate : {min: 7e-6, max: 5e-5}
- loss\_function : Cross-Entropy, Focal Loss
- Seed : {"min": 1, "max": 256}

## ● 최적의 하이퍼 파라미터

모델	Batch size	learning rate	epoch	Loss func	seed
klue/roberta-large	16	1.8e-5	4	Focal loss	53

## ● 추론 결과 후보정

- train data에서 subject\_type과 label head(org:, per:)가 일치하지 않는 경우는 32470개의 데이터 중 4개임을 확인

- 베이스라인 코드 실험 결과, test data 7765개 중 57개의 data에서 subject type과 predicted label head가 일치하지 않는 것을 확인
- 모델 예측 시, subject type과 일치하지 않는 label head를 고려하지 않게 함으로써, 성능 향상

#### 4) 앙상블

- **Soft voting**


: 다양한 전처리 방식과 서로 다른 seed 가진 모델을 통해 나온 결과를 Soft voting ensemble 수행

- **Soft voting 결과끼리의 Soft voting**

: 서로 다른 조건에서 나온 앙상블 결과를 또 한 번 앙상블하여 근소한 성능 향상을 이루어냄

단일 모델 최고점에서 soft voting후 **75.4402** → **76.9665** 점 기록

#### 4. 프로젝트 수행 결과

순위	팀 이름	팀 멤버	micro_f1 ↕	auprc ↕	제출 횟수	최종 제출
1 (1 ▲)	NLP_07조		75.9857	83.5148	77	4d

- Private micro f1 score 75.9857로 최종 순위 1위로 마감

#### 5. 자체 평가 의견

**+ Keep** : 잘했던 것, 좋았던 것, 계속할 것

- 1주차에 수업을 다 듣고 각자 충분히 데이터를 살펴보는 시간을 가지는 것이 좋았다.
- 데이터를 충분히 들여다봄으로써 이상적인 추론 결과에 대해 아이디어를 얻을 수 있었다.

**- Problem** : 잘못했던 것, 아쉬운 것, 부족한 것 → 개선방향

- 각자 실험을 진행하다 보니 하나의 프로젝트로 통합하는 시기를 놓친 부분이 있다.
- 몇 가지 모델을 사용해 봤으나 전처리가 roberta에 적합한 방식들이라서 성능 차이가 커 (micro f1 score기준 4 point) ensemble에 roberta만 사용한 점이 아쉽다.
- Task에 대한 사전 조사가 부족했다. 문제 정의로 시작하여, task관련 리서치를 먼저 탐구했으면, 관련 기법들을 일찍 알 수 있었을 것.
- Pre-trained 된 모델을 다양하게 알았더라면 더 좋았을 거라는 아쉬움이 존재한다.

**! Try** : 도전할 것, 시도할 것

- 다음 프로젝트에서는 Baseline 코드를 먼저 협의해서 공유한 상태로 실험을 진행하면 좋을 것 같다.
- 동일한 조건으로 실험할 수 있도록 실험 페이지를 만드는 것이 좋을 것 같다.

- 다른 모델들을 사용할 때 단순히 모델을 변경하는게 아니라 해당 모델에 대한 이해를 통해 그 모델에 적합한 방식으로 최적화 하여 사용해봐야 할것 같다.
- 부트캠프 남은 기간동안 몸관리에 신경을 쓰자!

## 6. 개인 회고

### 이상경\_T6121

#### I. 나는 무엇을 했는가?

- A. 데이터 분석 : 데이터의 라벨별 분포 확인, **subject** 단어의 타입에 따른 나올 수 있는 **label**의 종류 파악
- B. 데이터 증강 : 데이터를 보면 **object** 단어와 **subject** 단어를 바꿔도 유지되는 문장들이 존재한다. 예를 들어 **alternated\_name** 은 단어의 위치를 바꿔도 같은 의미이고, **parents**인 라벨은 라벨의 위치를 바꾸면 **children**이 된다. 이를 통해 데이터 증강을 시도해 보았지만 유의미한 변화는 존재하지 않았다.
- C. 모델 비교 : **klue/bert-base**, **klue/roberta-large**, **snunlp/KR-ELECTRA-discriminator** 등 여러모델을 사용해본 결과 **klue/roberta-large**의 성능이 제일 뛰어나서 이 모델을 선택후 개선을 해나갔다.
- D. 전처리
  1. 단어를 **type**으로 변환하기 - 처음에는 문맥상으로 단어를 타입으로 같은 고유어로 의미차이가 없을 것이라고 생각하여 진행하였다. - 성능 감소
  2. 단어의 인덱스 번호를 추가하여 순서 인식시키기 - 기본모델에 비하면 성능 감소
  3. 단어를 그대로 유지시키고 태그 붙이기 - 소폭 상승
  4. 태그를 앞뒤로 감싸고 안에 타입과 단어를 모두 기입 - 성능 향상
  5. 기존 **sentence** 에도 단어를 4번과 같이 변환 - 성능 향상
  6. 태그를 **special token**대신 이미 저장된 특수문자로 변환 - 성능 향상
  7. **Semantic Query** 사용 - 성능 향상
  8. 단어의 타입을 한국어로 치환 - 성능 향상
- E. 하이퍼 파라미터 튜닝 : **Wandb sweep**을 통해서 **seed**와 **learning rate**의 값 중 가장 좋은 값을 발견하고 적용하였더니 성능이 크게 향상되었다.
- F. 앙상블 : 다양한 전처리 방법과 동료들이 행한 다양한 전처리 방법을 **soft voting**하여 앙상블을 하였더니 **score**가 상승하였다.

#### II. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

- A. **Pre-training** 된 모델은 기본적으로 임베딩이 되어있기 때문에 단어의 유사도가 어느정도 측정되어있다. 따라서 관계를 정할경우 **special token**으로 처리를 하기보다는 이미 임베딩 된 단어를 사용하는 것이 더 효과적인 것 같다.

#### III. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

- A. 테스트 점수와 **validation** 점수의 차이가 크게 존재하여서 **validation** 데이터셋을 구성하는데 굉장히 많은 시간을 투자하였다. 이를 통해 **f1 score**가 크게 상승하였다.

#### IV. 아쉬웠던 점은 무엇인가?

- A. 위에 새롭게 시도한 변화에서 적은 것에서 아쉬운 점이 존재하는 데 **object - type**과 **subject - type**의 **pair**를 기준으로 **validation** 을 나눌뿐만아니라 **label**을 기준으로 나눈 것도 추가를 했어야 했다고 생각한다. 하지만 나눌 당시에 이 기준을 한번에 나눠야한다는 생각이 강하여 각각 나누면 된다는 것을 생각하지 못하였다.

## 이승백\_T6126

### 1. 학습 목표

- KLUE 강의 및 미션을 잘 이해하고, 데이터 분석과 전처리 및 모델링을 수행하여 팀에 기여하는 것입니다.

### 2. 학습 목표 달성을 위해 무엇을 어떻게 했는가?

- KLUE 강의를 통해 기본 개념을 학습하고, 기본 코드를 분석하며 개선했습니다.
- 데이터 불균형, **Object**와 **Subject Entity Type** 등 데이터 분석을 통해 여러가지 성능 개선 방법을 제시했습니다.
- **Validation set**을 만들어서 성능 개선 제시했습니다.

### 3. 실패한 시도와 그 과정에서 얻은 교훈은 무엇인가?

- 여러 방법으로 데이터를 **sampling**하고 전처리하여 성능향상을 기대했지만 결과는 좋지 못했고, 결과의 이유를 분석했지만 제대로 되지 않았는지 결론적으로 성능 향상은 없었습니다. 어떠한 방법이든 빠르게 적용하여 결과를 확인하고 개선하는 것이 시간적, 성능적으로 좋을 것 같습니다.
- 데이터를 분석할 때는 여러 시점으로 자세히 들여다 보아야 한다는 것을 알게 되었습니다. 하나의 아이디어에 꽂혀 매몰되지 않도록 주의해야겠습니다.

### 4. 어떤 방식으로 모델을 개선했는가?

- **Object**와 **Subject Entity**과 **Label** 분석을 통한 개선 아이디어 제시.

### 5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 감기에 걸려서 프로젝트에 온전히 집중하지 못한 것이 아쉽습니다. 팀원이 실험을 깔끔하게 정리하고 공유한 것을 보고 많이 배우는 것 같습니다. 팀원들의 아이디어와 실험이 빠르게 공유가 되면 더 시간을 알차게 쓸 수 있을 것이라 예상됩니다.

### 6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- **Base Line**의 코드들을 실험을 빠르게 적용할 수 있도록 깔끔하게 정리하려 합니다.
- 아이디어를 정리하고 빠르게 적용하여 가능한 많은 실험 및 정리를 하려 합니다. 노션과 **github** 등의 프로그램을 활용하여 다시 봤을 때 구현이 가능하도록 정리하려 합니다.



## 이주용\_T6137

### 1. 나는 내 학습 목표 달성을 위해 무엇을 어떻게 했는가?

- 데이터를 분석하고 전처리 방식에 대해 이해하고, 토큰화 과정에 대해 이해하여 다양한 모델을 사용하면서 앙상블을 통해 대회에서 높은 평가를 얻기 위한 시도들을 해보았다.
- 큰 모델을 사용하면 더 좋은 성능을 보일 것이라는 생각으로 파라미터가 많은 모델들을 이용해 학습해보려는 시도들을 해보았다.

### 2. 나는 어떤 방식으로 모델을 개선했는가?

- 데이터를 분석해 샘플링 방법은 바꾸고 **UnderSampling**을 통해 **label imbalance**한 데이터를 **balance**하게 만들어 학습을 시도해보았다.
- 더 많은 모델들을 통해 비슷한 방식으로 학습을 시도해보았다.

### 3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

- **UnderSampling**을 통해서 **no relation** 과 같은 큰 값을 가지는 데이터의 양을 줄였는데 막상 **Confusion matrix**를 통해 확인해보니 **no relation**을 더 잘 못맞추는걸로 보였다.
- 맞추기 쉬운 데이터를 학습시키는 것보다 맞추기 어려운 애매한 **label**들을 학습시키는 것이 더 중요하고, 단순히 **label**을 통해 데이터의 밸런스를 생각하기 보다 어떤 부분이 정답률이 낮은지 확인해보고, 전처리 과정을 더 잘 수행하는게 중요하다는 깨달음을 얻었다.

### 4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

- 전보다 데이터 분석에 공을 들여서 과제를 수행했고, **Fine-tuning**을 통해 기 학습된 모델을 학습시키는 경우 이전에 **pretrain** 방식에 대해 고려하여 **input**을 만드는 것이 성능 개선에 영향을 준다는 것을 알수 있었다.

### 5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- **5B** 이상의 거대모델들을 사용해 학습을 시도해봤는데, 하드웨어상의 한계로 인해 실제 학습에는 제한이 있어 아쉬웠다.
- 생각보다 큰 모델의 성능이 기대했던 만큼 좋지 않았고, 이를 개선하기 위해서 다양한 실험을 해보기에는 모델의 크기가 크고 시간은 제한되어 있어서 시도를 많이 못해본 점이 아쉽다.
- 이를 개선하기 위해 해당 모델을 어떻게 사용해야 더 좋은 성과를 낼지 고민해 봐야 하는데 그런 시간이 적었던 것이 아쉬웠다.
- 깃 정리와 작업 내용을 정리하는게 개인적으로 미흡했었다.

### 6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- 모델의 특징을 더 많이 알아보고 이해하여 어떻게 사용하는것이 가장 효과적일지 고민해 봐야겠다.
- 데이터 셋 분할을 통해 모델의 추론 결과와 **Ground truth**를 비교하여 무엇이 문제점이고 어떻게 개선할지에 대해 보다 빠르게 확인하고 고민해보는게 성능 개선에 도움이 되었다. 같은 방식을 다른 프로젝트에서도 잘 도입해서 사용하면 좋겠다.
- 좀 더 작업 내용을 잘 정리해 문서화하여 다음 작업이나 타인이 이용할수 있게 만들어 보려는 노력을 해야겠다.

## 정종관\_T6157

나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

- 이번 대회에선 모델보단 데이터에 집중하였음
- 데이터 분석을 통해 **input-output** 관계를 추론하여, 추론 후보정 알고리즘을 고안해냄. 이를 통해 모델 성능 개선.

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

- 모델에 좋은 밥을 먹이는 것이 중요하다. 데이터 클리닝, 데이터 전처리가 모델 성능 개선에 중요하다.
- 데이터를 유심히 들여다보다 보면 인사이트를 얻을 수 있다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 프로젝트 모듈화 및 협업 프로세스 구축 실패
- 대회 주제관련 리서치 부족

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- 프로젝트 시작과 동시에 **task**에 대한 이해를 먼저 하는 것이 중요하다고 생각. 관련 논문을 먼저 읽을 것.
- 프로젝트 모듈화, 깃헙 레포 생성을 통해 협업 프로세스를 우선 구축할 것.

## 정지영\_T6158

1. 나는 내 학습 목표 달성을 위해 무엇을 어떻게 했는가?
  - 베이스라인 코드의 구조와 모델의 원리를 제대로 이해하기 위해 다른 팀원들의 코드에 의존하지 않고 **End-to-End** 로 먼저 모든 학습 과정을 진행해보았다.
  - 이를 통해 **Pandas, PyTorch, Transformers** 에 대한 이해도를 높일 수 있었다.
  - **Github**에 코드를 업로드하기 전에 최대한 코드를 잘 정리해보려고 하였고, 이 과정을 통해 학습 코드를 체계적으로 모듈화 할 수 있게 되었다.
2. 나는 어떤 방식으로 모델을 개선했는가?
  - 모델이 **Subject Entity**와 **Object Entity** 간의 관계를 더 잘 파악할 수 있도록 **Type Token**을 **Special Token** 활용하도록 전처리함으로써 모델의 성능을 개선하였다.
  - **Semantic Typing**을 통해 **Input Data**의 앞부분을 자연어 **Query** 형식으로 바꾸어줌으로써 모델의 성능 향상을 이루어냈다.
  - **Soft Voting, Hard Voting, Weighted Hard Voting** 등의 앙상블 기법을 구현하였다.
3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?
  - 모델의 성능을 개선하기 위한 실험도 중요하지만, 더 중요한 것은 데이터를 꼼꼼히 살펴보고 전처리하여 좋은 데이터를 모델에 넣어주는 것이라는 깨달음을 얻었다.
  - 모든 가설은 다른 조건을 동일하게 통제하여 차근차근 검증해나가야 더욱 효율적인 것임을 알게 되었다.
4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?
  - **EDA** 과정에만 3일 정도의 시간을 투자하여, 데이터를 보다 더 철저하게 살펴보기 위해 노력하였다.
  - 이 과정에서 데이터를 처리하는 것에 익숙해져서, 프로젝트 과정 내내 데이터 처리에 대한 다양한 실험을 수월하게 진행할 수 있었다.
5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?
  - **Validation Data**를 샘플링하는 과정에서 깊은 고민 없이 **label** 별로 비율을 맞춰 샘플링하였는데, 이것이 결국 학습한 모델의 성능을 제대로 평가하지 못하는 이유가 되었다.
  - 데이터에 있는 **Source** 정보를 **Special Token**으로 활용해보았으나, 유의미한 성능의 향상은 이루어지지 않았다.
  - **Config** 파일을 처음부터 잘 정리해놓고 시작하지 않아서, 프로젝트 중반에 많은 실험을 수행해야 하는 기간임에도 효율적인 실험 수행을 하지 못하였다.
  - 전체적인 프로젝트 **Timeline** 및 업무 순서를 정해놓지 않아서, 프로젝트 초반에 했어야 하는 실험이 후반으로 밀리다 보니 제대로 수행하지 못한 부분이 있다.
  - 각자 로컬 환경에서 실험하는 형태로 진행하다 보니, 프로젝트 중간에 공통된 **baseline code**로 합치는 타이밍을 놓쳐 **Github** 사용을 거의 하지 못했다.
6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?
  - 프로젝트 시작 전에 팀원들을 독려하여, 조금 더 체계적인 타임라인을 정해놓고 협업하기
  - **Validation data sampling**을 또 해야 할 경우, 이것이 모델의 약점을 엄밀히 검증할 수 있는 **validation data**인지에 대해 충분히 고민하고 **sampling** 하기
  - 대회 초반에 시간 투자를 조금 많이 해서라도 공통의 **baseline code**를 정하고 실험해보기
  - **Checkpoint**를 활용해 모델을 이어서 학습하는 시도해보기

## 양서현\_T6099

### - 학습 목표 달성을 위해 무엇을 어떻게 했는가?

프로젝트 주제와 주어진 베이스라인 코드의 이해에 중점을 두고 시작했다. 그 이후에 KLUE 강의의 학습과정을 따라 프로젝트를 진행하려 노력했다. 그 중에서도 데이터의 특성에 대해 이해하고, 데이터를 학습에 용이하게 정제하려 노력하였다. 특히, 강의 실습에서의 **Special token** 추가 방법론에 착안해 논문을 참고하여 직접 input의 전처리 구현 작업에 집중적으로 참여하였다.

### - 어떤 방식으로 모델을 개선했는가?

1. 데이터 측면 : 전처리를 통해 다양한 input 형식을 직접 구현해 모델 성능 향상에 기여하였다.
  - 데이터 전처리(Special Token, Semantic typing 등)
    - A. Entity Marker : word → [s:type]
    - B. Typed Entity Marker with Punctuation : word → #^type^word#
    - C. Semantic Typing : subject word와 object word의 관계 + B. sentence
2. 모델 측면
  - W&B Sweep 을 적용한 베이스라인 코드 작성 및 공유
  - Confusion Matrix를 이용한 이상치 분석 및 개선방안 도출
3. 프로젝트 초기 계획 수립
  - 프로젝트 초기 가능한 업무 정리 및 이후 계획 수립

### - 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

성능 향상에 도움이 될 것이라 생각했지만 실제 성능이 비슷하거나 떨어지는 결과가 나와 채택되지 못한 가설들이 많았다.

1. 데이터 클리닝(한자 및 특수 문자 제거)

Entity word 자체에 포함 되어있는 단어를 제외한 sentence 내 모든 한자와 특수 문자를 제거하면 성능 향상에 도움이 될 것이라 예상하였다. 최종적으로 [UNK] token을 포함된 문장을 총 171개 까지 줄일 수 있으나, 오히려 성능이 떨어지는 결과가 나와 채택하지 않았다. 그 이유는 이들을 제거함으로써 오히려 단어의 의미가 소실되거나 왜곡되었기 때문이라고 생각했다.
2. Label Smoothing 실험

주어진 데이터셋에 no\_realtion label이 많아 불균형 해소를 위해 구현해봤지만 기존 Cross-Entropy과 비슷하거나 팀원이 구현한 Focal Loss 보다 성능이 떨어져 채택되지 않았다.
3. 데이터셋 분할

여러가지 요인들을 조정해봐도 오르지 않은 성능이 팀원이 나눈 데이터셋으로 학습하니 무려 1.5점이나 올랐다.

  - 기존: Label 별 train\_test\_split 7:3
  - 변경: [Subject\_type, Object\_type] pair 별로 validation 데이터셋 추출

### - 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

1. 모델의 학습 방법도 중요하지만 데이터가 가장 중요하다는 것을 느꼈다. 다음 기회가 있다면 더 많은 방법과 수단을 동반하여 데이터와 관련된 작업들을 많이 진행할 것 같다.
2. 실험 전 실험 이유와 가설을 명확하게 설정 후 실험을 수행하고, 실험의 버전 관리에 신경써 정리할 수 있도록 할 것이다.