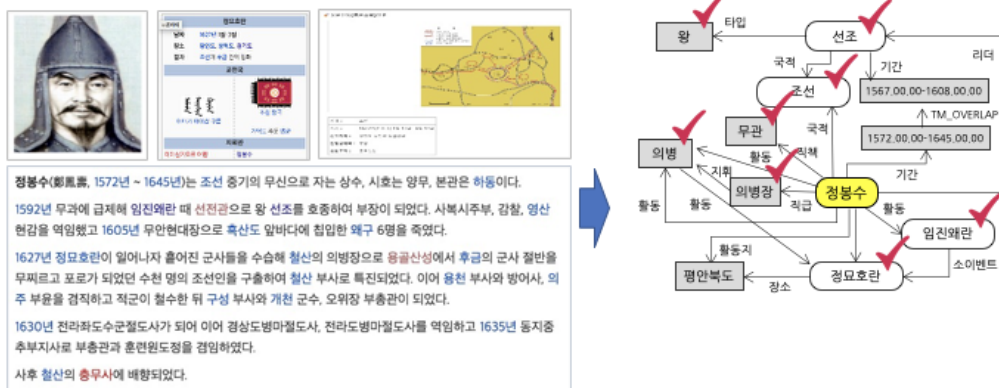




프로젝트 Wrap UP

프로젝트 개요



이 사람은 임진왜란 때 선전관으로 왕을 모셨고, 정묘호란 때는 의병을 일으켜 백성 수천 명을 구출했다. 철산의 용골산성에서 후금을 격퇴하는 전과를 세운 이 사람은 누구일까?

→ 정몽수

관계 추출(Relation Extraction)은 문장의 단어(Entity)에 대한 속성과 관계를 예측하는 문제입니다. 관계 추출은 지식 그래프 구축을 위한 핵심 구성 요소로, 구조화된 검색, 감정 분석, 질문 답변하기, 요약과 같은 자연어처리 응용 프로그램에서 중요합니다. 비구조적인 자연어 문장에서 구조적인 triplet을 추출해 정보를 요약하고, 중요한 성분을 핵심적으로 파악할 수 있습니다.

프로젝트 팀 구성 및 역할

- 김한성: 모델 팀 / 대회 주제 관련 SOTA모델 탐색 및 구현
- 최동민: 모델 팀 / 대회 주제 관련 SOTA모델 탐색 및 구현
- 염성현: 데이터 팀 / 대회 주제 관련 데이터 탐색 및 효율화 + 베이스라인 개선
- 이재욱: 데이터 팀 / 대회 주제 관련 데이터 탐색 및 효율화 + 베이스라인 개선
- 홍인희: 데이터 팀 / 대회 주제 관련 데이터 탐색 및 효율화 + 베이스라인 개선

프로젝트 수행 절차 및 방법

데이터 분석 및 전처리, 증강

EDA

- [UNK] token
 - [UNK] token 출몰 시, 원 문장 의미 소실 문제 발생, [UNK] token 분포 확인, 아래는 예시

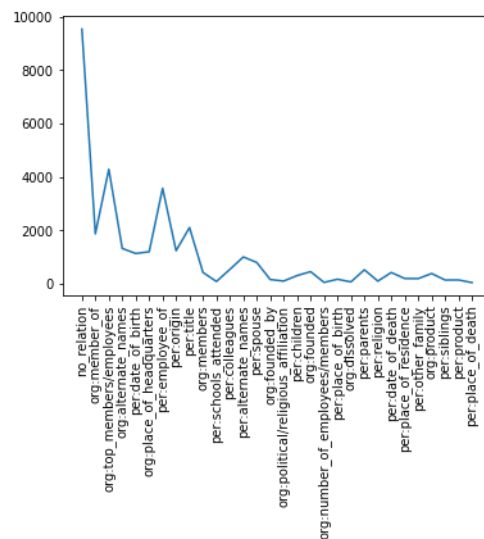
KBS 전주방송총국(KBS 全州放送總局)은 전라북도 지역을 대상으로 하는 한국방송공사의 지역 방송 총국이다.
 {'word': 'KBS 전주방송총국', 'start_idx': 0, 'end_idx': 9, 'type': 'ORG'}
 {'word': 'KBS 全州放送總局', 'start_idx': 11, 'end_idx': 20, 'type': 'ORG'}
 org:alternate_names
 =====
 ['KBS', '전주', '##방송', '##총', '##국', '(', 'KBS', '全', '州', '[UNK]', '[UNK]', '[UNK]', '[UNK]', ')', '은', '전라북도', '지역', '##을', '대상', '##으로', '하', '##는', '한국', '##방송',

- [UNK] token이 발생하지 않는 multilingual model 사용, 해결 방법 모색, 성능 향상 관측

	description	hyperparameter	valid_micro_f1 / auprc	leader_micro_f1 / auprc
15	baseline + xlm-roberta-base	동일	81.877 / 72.160	58.6925 / 57.3416
7	baseline + klue/roberta-base	동일	83.1924 / 72.9139	63.5838 / 60.3213
	xlm + klue		inference	63.6557 / 60.4378

• label, Object_entity, Subject_entity

- Label 분포의 imbalance 문제 확인, 추후 데이터 증강과 Focal loss 활용으로 해결 시도



전처리

- 중복 문장 처리 : 중복 문장 84개 존재, 제거 조치

22090	1959-60 시즌 이후의 알파 에트니카 / 수페르리가 엘라다, 그리스 컵, 그리고...	{'word': '수페르리가 엘라다', 'start_idx': 25, 'end_idx': 38...}	{'word': '그리스', 'start_idx': 36, 'end_idx': 38...}	no_relation	wikipedia
9618	1959-60 시즌 이후의 알파 에트니카 / 수페르리가 엘라다, 그리스 컵, 그리고...	{'word': '수페르리가 엘라다', 'start_idx': 25, 'end_idx': 38...}	{'word': '그리스', 'start_idx': 36, 'end_idx': 38...}	no_relation	wikipedia
25368	2001년 4월 민주국민당 김윤환 의원과 민주당 사이의 정책 연합과 관련 현...	{'word': '민주국민당', 'start_idx': 9, 'end_idx': 1...}	{'word': '민주당', 'start_idx': 23, 'end_idx': 25...}	no_relation	wikipedia
2500	2001년 4월 민주국민당 김윤환 의원과 민주당 의원 사이의 정책 연합과 관련 현...	{'word': '민주국민당', 'start_idx': 9, 'end_idx': 1...}	{'word': '민주당', 'start_idx': 23, 'end_idx': 25...}	no_relation	wikipedia

- 잘못된 Relation Label 처리, 아래는 예시

19531
 페르시아 제국의 다리우스 3세가 알렉산드로스 대왕에게 패하자 그를 살해한 박트리아의 총독 베수스는 민족적 저항을 조직하려고 하였다.
 {'word': '다리우스 3세', 'start_idx': 9, 'end_idx': 15, 'type': 'PER'}
 {'word': '박트리아', 'start_idx': 41, 'end_idx': 44, 'type': 'LOC'}
 per:date_of_death

- Labeling이 잘못된 데이터를 관측하여 수정 작업을 진행, 약 200여개의 데이터 수정

Translation & Augmentation

- **Back translation**
 - Backtranslation을 사용하여 Data Imbalance를 해결하고자 시도
- **papago 언어 감지 & 언어 번역을 통한 augmentation**

```
print(enko(test_con)) # 영어 -> 한국어
print(koen(test_con)) # 한국어 -> 영어
```

김동성(金東聖, 1980년 2월 9일 ~)은 대한민국의 쇼트트랙 선수이다.
Kim Dong-sung (born February 9, 1980) is a South Korean short track speed skater.

- BackTranslation 을 진행 중 Subject, Object에 해당하는 단어가 소실되는 경우를 발견
- 단어의 국적을 확인한 후 번역한다면 단어 소실 문제를 해결할 것이라 기대
- papago API 언어 감지를 사용 후 해당 단어를 한국어로 번역하고 전체 데이터에 추가

원본 문장 : 김동성(金東聖, 1980년 2월 9일 ~)은 대한민국의 쇼트트랙 선수이다.
金東聖 김동성 {"langCode": "ja"}

변경된 문장 : 김동성(김동성, 1980년 2월 9일 ~)은 대한민국의 쇼트트랙 선수이다.

• 데이터 증강 except no_relation, alternate names

◦ 관측 및 실험

- 번역된 데이터 6,506개 중, no_relation 1,541개, alternate_names 906개
- no_relation 을 제외하고 데이터 추가
 - Data Augmentation 진행 시 Data Imbalance 유발
- no_relation + alternate_names 를 제외하고 데이터 추가
 - alternate_names 에 해당하는 Data 번역 시 똑같은 단어가 나오는 경우가 많았음

◦ 결과

- no_relation만 제외했을 시 Baseline 성능보다 Micro F1 Score 1.05 감소
- no_relation, alternate_names 모두 제외 시 Micro F1 Score 0.11, AUPRC 2.06 상승

Noise Data Experiment

- 가설 : no relation label의 noise 데이터를 추가, 학습을 어렵게 만들어 성능 향상 기대
- 실현 : entity 단어를 무작위 shuffle 한 후, 원 문장의 단어와 치환
 - 원 데이터와 subj_entity 치환 데이터

train.head(10)

id	id	sentence	subject_entity	object_entity	label	source
0	0	<Something>는 조지 해리슨이 쓰고 비틀즈가 1969년 앨범 <Abbey R...	{'word': '비틀즈', 'start_idx': 24, 'end_idx': 26...	{'word': '조지 해리슨', 'start_idx': 13, 'end_idx': ...	no_relation	wikipedia
1	1	호남이 기반인 바른미래당-대안신당-민주평화당이 우여곡절 끝에 합당해 민생당(가칭)으...	{'word': '민주평화당', 'start_idx': 19, 'end_idx': ...	{'word': '대안신당', 'start_idx': 14, 'end_idx': 1...	no_relation	wikitree
2	2	K리그2에서 성적 1위를 달리고 있는 광주FC는 지난 26일 한국 프로축구연맹으로부터...	{'word': '광주FC', 'start_idx': 21, 'end_idx': 2...	{'word': '한국프로축구연맹', 'start_idx': 34, 'end_idx': ...	org:member_of	wikitree
3	3	균일가 생활용품점 (주)아성다이스(대표 박정부)는 코로나19 바이러스로 어려움을 겪...	{'word': '아성다이스', 'start_idx': 13, 'end_idx': ...	{'word': '박정부', 'start_idx': 22, 'end_idx': 24...	org:top_members/employees	wikitree
4	4	1967년 프로 야구 드래프트 1순위로 요미우리 자이언츠에게 입단하면서 등번호는 8...	{'word': '요미우리 자이언츠', 'start_idx': 22, 'end_idx': ...	{'word': '1967', 'start_idx': 0, 'end_idx': 3...	no_relation	wikipedia
5	5	: 유엔, 유럽 의회, 북대서양 조약 기구 (NATO), 국제이주기구, 세계 보건 ...	{'word': '북대서양 조약 기구', 'start_idx': 13, 'end_idx': ...	{'word': 'NATO', 'start_idx': 25, 'end_idx': 2...	org:alternate_names	wikipedia

chs_train							Python
id		sentence	subject_entity	object_entity	label	source	
0	0	<Something>는 비틀즈이 쓰고 베이징가 1969년 앨범 <Abbey Road...>	{'word': '베이징', 'start_idx': 21, 'end_idx': 23...	{'word': '비틀즈', 'start_idx': 13, 'end_idx': 15...	no_relation	wikipedia	
1	1	호남이 기반인 바른미래당-민주평화당-전남체육회가 우여곡절 끝에 합당해 민생당(가칭)...	{'word': '전남체육회', 'start_idx': 20, 'end_idx': ...}	{'word': '민주평화당', 'start_idx': 14, 'end_idx': ...}	no_relation	wikitree	
2	2	K리그2에서 성적 1위를 달리고 있는 국제 표준화 기구는 지난 26일 광주FC으로부...	{'word': '국제 표준화 기구', 'start_idx': 21, 'end_idx': ...}	{'word': '광주FC', 'start_idx': 39, 'end_idx': 4...	no_relation	wikitree	
3	3	균일가 생활용품점 (주)대한민국(대표 박정부)는 코로나19 바이라스로 어려움을 겪고...	{'word': '대한민국', 'start_idx': 13, 'end_idx': 1...}	{'word': '아성다이소', 'start_idx': 21, 'end_idx': ...}	no_relation	wikitree	
4	4	요미우리 자이언츠년 프로 야구 드래프트 1순위로 서울공향에게 입단하면서 등번호는 8...	{'word': '서울공향', 'start_idx': 27, 'end_idx': 3...}	{'word': '요미우리 자이언츠', 'start_idx': 0, 'end_idx': ...}	no_relation	wikipedia	

- 결과 : 성능 비교 - BaseLine : klue/roberta-small
 - 베이스 라인에 노이즈 데이터 증강 방식을 달리하며 실험

모델명	valid score	leaderboard score
BaseLine	81.5200 / 71.2845	57.8320 / 52.9398
BaseLine + noise aug (subj+obj)	78.1988 / 67.1051	54.3647 / 43.6026
BaseLine + noise aug (subj)	79.82438 / 70.56869	55.8859 / 45.3100
BaseLine + noise aug (subj / obj)	79.23778 / 69.89351	55.2676 / 50.1394
BaseLine + only noise aug data (subj / obj)	66.83684 / 52.79786	34.3742 / 33.0628

- 결과 해석 - 성능 하락
 - no_relation label의 noise 데이터를 추가한 것이 label imbalance를 심화시킨 것으로 보임
 - test set 7764개 중 no_relation 예측이 4578개로 기존 4057개 보다 많아진 모습을 보였음

입력 데이터 구성

IRE

목적

1. Entity에 포함되어 있는 NER type 사용
2. PLM의 Knowledge 활용

구현

- marker의 기법을 활용, input 구성을 다르게 하며 실험 진행

Method	Input Example	BERT _{BASE}	BERT _{LARGE}	RoBERTa _{LARGE}
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

- 영문 type marker가 잘리는 문제를 발견하여 한글로 치환하여 적용, 아래는 예시

@*사람*마크 주커버그@는 #^기관^페이스북#을 설립했다.

결과

1	BaseLine + 단일 문장 분류	82.6644 / 73.0586	klue/roberta-small
20	BaseLine + Entity mask	78.174 / 64.965	klue/roberta-small
21	BaseLine + TEM_prunct	83.254 / 74.068	klue/roberta-small

- TEM_prunct marker 적용 시, F1-score기준 0.6%, auprc기준 1%의 성능 향상 관측

모델

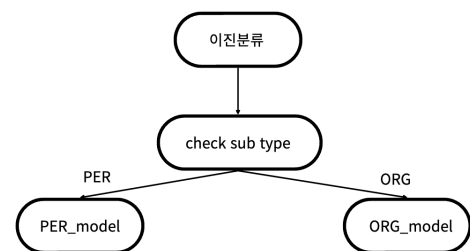
이진 분류 모델

배경 : 분류 대상의 수를 줄이면 모델의 분류 성능을 높일 수 있을 것이라 기대

구현 : 이진 분류로 relation/no_relation 분류 → PER/ORG subject type 모델을 두고 각각의 label 분류

구성한 프로젝트 구조

```
for_binary/
├── binary_train.py - first model
├── train_for_per.py - second model
├── train_for_org.py - last model
├── load_data.py - REDataset setting
├── binary_utils.py - labeling or other utils
├── utils_for_per.py - labeling or other utils
├── utils_for_org.py - labeling or other utils
├── config/ - abstract all config for each model
│   ├── config : for binary
│   ├── config_for_per
│   └── config_for_org
├── data/ 새로 추출한 데이터
│   ├── org_train.csv
│   └── per_train.csv
├── run.sh/ - for all setting
├── inference.py/ - for all setting
└── thanks for comming I'm Yeombora
```



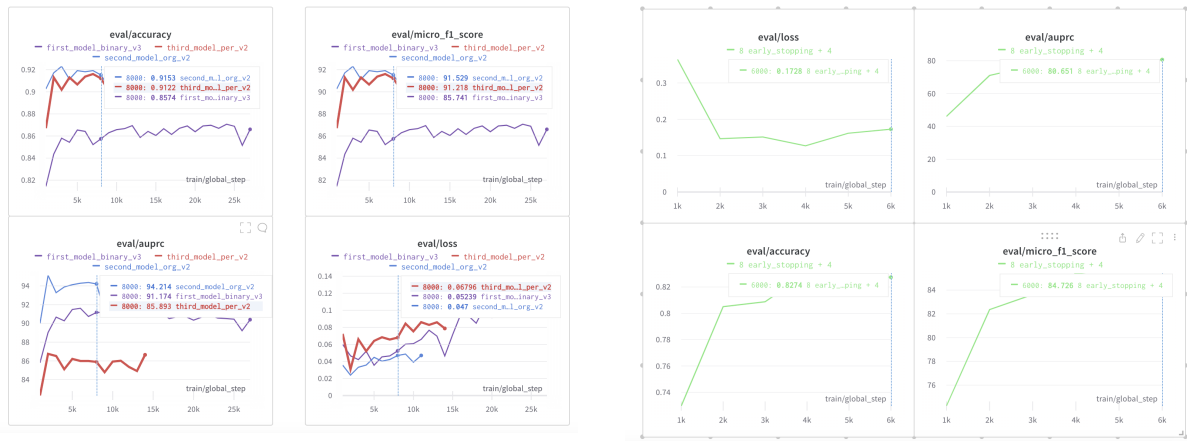
결과

이진분류 모델의 결과

21	Finished	61.5959	56.9665
----	----------	---------	---------

기존 베이스라인의 결과

1	Finished	63.0376	57.8788
---	----------	---------	---------



- Valid 기록 상 성능 향상을 보였으나 Leader board에서는 성능이 하락
- 모델 병합 과정에서 일반화의 문제가 있었을 것으로 추측, 해당 부분에 대한 회고가 필요

RECENT 구현

배경 : Subject-Object Entity Pair 별로 각각 다른 Classifier를 두어 문제를 단순화하는 방법

구현 : Binary (relation/no_relation) 예측 후, Multi-classifier 예측을 덮어쓰는 방식으로 구현

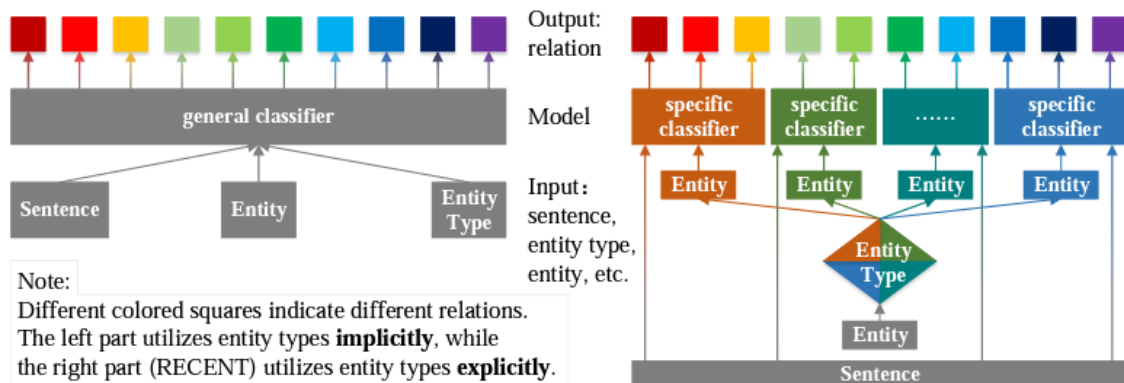


Figure 3: Relation classification with entity type restriction. The left part does not consider the restriction of entity types on relations and only feeds entity types as features into a general classifier. The right part explicitly utilizes entity types to restrict candidate relations and learns a specific classifier for each pair of entity types.

- 이진 분류 방식과 라벨 분류기 구성 방식을 다르게 하며 실험
 - BaseLine Classifier / Binary Classifier(CE Loss / BCE Loss)
 - Multiple Label-Classifer / Grouped Multiple label Classifier
- 분류기 (상위 3개)

1번 분류기 : ORG_PER		
org:top_members/employees	org:founded_by	org:alternate_names
2번 분류기 : ORG_ORG + ORG_POH		
org:member_of	org:alternate_names	org:product
org:members	org:place_of_headquarters	org:political/religious_affiliation
org:top_members/employees	org:founded_by	

3번 분류기 : PER_PER + PER_POH + PER_ORG		
per:employee_of	per:title	per:alternate_names
per:spouse	per:colleagues	per:parents
per:origin	per:children	per:other_family
per:product	per:siblings	per:religion
per:schools_attended		

결과 - 성능 하락 : 왼쪽은 분류기 적용 결과, 오른쪽은 분류기 미적용 결과

17 **Finished** 63.9481 64.4399 2 **Finished** 65.9701 63.0621

- Binary Classifier 보다는 BaseLine Classifier을 배경 모델로 사용하는 것이 성능이 좋았음. Binary 설정이 Entity-Label 간 관계를 희석시킬 수도 있다고 보여짐
- Multiple Label Classifier 또한 적용 시 성능이 떨어진 것으로 나타났는데 이는 데이터 수가 적은 entity pair의 classifier에 충분한 학습 데이터가 주어지지 않았기 때문으로 보임

R-BERT 구현

배경 : CLS 토큰 뿐만 아니라 각 Entity를 대표하는 토큰 정보 또한 활용, 모델 학습 보조 기대

- CLS토큰, Entity1 토큰, Entity2 토큰의 hidden state를 사용

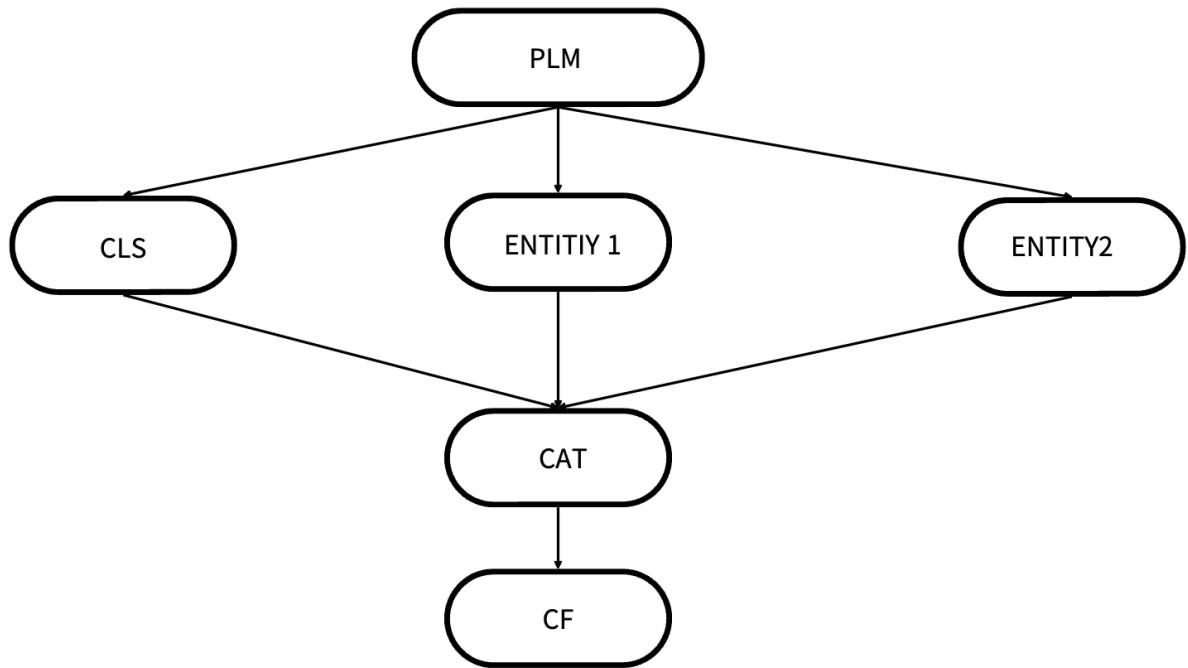
구현

$$\begin{aligned}
 H'_1 &= W_1 \left[\tanh \left(\frac{1}{j-i+1} \sum_{t=i}^j H_t \right) \right] + b_1 \\
 H'_2 &= W_2 \left[\tanh \left(\frac{1}{m-k+1} \sum_{t=k}^m H_t \right) \right] + b_2
 \end{aligned}
 \tag{1}$$

$$H'_0 = W_0 (\tanh(H_0)) + b_0 \tag{2}$$

$$\begin{aligned}
 h'' &= W_3 \left[\text{concat} (H'_0, H'_1, H'_2) \right] + b_3 \\
 p &= \text{softmax}(h'')
 \end{aligned}
 \tag{3}$$

- PLM에서 last hidden state를 가져와 H0 : CLS, H1 : Entity1, H2 : Entity2를 구성



- `add_pooling_layer = False` 를 두어 hidden states를 그대로 가져와 pooler를 새로 구성
- entity의 index를 추출하는 과정을 거쳐 진행

결과

구현한 RBERT 모델				베이스라인 모델			
24	Finished	68.9451	75.0652	1	Finished	63.0376	57.8788

- 새로 구성한 Layers는 parameters가 random initialize 되어있고, PLM의 parameters는 이미 기학습 당시 계속 update되어 왔기 때문에 서로 최적의 learning rate가 다르다고 판단

```

optimizer = optim.AdamW([
    {'params': model.plm.parameters(),
    {'params': model.dense_for_cls.parameters(), 'lr': cfg.train.second_lr},
    {'params': model.dense_for_e1.parameters(), 'lr': cfg.train.second_lr},
    {'params': model.dense_for_e2.parameters(), 'lr': cfg.train.second_lr},
    {'params': model.entity_classifier.parameters(), 'lr': cfg.train.second_lr}
    ], lr=cfg.train.lr, weight_decay=0.01, eps = 1e-8)
scheduler = CosineAnnealingWarmRestarts(optimizer, T_0=cfg.train.T_0, T_mult=cfg.train.T_mult, eta_min=cfg.train.eta_min)
  
```

- `pytorch` 문서를 참고, second_lr config로 parameter 별 학습율을 조정한 후 다시 학습 진행

28번 second_lr : 2.4e-4				초기 RBERT모델			
29번 second_lr : 3e-5				24	Finished	68.9451	75.0652
29	Finished	72.2307	75.5892				
28	Finished	71.8489	75.0746				

- 동일 learning rate 사용 시보다 성능이 향상됨, lr과 second lr 차이가 명확할 수록 성능이 우수

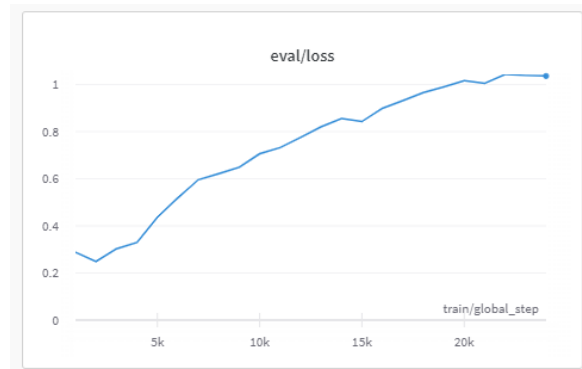
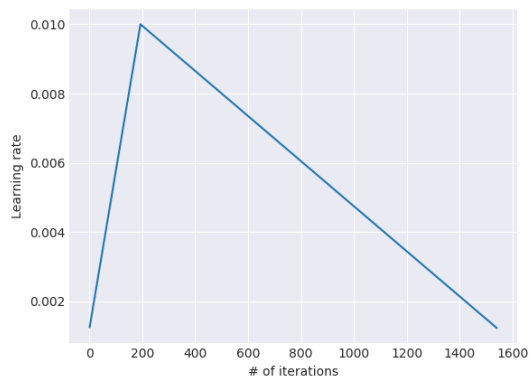
학습 설정

Loss Search

Overfitting에 대한 염려 : 학습 진행에 따라 evaluation loss가 지속 상승

Default Scheduler : **OneCycleLR - linear**

- **OneCycleLR - linear** 는 최대값 이후 learning rate가 지속 감소, over-fitting 문제 원인 판단

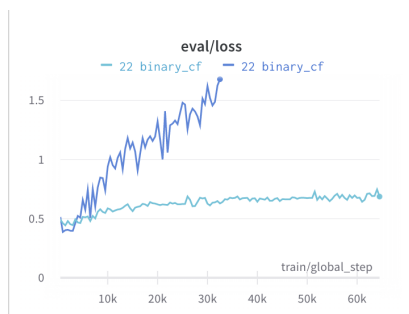
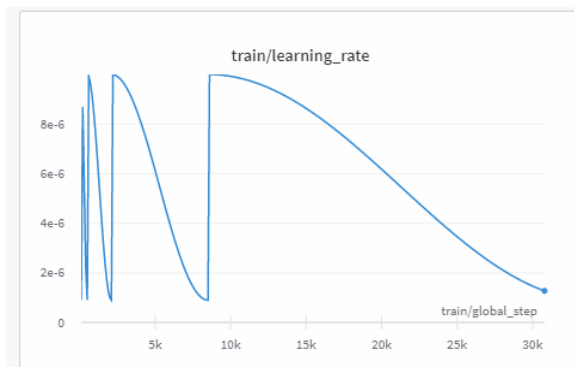


Alternative Scheduler Experiment

CosineAnnealingWarmRestarts with AdamW

- multi-cycle + cosine 사용, 주기를 돌며 learning rate 재조정

적용 결과

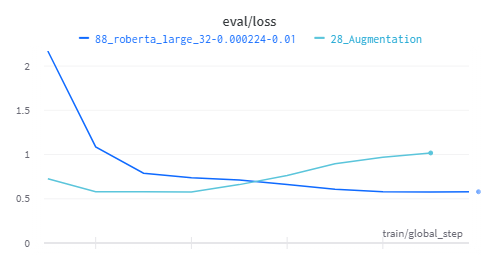
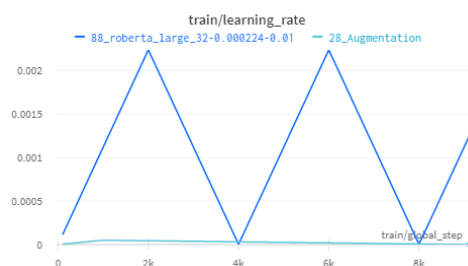


결과

Cyclical Learning Rate with SGD

- 실험 진행 전 최적의 LR을 찾기 위하여 LRFinder 함수를 사용하여 2.24E-04의 값을 찾음
- 해당 LR의 base,max 를 **OneCycleLR - linear** 의 실험의 LR과 비교하여 설정

적용 결과



`CosineAnnealingWarmRestarts` 을 사용한 경우, evaluation loss의 증가폭 줄어든 것을 관측하였음

`Cyclical Learning Rate` 를 사용한 경우, loss가 꾸준히 내려가는 결과를 관측하였음

Focal Loss

가설 : Dataset의 Class imbalance가 심한 만큼, Cross Entropy Loss 대신 Class imbalance dataset을 학습할 때 도움이 되는 Focal loss를 사용한다면, 성능 향상을 도모할 수 있을 것

구현 : class별 개수를 계산하여 Cross-entropy 대신 Focal loss를 사용하도록 Trainer를 custom하였음

결과 : Test data기준으로 **micro-f1 score**는 **0.8%**, **AUPRC**는 **0.7%** 정도 상승하였음

1	Finished	58.0339	55.1663	2	Finished	58.8475	55.8467
---	----------	---------	---------	---	----------	---------	---------

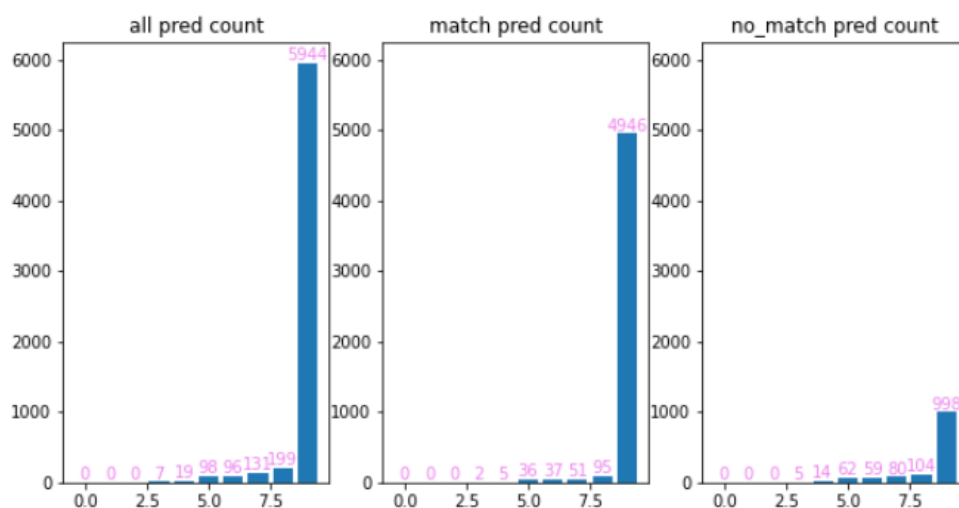
출력 분석 및 후처리

Validation Dataset inference analysis

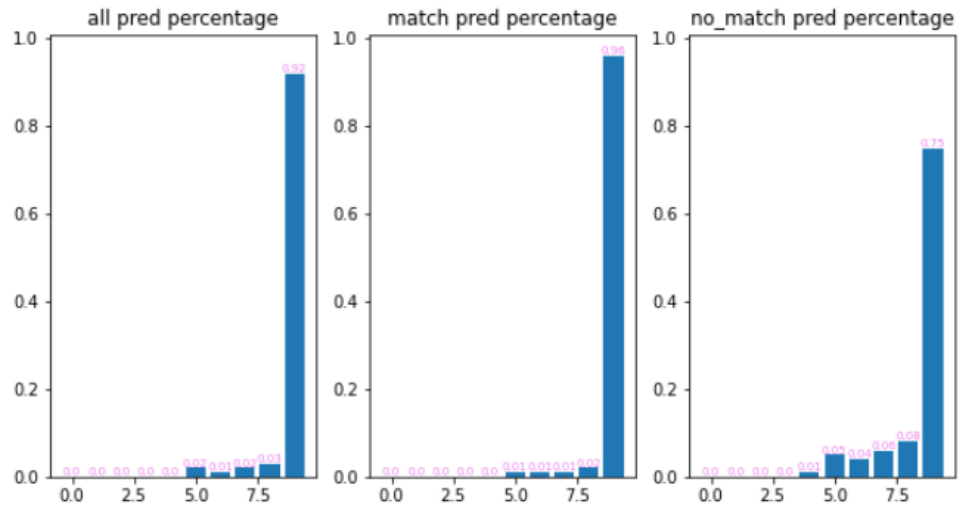
가설 : 모델 추론의 경향성을 관찰해 잘못된 예측을 filtering하는 데 기여할 수 있음

관찰 : validation data 데이터로 추론 작업을 실행하고 그 결과를 분석하였음

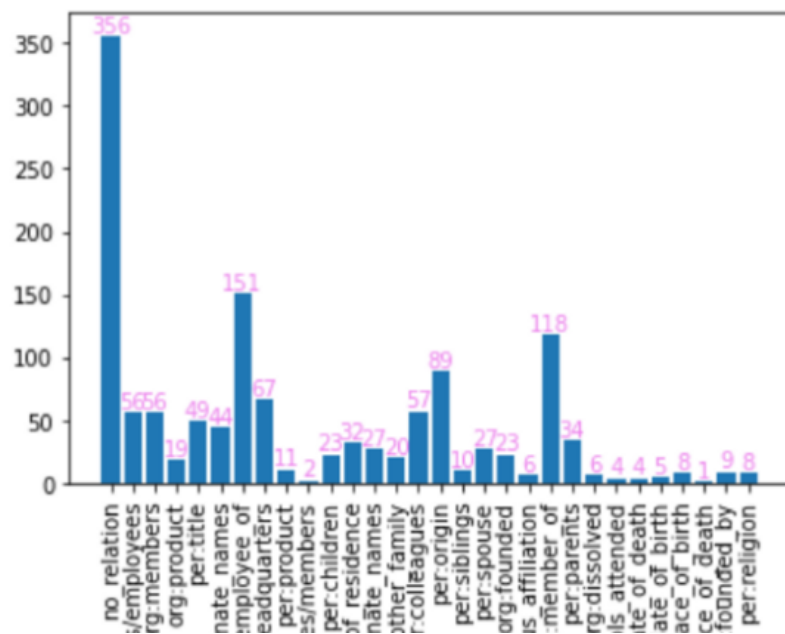
- 예측 확률값의 percentage count 분석, 10% 구간 마다 관측값의 수를 확인



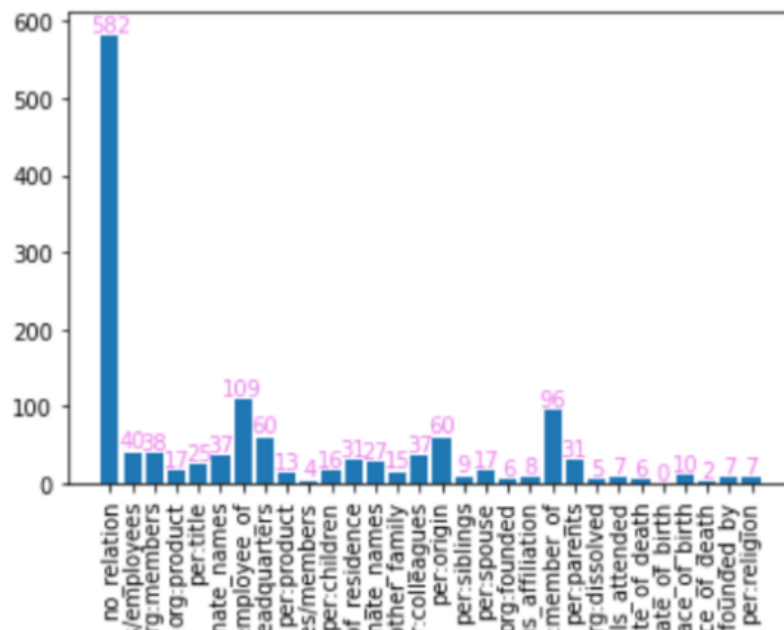
- 예측 확률값의 percentage count 분석, 10% 구간 마다 관측값의 상대 도수를 확인



- 분석 : 맞춘 데이터에서 0.9를 넘는 데이터 비율이 전체 보다 높게 나왔음. 확신을 가지고 맞춘다는 느낌. 하지만 못 맞춘 데이터에 대해서는 그 확신의 정도가 떨어짐.
- 예측이 틀렸을 경우 prediction label(예측) 별 관측값의 수를 확인



- 예측이 틀렸을 경우 gold label(정답) 별 관측값의 수를 확인



- 분석 내용 : Gold Label의 no_relation들이 잘못 예측되는 경우가 상대적으로 매우 많았으며 예측이 상대적으로 많이 틀리는 특정 label들이 존재하였음

실험 : pred_label에 기준점이 되는 threshold를 적용, threshold 아래는 no_relation으로 치환

- no_relation이 아닌 두 번째로 높은 확률값을 가진 index의 label로 치환하는 실험을 추가 진행함

실험 결과 : pred threshold를 0.9로 설정 시 성능 소폭 상승, second threshold 도입 시 성능 하락

label threshold ...

Aa Description	Column 3	pre_thre...	second_...	leaderboard f1
baseLine TwoSen	klue/roberta-small	X	X	58.8269
baseLine TwoSen	klue/roberta-small	0.9	X	58.8921
baseLine TwoSen	klue/roberta-small	0.9	0	56.5469
baseLine TwoSen	klue/roberta-small	0.9	0.1	56.8910

결과 해석 : threshold 설정이 no_relation을 더 맞추게 하였고 F1-score를 높인 것으로 보임

- 하지만 이후 팀 내 가장 score가 좋았던 결과물에 적용했을 때, micro_f1 score가 74.0097에서 72.795로 떨어진 것을 보아 예측 성능이 좋은 모델에서는 threshold 설정이 어렵고 또 성능 향상에 기여하기 어렵다는 것을 발견하였음

CoRE 구현

배경

- 모델이 갖는 bias가 있을 것이라 가정하고 inference 단계에서 해당 bias를 제거하는 방법
 - context를 읽지 않고 entity 정보만으로 label을 추론하는 현상 관측

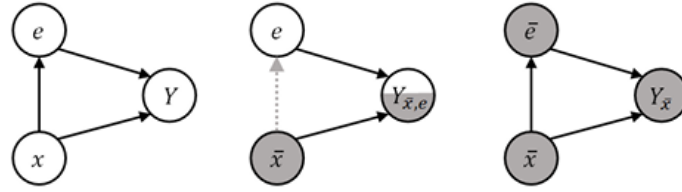


Figure 3: The original causal graph of RE models (left) together with its two counterfactual alternates for the entity bias (middle) and label bias (right). The shading indicates the mask of corresponding variables.

구현

- Entity bias, label bias 정의 (Entity에 대한 편향, 입력을 제하고 모델이 label에 대해 갖는 편향)
- inference시 `Sentence`, `Only Entity`, `input Nothing` 각각의 input으로 추론하도록 구성
- RBERT모델이 entity hidden state를 사용하기에 편향이 존재할 것으로 가정, 해당 모델에 적용

```

lamb_1 = -0.6
lamb_2 = -0.2
def la(a, b, c, d):
    return a+lamb_1*b+lamb_2*c + d #음수발생! label_constraint

new_prob = []
for i in tqdm(range(len(output_prob))):
    tmp = list(map(la, output_prob[i], mask_output_prob[i], mask2_output_prob[i], label_constraint[i]))
    tmp = softmax(np.array(tmp))
    new_prob.append(tmp.tolist())

```

결과 - 기존 모델 대비 성능이 하락함

RBERT + CoRE				기존 RBERT			
27	Finished	67.0410	68.1582	24	Finished	68.9451	75.0652

한계

1. lamb_1, lamb_2에 대한 최적값 찾기 시간 부족
2. 데이터 편향 여부와 그 정도를 확인하는 작업이 필요했으나 진행하지 못하였음

참고 문헌

An Improved Baseline for Sentence-level Relation Extraction (Zhou et al, 2022). URL

<https://doi.org/10.48550/arXiv.2102.01373>

Relation Classification with Entity Type Restriction (Lyu et al, 2021). URL <https://arxiv.org/abs/2105.08393>

Should We Rely on Entity Mentions for Relation Extraction? Debiasing Relation Extraction with Counterfactual Analysis (Wang et al, 2022).

<https://arxiv.org/abs/2205.03784>

Cyclical Learning Rates for Training Neural Networks (Smith, 2017) URL <https://arxiv.org/abs/1506.01186>