

KLUE: Relation Extraction Wrap-up Report

NLP-09 조(역삼동불나방)

1 프로젝트 개요

1. 프로젝트 주제 및 목적

- 관계 추출(Relation Extraction)은 문장의 단어(Entity)에 대한 속성과 관계를 예측하는 NLP Task 로, 비구조적인 자연어 문장에서 구조적인 triple 을 추출해 정보를 요약하고, 중요한 성분을 핵심적으로 파악할 수 있다.
- 본 프로젝트는 주어진 데이터셋을 바탕으로 문장 내 두 단어의 관계를 30 개의 관계 Label 에 대한 예측 확률을 추론하는 모델을 만드는 것에 목적을 둔다.

2. 프로젝트 환경

컴퓨팅 환경	5 인 1 팀, 인당 V100 서버를 VSCode 와 SSH 로 연결하여 사용
협업 환경	Notion, GitHub, WandB, Google Drive
의사 소통	Slack, Zoom, 카카오톡

3. 프로젝트 구조

a. 데이터 수

- 문장 및 단어 정보(Train 32,470 / Test 7,765)

b. 데이터셋 구조

Column	설명
id	샘플 순서 ID
sentence	관계 추출을 위한 단어들을 포함한 문장
subject_entity	Subject Entity 에 대한 정보(단어, 시작 인덱스, 끝 인덱스, 타입)
object_entity	Object Entity 에 대한 정보(단어, 시작 인덱스, 끝 인덱스, 타입)
label	두 Entity 사이의 관계 (30 개의 Label)
source	샘플 출처

c. Entity Type 설명

Type	ORG	PER	DAT	POH	NOH
설명	조직, 단체	사람	시간, 날짜	단체, 사람을 제외한 고유명사	기타 숫자를 포함한 단어

d. Relation Class 설명

Relation Class	Description
<i>no_relation</i>	No relation in between (e_{subj}, e_{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

e. 평가 지표:

- 1) no_relation class 를 제외한 **micro F1 score**
- 2) 모든 class 에 대한 **area under the precision-recall curve (AUPRC)**

f. 대회 중 리더보드 평가 기준: Public Score (Test 의 50%를 바탕으로 평가)

g. 최종 평가 기준: Private Score (Test 의 100%를 바탕으로 평가)

h. 하루 제출 횟수: 10 회

2

프로젝트 팀 구성 및 역할

1. 팀 구성 및 역할

- 전현욱 : 팀 리더, Ensemble 구현, torch 모델 구현, 단일 모델 학습
- 곽수연 : 데이터 전처리 및 증강, 단일 모델 학습
- 김가영 : Entity Tagging 실험, Prompt 실험, 단일 모델 학습
- 김신우 : Rule-based 모델 구현, Entity Tagging 실험, 단일 모델 학습
- 안윤주 : 데이터 전처리 및 증강, 단일 모델 학습

1. 프로젝트 기간

- 2024-01-03 10:00 ~ 2024-01-18 19:00

일	월	화	수	목	금	토
31	1월 1일	2	3	4	5	6
			강의 수강		EDA 및 베이스라인 분석	
7	8	9	10	11	12	13
EDA 및 베이스라인 분석		가설 수립 및 검증				모델 평가
			데이터 증강			
14	15	16	17	18	19	20
모델 평가		앙상블 실험 및 최종 제출				

2. 활용된 기술 및 라이브러리

- 개발 언어: Python
- 데이터 전처리 및 증강: Pandas, Numpy, GoogleTrans
- 모델링: PyTorch, HuggingFace
- 성능 분석: WandB

3. 프로젝트 세부 수행 절차

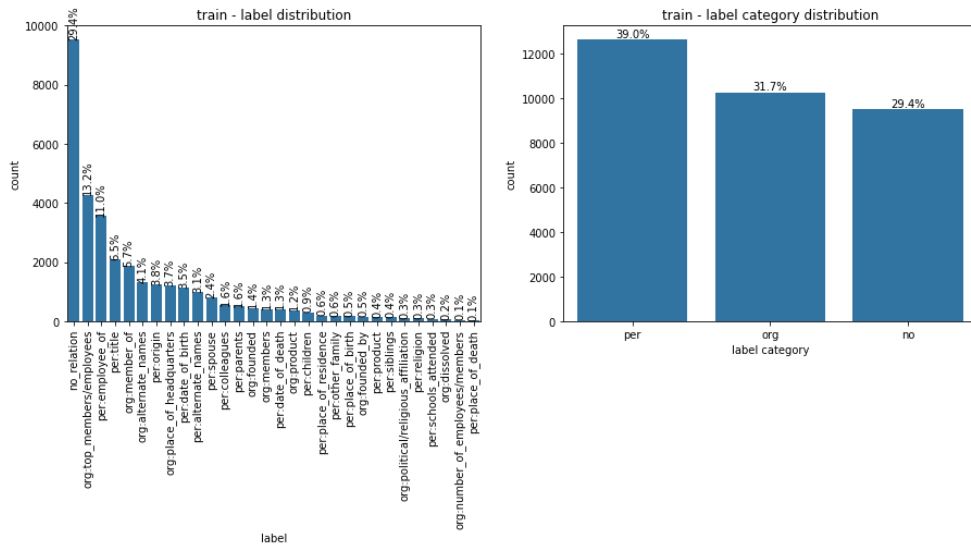
- 1) 2024-01-05 (금) 까지 대회 기초 강의 전부 수강 완료
- 2) 서버 수령 후 GitHub 및 작업 환경 설정
- 3) 대회 개요 및 데이터 성질, 베이스라인 코드 분석
- 4) EDA 진행 후, 분석 내용을 바탕으로 가설 설립
- 5) 모델 선정 및 역할 분담 (데이터 전처리 및 증강 2인, 가설 시험 3인)
- 6) 데이터 전처리 및 증강
- 7) 가설 실험 및 검증 (5번과 동시 진행)
- 8) 검증 결과를 바탕으로 최종 방법 선택 후 모델 학습 및 평가
- 9) 평가한 모델 성능 비교 후 Ensemble 진행
- 10) 최종 결과물 제출

1. 작업 환경 설정

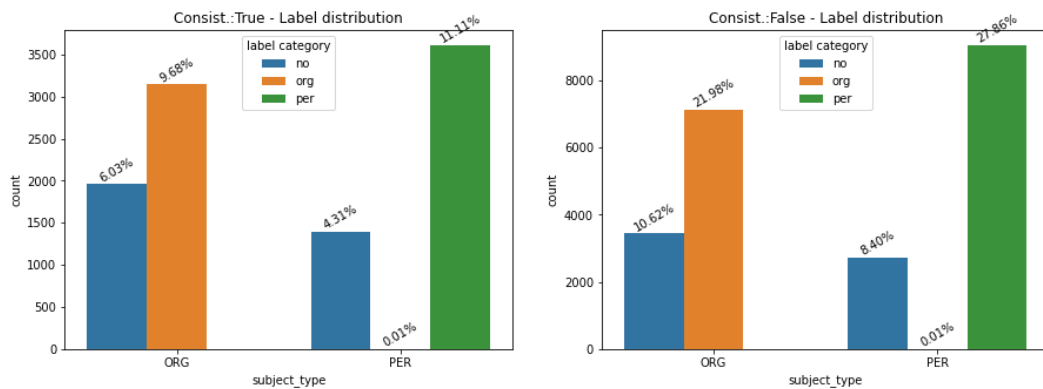
- GitHub : branch 별로 버전 관리
- V100 서버 : 코드 작성 및 모델 학습 (GPU CUDA 버전 : 11.4)
- WandB : 팀 프로젝트 공간에서 모델 성능 분석

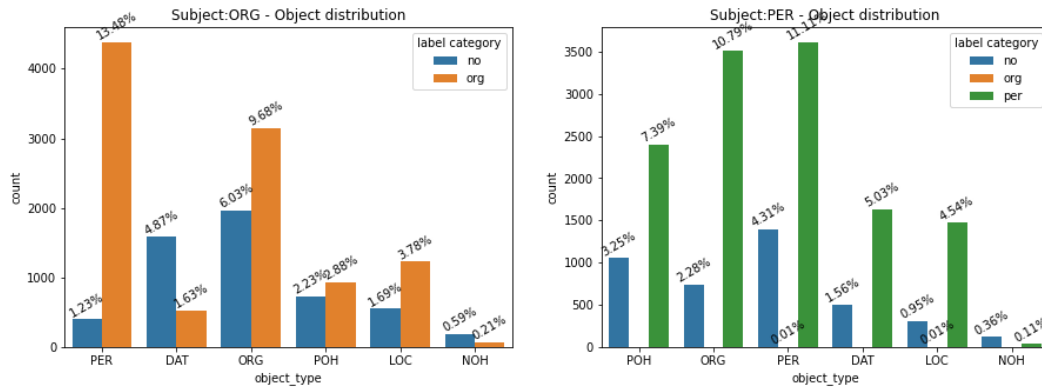
2. EDA

- Label 분포 확인
 - 데이터 불균형(no_relation 이 전체의 30%를 차지)



- Entity Type 과 Label 사이의 관계
 - Subject Type 이 ORG 면 Label 이 **org:**** 또는 **no_relation** 이고, PER 이면 Label 이 **per:**** 또는 **no_relation** 임을 확인





3. 가설 수립 및 검증

- 데이터 증강을 하면, 데이터 불균형이 해소되어 성능이 향상될 것임.
- EDA 를 해본 결과 Subject Entity 의 타입과 Object Entity 의 타입에 따라 메인 카테고리라 세부 카테고리가 달라짐을 확인했으며, Ruled-based 로 접근하면 분류 성능이 향상될 것임.
- 문장에서의 Entity 위치를 나타내는 Positional 정보를 주면 성능이 향상할 것임.
- 여러 Entity Tagging 방법을 활용하면 Entity 의 위치 전달 및 강조하는 효과가 있어 성능이 향상될 것임.

4. 데이터 증강

- 딥러닝 학습을 보다 효과적으로 하기 위해 데이터 증강이 필요하다고 판단 따라서, 아래와 같은 방법으로 시도

1) 한국어 LLM 모델

- 오픈소스를 활용하여 시도할 수 있는 한국어 LLM 모델 7 가지를 실험
- CUDA GPU 버전이 낮아 최신 LLM 모델은 활용 불가

```
RuntimeError: The NVIDIA driver on your system is too old (found version 11040). Please update your GPU driver by downloading and installing a new version from the URL: http://www.nvidia.com/Download/index.aspx Alternatively, go to: https://pytorch.org to install a PyTorch version that has been compiled with your version of the CUDA driver.
```

- 활용 가능한 LLM 모델도 결과가 부적절
- 결국, 최종 결과에는 사용되지 못한 방법론
- 시도한 한국어 LLM 모델 7 가지

★ GPU 버전 문제로 활용하지 못한 모델

1) nlpai-lab/KULLM, 2) beomi/llama-2-ko-7b, 3) beomi/KoAlpaca, 4) krafton-ai/KORani, 5) quantumaikr/KoreanLM, 6) kakaobrain/kogpt

★ 결과가 부적절해 활용하지 못한 모델

1) Polyglot-ko 12.8B(QLoRA) with 4bit, 2) SKT-AI/KoGPT2

2) 역번역(Back Translation)

- Entity 의 "word"를 각각 [sub] 토큰과 [obj] 토큰으로 치환
- 간편하게 사용하기 좋은 GoogleTrans 를 활용해 한국어를 영어로 번역 후, 번역된 문장을 다시 한국어로 번역
- 토큰을 원래 단어로 바꾸고, 번역 문장 중 '다'로 끝나지 않는 문장(명사로 끝나는 문장 등) 제거

예시	Sentence	Subject Word	Object Word	Label
원본 문장	개신교 신학자이자 유니온 신학교 교수로 일하던 라인홀트 니부어가 신학 교수 자리를 마련한 뒤, 초대장을 보냈기 때문이다.	라인홀트 니부어	개신교	per:religion
변경 문장	개신교 신학자 및 연합 신학교 교수로 일한 라인홀트 니부어 이후 그는 신학교 교수를 설립 한 후 초대장을 보냈습니다.	라인홀트 니부어	개신교	per:religion

5. 모델 선정

- Baseline 기준으로 성능이 높은 모델 위주로 선정
 - klue/roberta-large
 - monologg/koelectra-base-v3-discriminator
 - BM-K/KoDiffCSE-RoBERTa
 - nlpotato/roberta_large-ssm_wiki_e2-origin_added_korquad_e5
 - xlm-roberta-large
 - soddokayo/klue-roberta-large-klue-ner
 - sdadas/xlm-roberta-large-twitter
 - severinsimmler/xlm-roberta-longformer-large-16384

6. 모델 학습 및 성능

- **Train Sampling**
 - 한 모델이 학습하는 시간이 데이터의 양에 따라 길어지기 때문에 전체 데이터에서 약 5%만 샘플링해서 성능 평가를 진행
 - 여러 시드 값(42, 109, 777)을 기준으로 랜덤하게 추출한 데이터들을 바탕으로 성능 평가 진행
- **Train Valid Split**
 - train 데이터만 주어져 있었기 때문에 과적합 방지 및 최적의 모델 hyperparameter 를 찾기 위해 8:2 비율로 분할

- **Rule-based**

- train 데이터를 Subject Entity의 타입(PER/ORG)으로 나눠서 각각 학습시킨 두 개의 모델로 만들어, test 데이터도 각 모델 별로 예측한 결과를 병합
 - 1) 데이터 csv 파일을 PER, ORG로 분리해서 각각 단일 모델로 학습 및 예측
 - ➔ 기존 모델보다 F1 score 기준 **약 14점 성능 향상**
 - 2) 데이터 csv 파일을 분리하지 않고 baseline 코드에서 불러온 데이터를 Rule-based에 맞게 분리해서 각각 학습 및 예측.
 - ➔ 기존 모델보다 F1 score 기준 **약 7점 성능 하락**
- 동일한 방법론을 적용시켜 코드화했을 때, 성능 하락
 - ➔ 분석 결과 원인을 찾지 못하였고, 추가적인 노력을 들이는 것보다 다른 방법론에 시간을 더 투자하는 것이 낫다는 판단 하에 **방법론 폐기 결정**

- **Entity tagging**

- 다양한 Entity Tagging 방식을 실험해서 제일 성능이 높은 방법 선택
 - 1) Typed Entity Marker
 - 각 Entity의 위치와 특성을 식별할 수 있게 Entity를 $[S(O) + type]/[S(O) + type]$ 으로 감싸주었음
 - **ex) [SORG]비틀즈[/SORG], [OPER]조지 해리슨[/OPER]**
 - 2) Typed Entity Marker Punctuation Version
 - Entity와 Entity Type에 강조의 효과를 주기 위한 문장부호 사용
 - Entity의 Subject는 @, Object는 #을 사용하였으며, Entity Type의 경우 Subject는 ~, Object는 ^로 추가
 - Entity Type의 경우엔 스페셜 토큰으로 지정
 - **ex) @ ~ [sub_type] ~ SUB_WORD @ ... # ^ [obj_type] ^ OBJ_WORD # ...**
 - 3) Baseline + Typed entity token + Subject/Entity Positional Token
 - Subject, Object Entity의 위치를 나타내기 위해 문장 내 Entity Word를 [sbj], [obj] 토큰으로 감싸주었음
 - Baseline의 input 방식을 변형해서 사용
 - [sep] 앞의 문장은 Entity type 토큰으로 Entity Type 정보를 제공
 - **ex) [CLS] [per] 손병희 [SEP] [org] 천도교 [SEP] 러일 전쟁 때 ... [sbj] 손병희 [/sbj] 가 동학의 ... [obj] 천도교 [/obj] 를 포교 ... [SEP]**
- 성능 비교 표

방법론	Micro F1	AUPRC
Baseline	67.43	46.21
1)	69.95	46.19
2)	67.17	42.73
3)	69.99	47.05

- **LSTM Layer for Fine-tuning**
 - 기존 Baseline 에서 사용된 Fine Tuning Linear Layer 를 사용하지 않고 LSTM layer 로 변형해서 logits 추출
 - 기존 Linear layer 보다 micro F1 score 기준 약 3 점 성능 향상
- **Weighted Cross Entropy Loss**
 - 데이터의 불균형성이 심했기 때문에 비율이 높은 label 은 학습을 덜 하도록 설정
 - 학습하는 Train 데이터의 비율을 참고해서 Weight 값 설정
- **HuggingFace 를 PyTorch 로 활용**
 - Baseline 코드는 HuggingFace Trainer 클래스를 사용했으나 Custom model, loss, optimizer, wandb 등 Customizing 의 어려움 존재
 - PyTorch Lightning 을 사용하려 했으나, 서버 환경에서 GPU 드라이버 버전 호환 문제로 인해 사용 불가
 - Torch Training Module 을 사용해 Pytorch-Lightning 및 HuggingFace Trainer 와 유사하게 코드를 구현
 - ➔ 큰 버전 이슈 없이 다양한 Customizing 을 적용
- **모델 학습 방법론**
 - 학습에 더 많은 데이터를 활용하기 위해 Train 데이터를 모두 사용
 - 우선적으로 각 모델 성능 평가를 위해 Train / Valid 로 나눠서 진행
 - 각 모델별 적절한 학습이 이루어지는 순간이 평균적으로 5 Epoch
 - 최종적으로 Ensemble 실험에 사용한 모델은 모든 Train 데이터를 활용해 5 epoch 만큼 학습

8. Ensemble 실험


- **Ensemble 모델 선정 기준** - Public score 가 높으며 예측 경향성이 다른 모델 위주
- **단순 평균 Ensemble** - 단순히 평균값을 취해서 Ensemble
- **가중 평균 Ensemble** - Public score 를 기준으로 가중치를 부여해서 Ensemble

9. 최종 채택한 방법론 및 모델


- 데이터 전처리 및 증강
 - 사용한 최종 데이터의 개수 : 33390 개
 - train : 26712 개 / valid : 6678 개
- 모델
 - 최종 Ensemble 모델
 - 1) klue/roberta-large
 - 2) nlpotato/roberta_large-ssm_wiki_e2-origin_added_korquad_e5
 - 3) sdadas/xlm-roberta-large-twitter
 - 4) soddokayo/klue-roberta-large-klue-ner
 - 5) severinsimmler/xlm-roberta-longformer-large-16384
 - 제출 Ensemble 모델
 - 1) 증강된 전체 train 데이터를 활용해 LSTM layer 를 변형시켜 학습시킨 klue/roberta-large 와 nlpotato/roberta_large 를 평균값으로 Ensemble 한 모델
 - 2) 증강된 전체 train 데이터를 활용해 LSTM layer 를 변형시켜 학습시킨 RoBERTa 계열 모두를 평균값으로 Ensemble 한 모델

10. 최종 결과

[Public Score] micro F1-score: 76.3116 / AUPRC: 81.1209

7 (-)	NLP_09조		76.3116	81.1209	90	14h
----------	---------	---	---------	---------	----	-----

[Private Score] micro F1-score: 74.0375 / AUPRC: 81.1955

10 (3 ▼)	NLP_09조		74.0375	81.1955	90	16h
-------------	---------	---	---------	---------	----	-----