

KLUE-RE 랩업 리포트

📅 날짜

@2022년 3월 21일 → 2022년 4월 7일

프로젝트 개요

프로젝트 개요

한국어 Relation Extraction

- 주어진 문장 속의 특정 단어(entity)들에 대한 속성과 관계를 추론하는 문제
 - 예) **오라클**(구 **썬 마이크로시스템즈**)에서는 자바 가상 머신을 제공한다.
entity: (subject)썬 마이크로시스템즈, (object)오라클
relation: 별칭 org:alternate_names
- input 문장을 받아 2가지 단계의 task를 수행한다.
 - 개체명 인식(Named Entity Recognition)**
본 대회는 데이터 상에 entity의 위치를 index 형태로 나타낸다.
 - 관계 추출(Relation Extraction)**
본 대회의 최종 목표
- Relation Extraction은 다음 2가지로 활용될 수 있다.
 - QA 시스템**: 문장 속 개체명 인식과 관계 추출을 통해 정보 요약이 가능하며, 이를 사용해 QA 시스템 구축이 가능하다.
 - 지식 그래프 구축**: 관계 추출은 지식 그래프 구축을 위한 핵심 구성요소이며, 비구조적인 자연어 문장에서 구조적 triple을 추출해 정보를 요약하고, 핵심을 파악할 수 있다.

데이터

- input: sentence , subject_entity , object_entity 을 입력으로 사용한다.
- output: 30개의 relation 중 예측값을 pred_label , 30개 클래스에 대한 예측 확률을 probs 리스트로 생성한다.

프로젝트 환경

KLUE-RE 프로젝트 환경 (1)

Aa 속성	☰ tools
개발 환경(HW)	aistage competition 서버 및 GPU(Tesla V100)
개발 환경(IDE)	Jupyter notebook VSCode
협업	Gather Github Notion Zoom wandb

프로젝트 팀 구성 및 역할

- 역할 분담 방식: 처음부터 역할을 구조화하고 나누기 보다는, 회의를 통해 도출된 아이디어 중에서 각자 관심있는 실험을 수행하는 방식으로 진행했습니다.

KLUE-RE 프로젝트 팀 구성 및 역할 (1)

Aa member	☰ tag	☰ role
김선재 _T3252	Ensemble Model Optimization	BERT-base 모델 실험, RoBERTa-large 모델 실험 및 최적화, k-fold 적용, Special token 실험, multi-ensemble 수행
차경민 _T3215	Augmentation EDA Model	RoBERTa-large 모델 실험 및 최적화, BERT-base 모델 실험,데이터 증강 - back translation 수행 및 실험, EDA 탐색 및 실험, entity 역관계 수행 및 실험, DAPT, TAPT 적용
이도훈 _T3140	Ensemble Model Optimization	RoBERTa-large 모델 실험 및 최적화, dgpt2-base 모델 실험, xlm-RoBERTa-large 모델 실험, ensemble 구축, multi-ensemble 실험
김태훈 _T3065	EDA management	EDA, 데이터 불균형 이슈 탐색, 팀 github repository 관리, pre-trained model 탐색, huggingface 모델 실험 환경 셋팅
강진희 _T3007	Augmentation Ensemble Optimization	데이터 증강 - 품사 전처리 데이터 생성, entity 교차 데이터 생성,RoBERTa-large 모델 실험 및 최적화, KoBERT 모델 실험, KoElectra-base 모델 실험, KoBART 모델 실험

프로젝트 수행 절차 및 방법

🕒 프로젝트 타임라인

1주차 EDA, Augmentation

- 자율적인 실험 아이디어 공유 및 실험 계획과 수행, 베이스라인 학습 및 데이터 증강

2주차 Model, Optimization

- 모델 아키텍처 선택 및 결과 공유, 모델 최적화

3주차 Optimization, Ensemble

- 모델 최적화 및 앙상블 수행

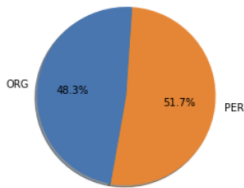
💡 협업 방식

- 모델 학습 코드, 앙상블 생성 코드 등 정리한 코드는 github 레파지토리에 공유한다.
- 모델 학습 시 wanDB 공통 프로젝트에 기록하며, run name은 모델명과 기타 실험 옵션이 드러나도록 설정한다.
- wanDB 기록 시, tag 사용 및 노트 사용을 적극적으로 활용하여 실험 공유가 용이하도록 기록한다.
- 오전 10시 데일리 스크럼에서 이전까지의 실험 결과와 이후 이어갈 실험 계획 혹은 새로운 실험 아이디어를 공유한다.
- 오후 4시 피어세션에서 오전-오후에 수행한 실험 결과를 공유하고, 팀원의 아이디어 및 실험에 피드백을 제공한다.
- 추가적 논의는 오후 9시 이후 게더타운 혹은 메신저를 통해 공유한다.

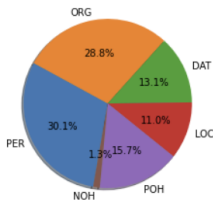
프로젝트 수행 결과

1. EDA

데이터 불균형

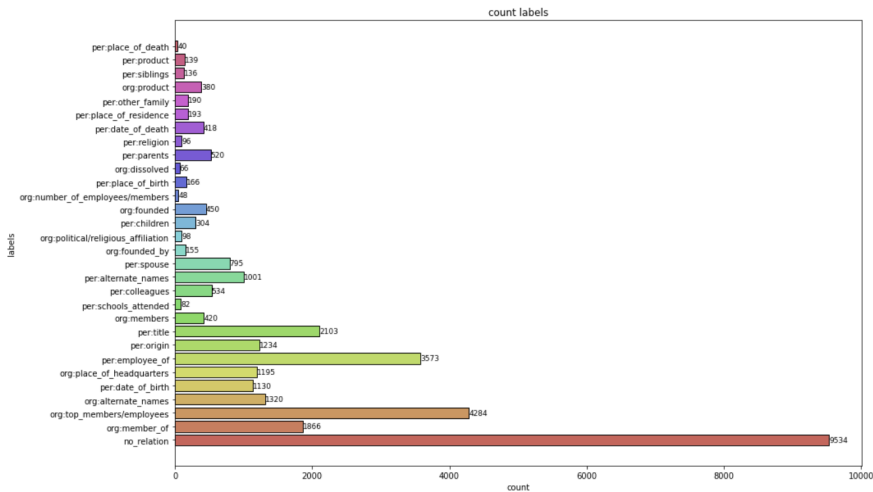


subject entity의 type 분포



object entity의 type 분포

- subject, object entity 비율 정도를 확인한 결과 object entity에서 데이터 불균형을 확인할 수 있었다.



30개 label(relation)의 문장 분포

- 30개 label의 분포를 통해 데이터 불균형을 확인할 수 있다. label은 모델의 타겟 변수이기 때문에 데이터 불균형을 처리할 필요가 있다고 보았고, metric 계산 시 사용되지 않는 no_relation 라벨을 제외한 29개 라벨에 대해 데이터 증강을 계획하게 되었다.
- 또한 불균형 문제를 처리하기 위해 K-fold를 적용한 학습 방식이나 train_test_split을 통해 validation 데이터를 나눠 학습에 사용하는 방법을 계획하였다.

중복 데이터 처리

- 대회 토론 게시판에 게시된 중복 데이터 및 mislabeling 이슈 토론 글을 바탕으로 중복 제거 및 라벨링 전처리
 - sentence 와 subject_entity , object_entity 모두 겹치는 중복 문장을 제거
 - 이재학_T3161 캠퍼님의 [중복데이터, 미스라벨링] <https://stages.ai/competitions/177/discussion/talk/post/1207>

- 정민지_T3196 캠퍼님의 [데이터 오타킹 문제 (재학님을 이어받아 진행한 내용 공유합니다!)]
<https://stages.ai/competitions/177/discussion/talk/post/1217>

2. Data Augmentation

Back Translation

- 파파고 API 제한 이슈 - 파파고 API 사용에 번역 1만자(약 90문장) 제한이 있다. train 문장이 약 3만 2천개이므로 많은 시간이 소요 될 것으로 예상했다.
 - 구글 API는 1만자 제한에서 자유롭다. 유료와 무료 버전 2가지가 있으며, 무료는 유료에 비해 상대적으로 성능이 떨어진다. 구글 API의 무료 버전을 기본으로 수행하고, 유료 버전은 번역 성능 검증용으로 사용하기에 적합하다고 판단
- 크롤링 - 파파고 API 1만자 제한 이슈에 대한 우회 방법으로
- 다양한 언어 선택 - 영어, 일본어
 - 한국어와 어순이 비슷한 언어로 역번역하는 것이 어순으로 인한 정보 왜곡을 방지할 것이라 생각했다. 레퍼런스가 많은 영어를 기본으로 역번역을 수행하고, 어순을 고려한 일본어 역번역을 부차적으로 수행했다.

품사 전처리

- no_relation을 제외한 29개 라벨의 문장 개수 평균은 약 850개, 중위수는 490개였다. 이와 비교하여 문장이 100개 미만인 label이 존재했고, 절대적인 문장 수가 부족하므로 Data Augmentation을 수행해야한다고 판단했다.
- 해당 augmentation 실험은 entity의 관계 해석에 영향을 미치지 않으면서 형태가 다른 문장을 생성하는 것을 목적으로 한다.
 - 문장 성분 중에서 의미 해석에 영향을 미치지 않는(제거해도 문장의 의미가 변질되지 않는) 성분은 형용사, 부사라고 판단하여 문장 성분 중 형용사와 부사를 제거한다.
 - `konlpy` 의 `Okt` 모듈을 적용하여 원본 데이터의 품사 태깅을 수행한다.
- 품사 전처리는 다음 절차로 수행한다.
 1. 문장 개수 400(default, 중위수 미만)개 미만의 label에 해당하는 문장을 대상으로 처리한다.
 2. `subject_entity` 와 `object_entity` 의 index 정보로 원본 `sentence` 에서 entity 단어를 `subject` , `object` 토큰으로 치환한다. (`subject` 와 `object` 는 `alpha` 로 태깅함을 확인)
 3. entity 토큰을 포함한 문장에 품사 태깅을 수행하고, entity 토큰을 다시 entity 단어로 치환한다.
- back translation(영어)과 문장 400개 미만 label에 대해 품사 전처리 문장 생성을 적용한 데이터로 RoBERTa-large를 학습시킨 결과, micro f1 2점 상승했다.

entity 교환

- subject entity와 object entity는 각각 사람(per)인지 조직(org)인지 등을 나타내는 type 정보를 가진다. EDA에서 object entity의 type 분포가 불균형하다는 것을 확인하였고, 데이터 증강에서 entity의 type도 고려할 필요가 있다고 생각했다.
- 문장이 적은 label에 대하여 같은 label과 각각 동일한 subject type과 object type 쌍을 가지는 문장끼리 분류하고, 동일한 속성인 entity의 단어만 교체하는 작업으로 문장을 추가 생성할 수 있다.
 - 동일한 label과 entity type 구조를 가지기 때문에 단어가 바뀌어도 관계 해석에 큰 차이가 없을 것이라 가정했다.
 - 예) label → per:members
문장1: 오늘부터 (subject)나희도는 (object)태양고의 펜싱선수이다.
문장2: (object)UBS의 (subject)백이진 기자는 펜싱 결승전 판정에 대해 단독 보도했다.
⇒ 문장1: 오늘부터 (subject)백이진 기자는 (object)UBS의 펜싱선수이다.
문장2: (object)태양고의 (subject)나희도는 펜싱 결승전 판정에 대해 단독 보도했다.



문장에서 entity를 제외한 나머지 성분은 entity간의 관계를 해석하기 위한 틀이라 보았고, 사람이 볼 때 문장의 호응이 어색할 수 있지만 모델이 해석하는 작업에는 크게 문제되지 않을 것이라 보았다.

- modify 전처리 작업과 back translation(영어), 품사 전처리를 마친 train_pos400.csv 데이터를 기준으로 문장 개수가 300개 이하인 label 17개에 대해서만 데이터 증강을 수행하였다. RoBERTa-large로 학습한 결과, micro f1 71점 후반의 성능이 나왔으나 최고 점수 대비 상승이 나타나지는 않았다.
따라서 데이터 증강이 성능 향상에 도움이 되지만, 항상 성능 향상을 일으키는 것은 아님을 알 수 있었다. 팀 내에서도 “이미 train 데이터도 편향되어 있는데, test도 마찬가지 아닐까? 그렇다면 결국 어떻게 하더라도 **편향 vs 편향**이기 때문에 **augmentation 기법을 통한 성능 향상이 크게 눈에 띄지 않을 것 같다**”는 의견도 있었다.

label 역관계

- subject entity와 object entity간의 관계를 나타내는 30개의 label은 서로 역관계를 가지는 label 쌍이 존재한다.
 - 예) per:members ↔ org:member_of, per:parents ↔ per:children
- EDA에서 확인한 label의 불균형을 해결하기 위해 역관계를 활용하여 문장 수가 적은 label에 대한 문장을 생성하고, 학습 데이터에 추가한다.
- entity 교체 모듈을 작성하여 데이터에 적용했지만, org:alternate_names, per:date_of_death 에 대해서만 교체 작업이 수행되었다. 이는 **코드 상의 문제**라고 보았으나, 동시에 진행한 back translation의 데이터 증강이 성공적인 효과를 보이며 이후에는 back translation 작업에 집중했다.

- 이후 프로젝트 결과를 보면 데이터 증강이 성능 향상에 효과가 있었는데, 이 작업을 중단하지 않고 데이터 증강으로 적용했으면 하는 아쉬움이 있다.

EDA

- KoEDA, KorEDA 사용으로 효과적인 데이터 증강 효과를 기대했으나 읽었을때 문맥 상 이해가 되지 않은 단어로 대체하거나 본문 장의 뜻이 변질된 문장을 얻었다. 데이터의 품질 또한 학습에 중요한 역할을 한다 판단했기에 해당 기술을 사용하지 않았다.

TAPT

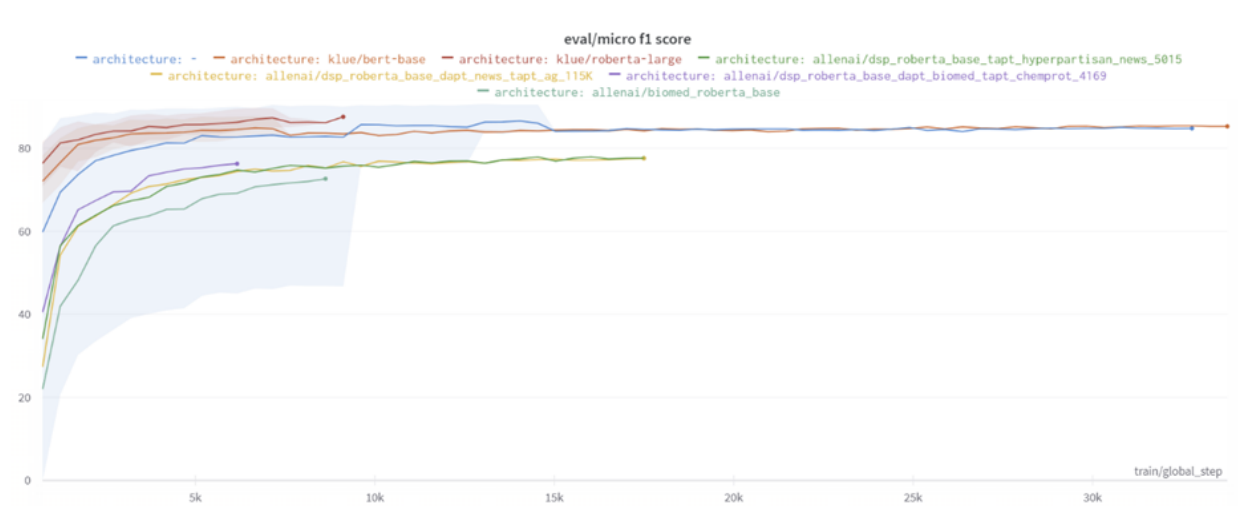
- Tapt 적용 모델을 사용했을 때 f1 score가 높을 것이란 예상과 달리 타 모델에 비해 낮은 f1 score를 보였다. 이는 주어진 데이터셋이 특정한 도메인에 편향된 정보가 아니었기 때문에 학습률이 떨어진 것으로 해석했다.

3. Model Architecture

모델 탐색 과정

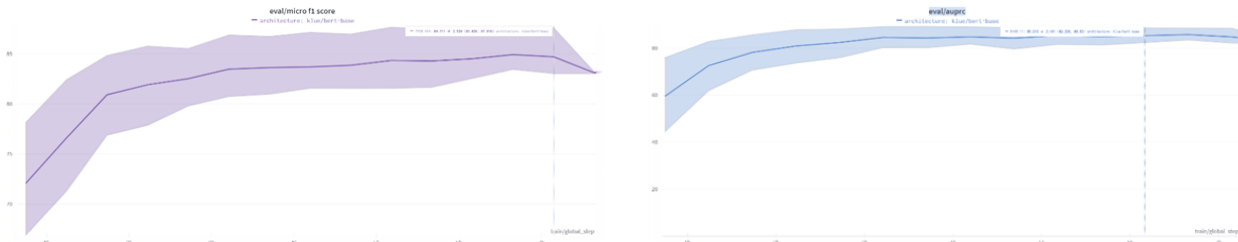
1. 베이스라인의 BERT-base 모델을 시작으로 모델 학습 및 결과 제출 수행 (public micro f1: 63.5429)
2. huggingface 모델 중 klue 모델인 RoBERTa-base 와 RoBERTa-large 를 적용하여 학습 및 결과 제출 BERT-base 와 비교하여 학습 성능이 좋은 klue/RoBERTa-large 모델을 기준으로 모델 최적화 수행. RoBERTa-large 적용 후 최소 70점대로 향상이 크게 되었음.
3. RoBERTa-large 외의 다양한 모델 실험을 목적으로 생성 모델 계열의 DGPT-2 실험. 하지만 RoBERTa-large 에 비해 성능이 좋지 않아 이후 모델 최적화에서 배제.
4. RoBERTa 계열의 높은 성능을 기대하며 다른 버전의 XLM-RoBERTa-large 를 실험했지만 RoBERTa-large 보다 좋은 성능을 보이지 않음.
5. 한국어 데이터에 맞게 한국어 pre-trained LM을 사용하면 성능이 좋을 것으로 기대하고 KoBERT , KoELECTRA , KoBART 를 실험. klue/RoBERTa-large 과 비슷하거나 그 이상의 성능을 기대했지만 성능이 좋지 않아 모델 최적화에서 배제.

모델 아키텍처 실험 결과

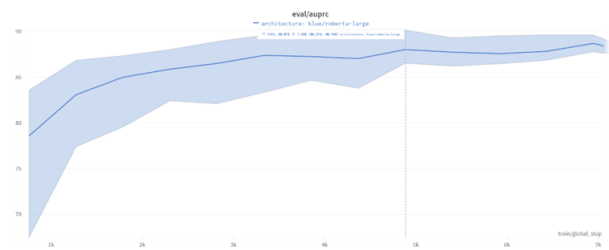
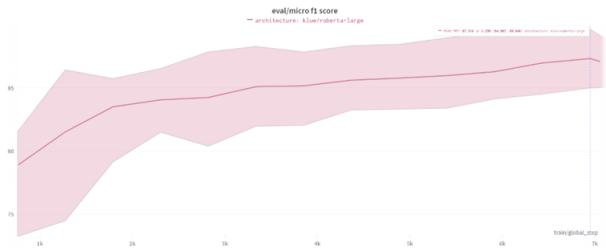


모델 아키텍처별로 평균값을 표현한 그래프이다.

다양한 모델을 사용해 학습시켜본 결과 대회 평가 지표인 micro f1 score과 auprc가 가장 높은 점수를 기록한 klue/RoBERTa-large, klue/Bert-base를 핵심 모델로 활용했다.



bert-base 모델은 기본 basic train 결과에서 training Argument 조정과 데이터 augmentation 을 통해 (83.026 → 87.616) 성능 향상을 이뤄냈다.



roberta-large 모델은 basic train 결과에서 lr_scheduler 파라미터 조정과 back translation data, 형용사 부사 제거 데이터셋을 활용해 (84.987 → 89.646) 성능 향상을 이뤄냈다.

4. Optimization

K-fold

- EDA를 통해 데이터 불균형을 확인했고, 이에 따라 불균형에 대응할 학습 방법으로 k-fold를 적용했다.
- default로 **k=5** 로 학습시켰다.

epoch

- default : **5**
- epoch를 **7** 로 늘렸을 때, public micro f1 점수가 0.7점 가량 상승했다. **10** 으로 늘리면 eval 점수가 좋았음에도 public 점수가 크게 떨어졌고, 과적합이라 판단했다. 따라서 **epoch=7** 로 고정함

learning rate

- default : **5-e5**

learning rate scheduler

- default : **linear** (public micro f1: 70.55)
- cosine** 으로 변경 후 **linear** 대비 micro f1이 2.5 정도 향상(public micro f1: 73.05)
- cosine_with_restarts** 를 추가로 적용했지만 micro f1이 오히려 66점대로 떨어졌다

batch size

- default : **32**
- 팀원 별 서버 메모리 사용량이 상이해 **[4, 8, 16, 20, 24, 28, 32]** 값을 사용해 학습시켰다. batch size가 클수록 학습소요시간 단축과 성능 향상의 경향을 보였다.

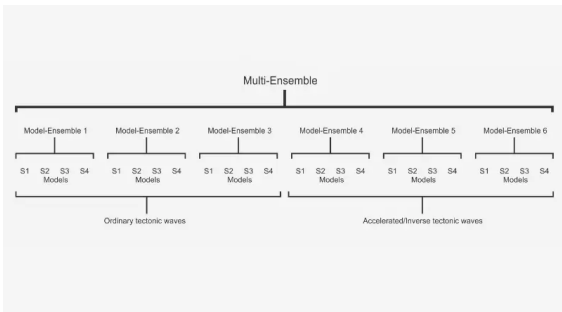
5. Ensemble

양상블 방법론

1. single-ensemble (voting)

- voting은 서로 다른 모델들의 prediction 값을 모아 마치 투표하는 방식으로 최종 결과물을 만드는 기법이다.
 - hard voting ⇒ prediction들 중 가장 많은 **갯수** 를 차지하는 값을 최종 값으로 결정한다.
 - soft voting ⇒ prediction들의 **평균** 을 계산하여 최종 값으로 결정한다.
 - 이번 프로젝트는 문장분류 task이다. 30개 label이 존재하기 때문에 각 prediction의 확률들을 전부 살펴보아야 한다.
 - hard voting을 적용하는 경우에는 다른 출력의 확률을 무시한 채로 편향이 계속 이어질 가능성이 높다고 판단하였다. 반면 soft voting을 적용한다면 prediction 각각의 값을 어느정도 최종 결과물에 반영할 수 있다고 보여 soft voting을 선택하였다.

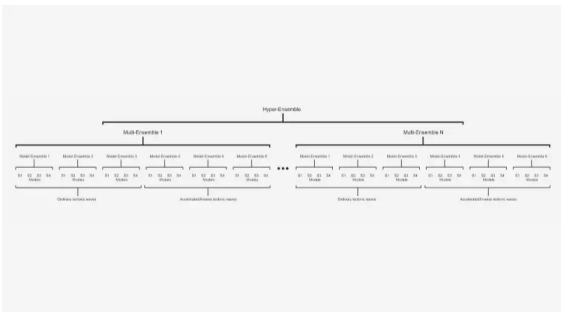
2. multi-ensemble



출처: Quantectum - [\[What are ensembles, multi-ensembles and hyper-ensembles?\]](#)

- 단일 모델에 대한 ensemble 결과를 여러 개 생성하고, 이를 묶어서 ensemble을 수행하는 것을 **multi-ensemble**이라고 한다.(ensemble의 ensemble)

3. hyper-ensemble



출처: Quantectum - [\[What are ensembles, multi-ensembles and hyper-ensembles?\]](#)

- multi-ensemble 결과를 여러 개 생성하고, 이를 묶어서 ensemble을 수행하는 것을 **hyper-ensemble**이라고 한다.

앙상블 실험 결과

- 1. 3-ensemble: public micro f1 상위 3개 결과의 앙상블(public micro f1: **73.37**)
 - 앙상블 성능 테스트를 위한 실험. 단일 모델 대비 점수가 큰 폭으로 오르지 않았지만, 점수 향상의 가능성이 보여 더 많은 결과 조합을 계획
- 2. 5-ensemble: **RoBERTa-large** 기반의 micro f1 상위 5개 결과의 앙상블(public micro f1: **73.05**)
 - micro f1 상위 기준의 앙상블은 모델 수를 늘려도 점수 향상이 크게 이루어지지 않았다. 그래서 **RoBERTa-large** 외의 모델을 포함한 조합을 계획
- 3. 7-ensemble: **RoBERTa-large** 모델 5개 + **xlm-roberta-large** 2개 앙상블(public micro f1: **74.32**)
 - **xlm-roberta-large** 의 경우 단일 결과로는 public micro f1이 60점대여서 앙상블 성능에 대한 기대가 낮았으나, 앞선 실험 대비 가장 큰 폭으로 오르게 되었다
- 4. 10-ensemble: **RoBERTa-large** 모델 7개 + **xlm-roberta-large** 1개 + **roberta-base** 1개 + **koElectra** 1개 앙상블(public micro f1: **74.54**)
 - public micro f1 점수를 보면 **xlm-roberta-large** 은 66점, **RoBERTa-base** 는 68점, **koElectra** 는 62점으로 단일 모델 성능이 낮은 편이었지만 최종 결과는 지금까지 진행한 ensemble 중에서 가장 높았다.
- 5. multi-ensemble: micro f1 기준 상위 4개 모델+ auprc 기준 상위 4개 모델 + **RoBERTa-large** 외 다른 모델 6개(**BERT-base** , **xlm-roberta-large** , **RoBERTa-base** , **KoELECTRA** , **KoBERT** , **dgpt-2**) (public micro f1: **74.87**)



높은 스코어에 기반한 앙상블보다는 **다양한 모델, 다양한 데이터셋을 학습시킨 모델**을 앙상블 시켰을때 높은 성능 향상을 보였다.
soft voting으로 하이퍼 앙상블을 하는 경우, 오히려 성능이 멀티 앙상블에 비해 0.2 하락했다. 이는 **레이블 값 평준화로 인한 현상**이라 해석했다.

- **micro f1 vs auprc vs 모델의 다양성**
 - rank 기준인 micro f1이 높은 모델을 앙상블하는 것과 다른 지표인 auprc가 높은 모델의 앙상블, 점수가 높지 않지만 기준으로 삼은 **RoBERTa-large** 외의 모델을 다양하게 섞은 앙상블 등을 실험해보았다.
 - 우선 리더보드 등수의 기준 지표가 micro f1이기 때문에 micro f1 점수가 높은 상위 5개~9개 모델까지 묶어서 앙상블을 수행해 보았다.
 - 단일 모델의 점수보다 높은 점수를 기록하였다. 하지만 **auprc 기준의 앙상블이나 모델의 다양성 중심의 앙상블에 비해 높은 성능을 보이지 않았다.**
 - 단일 모델의 제출 결과 중에서 micro f1에 비해 auprc가 비약적으로 높은 모델이 존재했다. 그래서 또 다른 성능지표인 auprc가 높은 결과 5개를 묶어서 앙상블을 수행했다.
 - micro f1 기준의 앙상블보다 높은 micro f1 점수를 기록했다. 앙상블이 모델 간의 보완을 목적으로 한다는 점을 미루어 보아 **micro f1 중심의 앙상블보다 auprc 중심의 앙상블에서 레이블 값의 교정이 활발하게 이루어졌음을 짐작할 수 있다.**
 - micro f1보다 auprc 중심의 앙상블이 더 높은 점수를 기록하여 다양한 관점의 앙상블을 시도할 필요를 느꼈다. 그래서 **모델 간 보완을 목적으로 하는 앙상블 매커니즘에 맞게, 제출 점수는 높지 않지만 RoBERTa-large가 아닌 다른 모델을 포함하여 앙상블을 수행하였다.**
 - 단일 모델에서 점수가 높지 않았지만, **다양한 모델을 섞을수록 더 나은 성과를 보였다.**
- **multi-ensemble**
 - 높은 점수를 보이는 앙상블 결과를 모아 soft-voting을 수행하여 multi-ensemble 결과를 만들었다.
 - 앙상블에 사용한 모델의 종류가 다양할수록 점수 향상 폭이 높게 나타났다. 이를 통해 **단일 모델의 성능이 안 좋더라도 앙상블을 위한 재료로 쓰일 수 있음을 배웠다.**
 - 깊은 앙상블 구조로 인한 **과적합**을 우려하기도 했다.

6. 최종 제출 모델

최종 제출 모델 선택 기준

- micro f1 점수를 기준으로 보면 상위 제출 결과는 모두 multi-ensemble이다.
 - multi-ensemble은 앙상블이 중첩된 구조로 과적합 문제가 발생할 수 있다. 따라서 2가지 제출물 중 1가지만 micro f1 순서에 따라 multi-ensemble 결과를 선택한다.
- 과적합으로 인한 private 점수 하락을 방지하기 위해 single-ensemble의 결과를 최종 선택한다.

최종 제출 결과

micro_f1_score : 74.5828 → 72.4900

auprc : 78.7720 → 78.2155

자체 평가 의견

👍 잘한 점

1. (협업) 지난 level 1 이미지 분류 대회와 비교하여 **협업툴을 적극적으로 사용하고, 협업 경험을 쌓을 수 있었다.**
 - 지난 level 1의 이미지 분류 대회 회고 시, 협업툴의 활용과 프로젝트 진행 방식에 대해 부족함을 느꼈고 이후 대회에서는 협업 툴의 적극 활용을 목표로하였다.
 - 대회를 시작하며, wanDB 팀 프로젝트 셋팅과 run 명령 규칙을 논의하였고 이후 실험에서 wanDB로 모두의 실험을 모니터링하고 사용한 파라미터를 기록했다. 특히 level 1 회고 이후, 다음 대회를 대비하여 wanDB 사용법을 익혔던 것이 큰 도움이 되었다.
 - k-fold 실험 코드, ensemble 코드, augmentation 버전의 데이터 파일 등 각자 맡은 기능을 개발하고 그 결과물을 팀 github repository에 공유했다.
(대회 마감일 기준 level 1 대비 commit 횟수 **140%** 증가)
2. (프로젝트 실험) **피어세션에서 공유한 실험 계획을 90% 이상 모두 실험해보았다.**
 - EDA를 통해 확인한 데이터 불균형과 멘토링 피드백을 기반으로 데이터 증강, 모델 실험 및 최적화 방법을 논의하였고, 피어세션에서 언급된 실험 계획을 **90%** 이상 모두 수행해보았다.
 - augmentation, ensemble 방법에 대해 자유롭게 아이디어를 논의하고 관련 방법론과 reference를 찾아 실험 계획을 발전시켰다.
3. (기술) **huggingface의 활용법을 배우고 pytorch 역량을 발전시켰다.**

💡 시도했으나 잘 되지 않은 것

1. (협업) github의 코드 공유, 데이터 공유는 적극적으로 수행했으나 **버전 관리는 활용하지 않았다.**
 - master-feature 브랜치를 나누고 프로젝트 버전관리를 적극 활용하는 등의 개선이 필요하다.
2. (모델 실험) 다양한 모델을 사용하기는 했지만 klue/roberta-large 외에는 성능이 좋지 않았다.(dgpt2,xlm-roberta,kobert,koelectra 등) 이는 **fine tuning의 부재 혹은 데이터나 task에 대한 고찰 없이 모델을 사용했기 때문인 것 같다.**
 - 사용 모델의 논문을 살펴보고 해당 task에 적절한지, 적절하게 사용하기 위해서 어떤 방식으로 최적화할 지 분석할 필요가 있다. 특히 논문의 실험 조건과 비슷한 parameter 세팅 등의 환경 구축이 필요하다.
 - 한국어 pre-trained LM을 사용했지만 성능이 기대했던 것만큼 나오지 않았다. fine-tuning이 제대로 수행되지 않았거나 혹은 최적화 수행이 부족했던 것 같다.
3. (special token) punctuation을 시도해봤지만 적용시키는 것이 쉽지 않았다. task에서 사용하는 스킴들은 정해져 있는데 한번이라도 시도해보면 좋았을 것 같다.

✎ 아쉬운 점

1. **첫 주치의 제출 횟수가 거의 없다.** 한정된 제출 기회와 시간을 고려했을 때, 놓친 시간과 기회가 아쉽다.
2. **모델의 사용법을 더 숙지할 필요가 있다.** 특히 huggingface에서 모델을 가져다가 쓰기 전에, 관련 논문을 보고 논문의 실험 환경과 동일하게 환경을 설정하거나 task에 맞는 최적화 방법을 더 생각해본 다음에 모델을 적용할 필요가 있다.
3. level 1처럼 **직접 모델을 쌓아보거나 pre-trained 모델에 레이어를 추가로 쌓는 등** 모델을 세부적으로 뜯어보고 사용하는 실험도 계획하면 좋을 것 같다.
4. **반복적인 EDA를 하지 않고, 프로젝트 초반의 EDA에서 그쳤다.**
 - 프로젝트에서 EDA는 초반에 데이터 불균형을 탐색하는 단계에서 적극적으로 쓰이고, 이후에는 EDA를 거의 수행하지 않았다. 멘토링과 피어세션 회고를 통해 inference 수행 결과를 탐색하고, 어떤 label에 대한 정확도가 떨어지는지를 확인하는 EDA 과정이 부족했음을 깨달았다.
 - 모델 실험이나 최적화 과정에서도 f1 score가 기대만큼 나오지 않으면, 실험을 버리고 새로운 실험을 셋팅했다. 하지만 결과가 좋지 않은 실험도 추가적으로 어떤 요인 때문인지 분석할 필요가 있다. 따라서 결과가 좋지 않다면, 왜 결과가 안 나오게 되었는지 확인 해보고 EDA를 다시 하면서 문제를 분석하는 과정을 가져야 겠다.

📖 배운 점 & 시사점

- **데이터의 중요성**
 - 데이터 증강이 데이터 문제를 모두 해결해주지 않는다. 증강을 시도하기 이전에 **처음부터 질 좋고 크기가 큰 데이터의 구축이 중요하다.**
 - **모델 선택이나 최적화의 작업 보다는 데이터가 정말 중요하다.** 그리고 데이터 분석이 더욱 중요하다. 데이터 분석을 통해 문제가 무엇인지 확인하고, 그 문제를 해결하기 위해서는 어떤 방법을 시도해볼 지, 그리고 그 결과를 다시 또 분석하는 실험 루틴의 중요성을 배웠다.
- 실제 유저가 사용시에는 정제된 데이터를 주고받지 않는다.
너무 정제된 데이터를 사용하는 대회와 같은 환경은 실제와는 괴리가 있지 않을까?
그렇다면 학습 데이터는 어떻게 구축하는 것이 좋을까?

개인 회고

차경민_T3215

- 이번 프로젝트에서 나의 목표는 무엇이었는가?
대회 전의 나의 목표는 크게 두가지로 Hugging face 많이 사용하고 공식 문서 많이 보기 그리고 팀원들과 함께 계획을 세워서 체계적으로 대회를 진행하는 것 그리고 협업 툴에 익숙해지는 것 이였다.
- 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?
매일 피어 세션마다 팀원들과 대회의 방향에 대한 이야기를 나누었다. 각자 어떠한 부분을 개발할지를 정하고 데이터 증강, 하이퍼 파라미터 수정, 앙상블 등 각자 역할을 나누어서 대회를 진행하였다. 또한 각자의 실험 결과를 wandb를 통해서 공유하며 의미 있었던 실험결과에 대해서 공유하였다.
- 나는 어떤 방식으로 모델을 개선했는가?
 - EDA
저번 대회에서 EDA의 중요성을 깨닫고 이번 대회에서는 차근차근 EDA부터 진행을 하였다. 각 sentence 별 길이의 분포, 문장 길이 분석, subject entity와 object entity 분포 확인, labels 분포 등을 확인하면서 labels 데이터의 심각한 데이터 불균형 문제를 파악하였다. 이를 해결하기 위하여 Back Translation, Easy Data Augmentation, 문장 간소화, 같은 entity간의 swap을 통하여 Data Augmentation을 진행하였다.
 - Easy Data Augmentation
텍스트 데이터의 불균형 데이터 문제를 해결해주는 방법 중 하나인 Easy Data Augmentation은 SR(유의어 교체), RI(단어 삽입), RS(단어 교환), RD(단어 삭제) 방법으로 텍스트 데이터를 증강시켜주는 방법을 말한다. 자연어 처리 분야에서 효과적인 데이터 증강 방법 중 하나라고 이야기한다. 나는 nlpaug, KoEDA, KorEDA,의 라이브러리를 사용해보았다. nlpaug는 한국어를 지원하지 않는 바람에 시도하지 못했고, KoEDA, KorEDA는 증강된 텍스트 데이터의 퀄리티 문제로 사용하지 않았다. 하지만 지금 뒤돌아봐서 생각해보니 증강된 텍스트 데이터를 한 번이라도 사용은 해봤으면 어땠을까 하는 생각이 든다. 사람이 봤을 때 어색한 문장이더라도 기계가 학습할 때는 조금 더 모델의 다양성을 증가시켜줄 수도 있겠다 라는 생각이 든다.
 - Back Translation
Back Translation도 텍스트 데이터의 불균형을 해결해주는 방법 중 하나이다. 처음에는 Google, Papago API를 이용하여서 Back Translation을 진행하려고 하였으나 하루에 지원되는 단어 제한과 퀄리티 문제로 인하여 진행을 하지 못했다. 하지만 파 이썬 라이브러리를 이용하여 웹 크롤링을 진행하였고 Papago 페이지로 query를 보내고 결과 값을 크롤링 해오는 방식을 통하여 Back Translation을 진행하였다. 물론 크롤링해오는데 시간이 오래 소모되기는 했지만 데이터를 증강하는 것에 의미가 있었다.
 - 하이퍼 파라미터 수정 & 앙상블
데이터의 증강도 어느정도 진행이 되어서 그 이후로는 계속해서 하이퍼 파라미터 수정을 진행하였다. 기존에는 하이퍼 파라미터의 중요성을 잘 인지하지 못했는데 계속 대회를 진행하면 진행할수록 learning rate, batch size에 따라서 모델의 성능이 달라지는 것을 체감할 수 있었다. 또한 잘 나온 모델들을 이용하여서 앙상블을 진행하였다. 역시나 앙상블을 이용하면 일반화가 되어서 모델의 성능이 조금 올라가는 것을 확인할 수 있었다.
- 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?
이번 대회는 한국어 텍스트 데이터의 증강이 주된 작업이었다. 이미지 데이터와는 달리 단어 하나만 바뀌어도 문장의 의미가 달라지는 텍스트 데이터의 증강은 새로 맞닥뜨리는 이슈였다. 나는 Easy Data Augmentation, Back Translation을 이용하여서 텍스트 데이터의 불균형을 최대한 줄여주고자 하였고 데이터를 증강시키면 시킬수록 어느정도 까지는 LB스코어와 auprc 값이 증가하는 것을 확인할 수 있었다. 하지만 데이터를 아무리 증강을 하여도 LB 스코어가 더 증가하지는 않고 오히려 떨어지는 것을 확인하였다. 데이터가 그냥 많다고 좋은 것이 아니라 데이터의 퀄리티 또한 중요하다는 부분이라는 것을 알게 되었다.
- 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?
한국어 텍스트 데이터의 증강을 진행하여 영어-한국어, 일본어-한국어 간의 Back Tanslatoin을 진행하였다. 기존의 베이스 모델에서 micro-f1 점수는 5점, auprc 점수는 8점 정도의 상승이 있었다.
- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?
대회가 마무리되고 든 생각이 사람들이 대회에서 적용하는 방법(혹은 스킵)들이 거의 다 정해져 있구나라는 생각을 하게 되었다. 이번 대회만 보더라도 성적이 좋았던 팀들과 그렇지 못한 팀들 모두 시도했던 방법들은 비슷하다는 생각을 하게 되었다. 이번에 다시 한 번 대회의 순위는 중요하지 않는구나 라는 생각이 들었다. 대회의 순위가 중요하지 않다면 "우리는 앞으로의 진행할 대회를 통하여 무엇을 얻으면 좋을까?" 라는 생각을 해보았다. 가장 먼저 드는 생각은 최대한 다양한 방법들을 직접 시도해보는 것, 그리고 많이 실패하는 경험에서 이유를 찾는 것이다. 또한 깃헙이나 wandb 등 협업 툴을 제대로 다루어야 겠다는 생각을 하게 되었다.

이번 대회에서 제일 아쉬운 부분 중에 하나가 문장에 Punctuation 을 추가해보지 못한 점이다. 개인적으로 각 entity에 punctuation을 넣어준다고 해서 모델이 더 잘 학습할 것 같지도 않았고, 별로 의미가 없다고 판단을 하여서 시도해보지 않았다. 이번에 모델의 성능이 좋았던 팀들을 보면 punctuation을 넣어준 팀들이었다. 이번에 시도해봤으면 다음 nlp 대회에서 충분히 좋은 스킬이 되었을건데 시도조차 해보지 않은 점이 너무 아쉽다.
- 한계/교훈을 바탕으로 다음 P-Stage에서 스스로 새롭게 시도해볼 것은 무엇일까?
다음 P-Stage부터는 기존의 베이스 모델을 그대로 사용하는 것이 아니라 우리 팀만의 베이스 라인 코드를 만들고 싶다. 대회가 마무리 되고 이전 기술들의 repo를 보니 각자 팀에 맞게 코드의 양식을 변경하여서 사용하는 것을 보니 우리도 우리 팀만의 베이스 라인 코드를 만들면 좋을 것 같다고 생각을 하였다. 또한 프로그램 실행을 shell을 이용하여서 진행해야겠다는 생각을 하였다. 또한 github도 단순히 pull, push만 하는 것이 아니라 pull request, 새로운 develop 브랜치를 만들어서 체계적으로 진행을 하면 나중에 도움이 될 것이라고 생각을 하였다.

이도훈_T3140

- 학습 목표
 - 이번에 진행한 프로젝트는 한국어 문장 분류 task였다. NLP 도메인의 심화과정을 배우면서 본격적으로 배운 것들을 시험해볼 기회라고 생각했다.
 - 개인적으로 가진 목표는 우선 BERT나 GPT 모델을 써보면서 트랜스포머 encoder 와 decoder의 차이점을 직접 느껴보는 것이었다. 문장 분류 task의 경우 아무래도 object와 subject간의 관계를 잘 반영할 수 있는 bi-directional한 BERT가 강점을 보일 것이라고 생각하였는데, 실제로도 그러한 결과를 얻었다.
 - 그 밖에도 자연어 데이터의 augmentation 방법들과 허깅 페이스의 사용방법, 그리고 지난 대회 이후 꼭 사용해보자고 하였던 wanDB 같은 협업 툴을 써보는 것이 목표였다.
- 시도해본 것들
 - 우선 이전 프로젝트의 다짐을 반영해 협업툴을 적극적으로 사용하였다. github를 이용해 단순히 코드 공유만 하는 것이 아니라, 그때 그때 필요한 기능들을 각 팀원들이 구현하고 가능한 공통의 코드를 사용하여 소통이 수월하도록 하였다. 또, 실험 모니터링 & 공유 기능이 탁월한 wanDB를 사용하여 각 팀원들이 진행한 실험을 다같이 살펴보면서 효율적인 실험 진행이 가능했다.
 - 허깅 페이스와 여러가지 PTM들을 자유자재로 다뤄볼 수 있도록 가능한 다양한 모델들을 불러와 프로젝트에서 써보았다. GPT 나 XLM, BERT, Roberta 를 사용했다.
 - 이전 대회에서는 ensemble을 시도해보지 않았다. 그래서 이번에는 꼭 써보고자 하였다.
- 배운 점
 - 자연어 데이터의 augmentation은 난이도가 있는 task라는 것을 배웠다. 특히 EDA에 대해 배우면서 단순히 문장을 조작하여 양을 늘리는 것은 문장의 원래 의미를 훼손시킬 수 있다는 것을 알게 되었다.
 - Back-translation 역시 원래의 문장과는 다른 의미로 번역되는 경우, 고유명사나 한자어가 증발해버리는 경우, subject-object 의 index가 바뀌는 문제, 번역 API의 호출 한계가 있는 문제 등을 겪으면서 Data의 중요성을 크게 느꼈다.
 - ensemble을 시도하면서 다양한 기법들이 존재한다는 것을 알게 되었다. 또, 실제로도 성능의 향상이 크게 나타나서 단순하지만 강력한 기법이란 점을 배웠다.
- 한계점
 - 다양한 모델을 사용해보고자 하였지만 코드를 돌아가게 하는 것 부터가 난관이었다. 돌아는 가더라도 성능이 상당히 낮게 나오거나, train 도중 metric이 실종되는 등의 문제를 겪었는데 해당 문제를 해결하는 것에 시간을 상당히 많이 사용하였다. 후에 마스터 클래스에서 재현이 안 되는 경우에는 원 논문을 읽고, 그 환경을 내가 적절하게 구축 한 것이 맞는지, 내가 하고자 하는 task나 data가 적절한 지를 알아보라는 말씀을 듣고 해결의 실마리를 찾게 되었다.
 - 반복적인 EDA를 해보지 않은 것이 아쉽다. 모델을 돌려보고 점수가 높던, 낮던 그 이유를 찾으려 하지 않았다. 정확히는 대중 어렵짐작으로 생각한 것이 문제였다. 다음부터는 모델의 inference를 직접 찍어보거나, confusion matrix 등 여러 시각화를 사용하면서 가설을 세운 대로 실험이 잘 진행되고 있는지, 혹은 어떤 문제가 있는지 등을 잘 모니터링 해야겠다고 느꼈다.
- 다음에 해볼것들
 - 특정 모델을 선택하고 돌려보기 이전에 반드시 해당 모델의 논문을 읽어볼 것
 - 무턱대고 실험을 진행하지 말고 반드시 EDA를 동반한 분석과 가설 수립 후 실험을 진행할 것
 - 데이터셋 구축 및 증강의 다양한 방법들을 숙지 및 시도해볼 것

김선재_T3252

[학습 목표]

- 생소한 과제 목표였기에 목표를 잘 분석하고 전체 프로세스에 대한 이해도를 높이는 것을 목표로 삼았습니다.
- 다양한 생각과 그에 대한 실천으로 새로운 방향성을 찾을 수 있는 것을 알기에 팀원들과 자유롭게 아이디어를 나누는 것을 목표하였습니다.
- 지난 대회에서 모두에게 아쉬움을 남겼던 협업에 관련된 부분을 wandb, 게더타운 등을 활용해 보완하기로 계획하였습니다.

[시도해 본 것]

Continuous learning – 데이터를 fold로 나누어 fold당 train, validation 셋으로 분리 후 이를 연속적으로 학습시킨다면 k_fold에 비해 높은 검증 신뢰도와 학습률을 보여주지 않을까 생각해 코드를 생성해 학습을 진행해 보았습니다. 그러나 k_fold를 적용했을 때의 스코어에 비해 오히려 미세한 스코어 하락을 보였습니다. 이에 대한 원인으로 본 대회에서는 epoch가 커질수록 성능이 저하되는 경향을 보였는데 fold당 여러 번의 epoch를 학습함으로 과적합 상태에 빠진 것으로 풀이했습니다.

Multi ensemble – 멀티 앙상블, 하이퍼 앙상블을 시도할 경우 레이블 지표 값들이 평준화되어 오히려 성능 저하가 될 수 있다는 단점을 고려해 멀티 앙상블 모델과 단일 모델을 섞은 앙상블을 시도했고 이를 통해 성능 향상을 이뤄냈습니다.

[모델 개선 방법]

training 과정에 집중해 대회를 진행하였습니다. 다양한 scheduler, optimizer를 사용해보고 hugging face에서 제공하는 training arguments, custom train안 파라미터들을 살펴보며 값들을 조정했습니다.

[한계, 아쉬운 점]

EDA를 많이 하지 않은 것이 아쉽습니다. 이번 대회는 데이터 분석을 함으로 많은 것을 얻어갈 수 있었음을 대회가 마감에 이르러서야 생각이 들었습니다.

간단한 punctuation 추가나 special token 추가를 했을 때 모델 성능 향상이 된다는 글을 읽어 시도했으나 생각보다 큰 모델 향상을 이뤄 내지 못했습니다. 해당 시도를 더 분석하고 시도해 보지 않은 것이 아쉽습니다.

강진희_T3007

학습 목표

- 가설-실험 계획-실험 수행-결과 정리 라는 일련의 과정으로 체계적인 프로젝트를 수행하는 것
- 데이터 증강-모델 아키텍처 실험-모델 최적화-양상블과 같은 프로젝트의 전반적인 단계에 참여하는 것
- 프로젝트 수행 기록을 꼼꼼하게 작성하는 것

모델 개선을 위해 시도한 방법

- level 1과 마찬가지로 데이터 불균형이 존재함을 확인하고, 데이터 증강을 위한 아이디어 실험을 크게 2가지 수행했다.
 1. 관계 해석 및 문장의 의미 파악에 영향을 미치지 않는 형용사와 부사 전처리, 전처리 버전의 문장을 추가하여 데이터 증강하기
 2. 동일한 entity type과 label의 문장을 entity 단어만 교차하는 데이터 증강
- 한국어 pre-trained LM의 높은 성능을 기대하고 huggingface의 `monologg/KoBERT`, `monologg/KoELECTRA`, `hyunwoongko/kobart` 를 적용

깨달은 것

- 데이터가 가장 중요하다. 양질의 데이터를 확보하는 것이 최우선의 과제이며, 데이터에 맞는 모델 아키텍처가 무엇인지 찾는 실험이 진행되어야 하고, 모델에 따른 최적화가 수행되어야 한다는 것을 느꼈다.
- 프로젝트를 수행하면서 목표 달성을 위한 가설을 세우고, 구체적인 실험 계획과 수행, 결과를 꼼꼼히 기록하는 프로세스의 중요성을 깨달았다. 대회에 끌려가는 것이 아니라 내 방식대로 주도할 수 있는 방법이기 때문이다.

새롭게 시도한 변화와 효과

- 지난 번 대회는 점수 올리기를 위한 목적 없는 시도가 난발했는데, 이번에는 **아이디어를 피어세션에서 공유하고 → 팀원의 피드백을 반영해 실험 계획을 세우고 → 계획한 내용을 끝까지 수행하고 → 결과를 공유**하는 일련의 프로세스를 경험했다.
 - 특히 결과 제출 기회에 연연하지 않고, 계획했던 결과물을 만들어낼 때까지 끈기있게 작업에 몰두한 것이 이전보다 성장한 점인 것 같다.
- 대회 첫 주차를 데이터 증강 작업에 모두 투자하였고, 이후 모델 아키텍처 실험에 증강한 데이터가 주로 쓰이면서 프로젝트에 제대로 기여한 것 같아 뿌듯하다.
- 모델 최적화와 양상블도 팀원의 실험 결과와 아이디어를 바탕으로 수행하였고, 이전 대회보다 더 제대로 된 협업을 수행한 것 같아 만족스럽다.
- 우선 작성한 코드나 수행한 내용, 에러가 난 내용까지 모두 적었다. 그러지 않으면 나중에 어떤 행동을 했는지 알 수 없어 대회를 마치면 ‘한 것이 없다’는 느낌을 받았기 때문이다. 이번에는 사소한 것까지 기록하였고, 덕분에 랩업 리포트 작성과 회고에 많은 도움이 되었다.
- back translation이나 TAPT처럼 정의된 방법론이 아니어서 작업을 수행하면서도 의구심이 계속 됐다. 하지만 제출 결과에서 성능 향상이 가시적으로 드러난 덕분에 대회 데이터셋에 맞는 방법론을 커스터마이징 한 것 같다는 만족감을 얻었다.
 - 또한, 아이디어를 구조화하고, 이를 코드로 나타내고 원하는 결과물을 구현하는 작업에서 희열을 느꼈다.

마주친 한계, 아쉬운 점

- 한국어 LM 적용 실험(KoBERT,KoELECTRA,KoBART)은 기대한 성능은 나오지 않았다. 반복되는 학습 오류와 기대 이하의 성능으로 미루어보면 모델을 가져다 쓰고, task에 맞게 fine-tuning하는 작업에 대한 이해도와 응용력이 아직 부족한 것 같다.
- 지속(반복)적인 EDA의 부재. 프로젝트 첫 단계에서 원본 train 데이터에 대한 EDA만 진행하고, 이후 모델 최적화 실험 등의 단계에서는 EDA를 진행하지 않았다. 결과물을 직접 찍어보고, 모델이 어떤 문제는 잘 맞추는지 혹은 잘 맞추지 못하는 지를 지속적으로 확인하고 이를 실험에 반영했으면 더 깊이 있는 프로젝트가 되었을 것 같아 아쉬움이 남는다.

다음 프로젝트에서 스스로 새롭게 시도해볼 것

- 모델을 적용하기 전에 논문을 먼저 리뷰하고, 논문 이해 후에 해당 모델을 주어진 과제에 알맞게 fine-tuning하는 실험을 중심으로 수행해야겠다.
- 모델이 어떤 label을 잘 맞추고, 잘 맞추지 못하는 지 지속적인 EDA를 통해 데이터를 직접 찍어보고 확인하는 프로세스로 실험을 수행해야겠다.
- 실패한 아이디어에 대해 원인이 무엇인지, 개선 방법은 무엇일지를 깊게 고민하는 시간을 의도적으로 가져야겠다.

김태훈_T3065

[학습 목표]

- NLP 분야에 있어 필수적인 태스크 중 하나이기에, 태스크 수행 중 쓰이는 기법들에 대해 체계적으로 정리해 보기
- RE 태스크는 결코 쉬운 태스크는 아니라고 생각해, 성능을 올릴 수 있는 스킬들에 대해 폭넓은 서치를 해보기
- 협업에 있어 체계적인 프로세스를 만들어 다음 스테이지를 대비하기

[시도한 방법들]

- 이전보다 원활한 커뮤니케이션, 개선된 협업 프로세스를 갖추는 것이 중요한 목표 중 하나였기 때문에 기본적으로 개더타운에서 소통을 위한 모임을 자주 가졌습니다. 또한 서로의 모델을 효율적으로 공유하기 위해 모델 명명규칙에 대해 논의했으며 그 과정에서 wandb 를 적극 활용했습니다.
- 이전보다 다양한 베이스라인을 시도해 보면서, 가용 가능한 모델들의 개수를 늘려 보았습니다. 또한 각 모델을 돌릴 때 명명규칙에 의거해 여러 옵션으로 돌리다보니 효율적으로 앙상블을 할 수 있는 여지가 높아졌습니다.
- 전체적으로 Data augmentation 이 유의미한 성능 개선을 불러 온다고 생각해, 다양한 방식으로 시도해 보고자 노력했습니다.

[배운 점]

- Data augmentation 의 파급력은 클지언정, 효과적인 방향으로 하는 것은 결코 쉽지 않다는 것을 알았습니다. 이 과정에서 더 깊은 생각을 하지 못하고 결과 의존적으로 진행하는 경우 비효율적인 프로세스가 될 수 있다는 것도 배웠습니다.
- 앙상블 역시 매우 효과적인 기법이지만, 여러 옵션이 존재해서 공부할수록 그 효과를 극대화할 수 있다는 것을 알게 되었습니다.

[한계, 아쉬운 점]

- 이전보다 더 많은 커뮤니케이션을 하는 것이 목표였는데, 어느 정도는 성공했으나 더 발전시킬 여지가 많은 것 같습니다.
- 개인적으로 다른 팀원 분들에 비해 꼼꼼하게 진행하는 부분이 부족하고, 시도해본 모든 부분에 대해 로그를 남기는 것이 아직 익숙하지 않아 여전히 아쉬웠습니다.
- EDA를 초반에만 하는 것이 아니라, 중간중간 각 모델의 성능에 의거해 진행하면 더 좋은 성능을 낼 수 있을텐데 현실적으로 진행하지 못한 것이 아쉽습니다.

[다음에 시도해볼 것]

- 모든 방법론을 결과 지향적으로 접근하는 것이 아니라, 태스크에 대한 깊은 고민과 모델 기초 이론에 대한 이해를 기반으로 진행해보기
- 또한 여러 모델의 성능을 비교해보고 그 과정에서 EDA를 꼭 진행해본 후에 다음 스텝으로 진행하기
- Data augmentation, 앙상블 등 성능을 크게 올릴 수 있는 스킬들에 대해 더 공부해서 인사이트를 갖추기