KLUE(Relation Extraction)

Wrap up Report

2021/09/27 ~ 2021/10/08

AI-ESG (11 조)

boostcamp

1. 프로젝트 개요

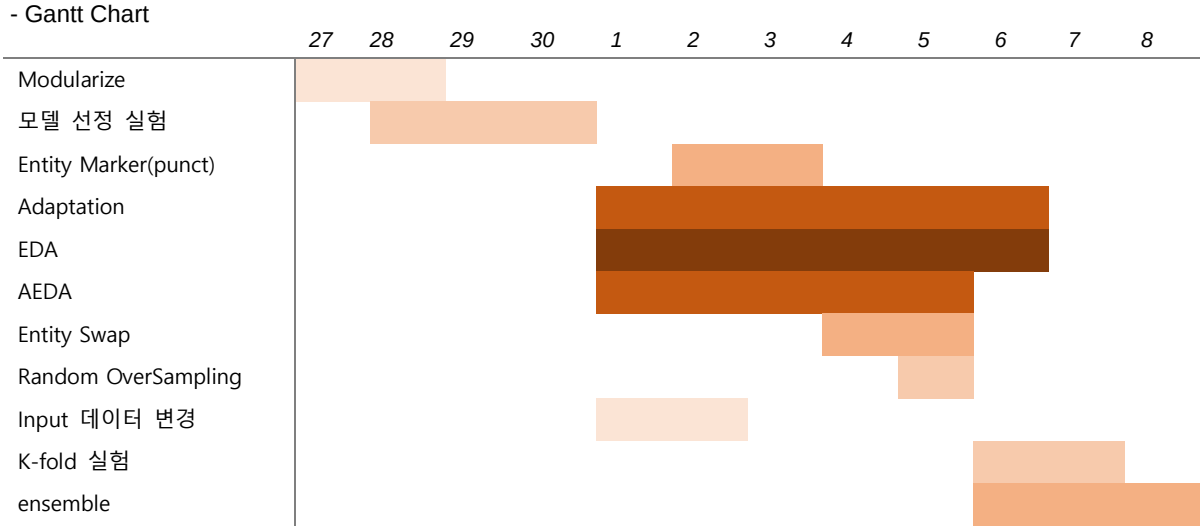
- Relation Extraction task
    - 문장 속 두 단어의 관계를 추론하는 모델 학습.
    - 모델은 두 단어와 문장을 입력으로 받고, 지정된 두 단어의 관계를 30 가지 class 중 하나로 예측한다.
- 해당 task 에서 추가적인 전처리와 테스트셋 언어모델 사전학습, 앙상블을 통해 KLUE 벤치마크* 대비 약 4%의 micro f1-score 향상을 확인하였다.
    - *(Park, Sungjoon, et al. "KLUE: Korean Language Understanding Evaluation." arXiv preprint arXiv:2105.09680 (2021))

2. 프로젝트 팀 구성 및 역할

- 박마루찬 (팀장) : environment, AEDA
- 문석암 (팀원) : 데이터 분석, model modularize, TATP
- 박아멘 (팀원) : dataset, EDA
- 우원진 (팀원) : model modularize, Random Oversampling, Train Dataset 구조 만들기
- 윤영훈 (팀원) : dataset, Typed Entity Marker(punct), Entity Swap Augmentation
- 장동건 (팀원) : environment, TATP
- 홍현승 (팀원) : dataset, 입력데이터 구조 만들기

3. 수행한 미니 프로젝트

- Gantt Chart



- 분석
    - 탐색적 분석
        - `subject_entity` type 은 `per`, `org` 로 2 가지로 분류된다.
        - `relation` 는 `subject_entity`의 entity 에의해 단체 또는 사람으로 시작한다
            - 단, 데이터에서 예외가 존재
                1. 잘못 표기된 경우
                    - 소녀시대 등의 org Label 이 per 인경우
                    - 도시명 등의 지명이 subject_entity 로 나와 `subject_entity` 가 잘못 나온 경우
            - 다른 예외 사항은 확인하지 못함.
    - 접근 방법, 발상
        - 논문에서 좋은 결과를 낸 모델을 선정.
        - 기존의 Pretraining 과 최대한 비슷한 입력을 주기 위한 입력 형태 변형
        - 데이터 불균형 해결 및 증강을 위한 어그멘테이션

- 모델 선정
    - 제공된 모델 중 접근이 쉽고, 관련 논문에서 성능이 좋았던 모델을 선정하였다.

- KLUE 벤치마크에서 제시한 파라미터들을 통해 결과를 구현하였다.

- 적용 사항

1. Typed Entity Marker (punct)

- **An Improved Baseline for Sentence-level Relation Extraction** 논문을 기반으로 진행
- data 로 주어지는 sentence 에 entity 의 type 과 함께 entity marker 를 추가하되 marker 로 special token 대신 문장부호를 사용하는 방법이다.
- 성능이 향상되었다. entity type 과 함께 entity 를 mark 하여 모델에게 더 많은 정보를 제공해줌으로써 성능이 향상된 것으로 여겨진다.

2. Adaptation

- 방법 설명 : pretrain 된 모델(ex bert 등)을 Masked language modeling 을 통해 사전학습 시킨 후 Finetuning 하는 방법이다.
- 적용 사유 : Domain 또는 Task 에 대한 언어 모델을 한번 더 학습 시키는 것이 당연히 더 각 단어간 관련도가 높을 것이라 생각했다. (참고 Don't Stop Pretraining: Adapt Language Models to Domains and Tasks)

- 적용 방안
  - 우선 해당 Task 에서는 Domain 을 한정할 수 없어서 DATP 는 사용이 힘들 것이라 판단하였다.
  - Dataset 을 통한 TATP(Task-Adaptive PreTraining)을 진행
- 적용 결과
  1. Train set 을 통한 진행
     - Submission F1 score 가 오히려 떨어짐
  2. Train + Test set 을 통한 진행
     - Submission F1 score 가 증가

3. EDA

- 기존의 EDA 에서는 동의어를 넣는 방식으로 학습 데이터를 확보하였다. 이와 유사하게 동의어는 아니더라도 `per`, `dat`, `org`, `loc`의 type 에 대해 `subject entity`와 `object entity`의 단어를 대체했다.
- 성능은 오히려 하락했다. 아무래도 문장 구조가 주요 단어를 제외하고는 동일해서 생기는 과적합 문제로 보인다. 특히 원래 라벨이 적었다면 생성된 동일 구조의 문장이 많기 때문에 해당 라벨을 더 엄격하게 따지는 경향성을 보이게 되어 오히려 점수가 하락한 것으로 보인다.

4. AEDA

- 임의의 문장부호를 입력 문장의 단어 사이에 무작위로 삽입하였다.
- 문장부호 : (`,` `.` `;` `:` `?` `!`)
- 총 삽입 갯수 : 원본 문장의 단어 수 대비 0%(미적용과 동일) ,30%, 100%로 실험했다.
- 결과 : 유의미한 차이가 발생하지 않았다.
- 논문 결과와 실험 결과를 볼 때, 더 적은 데이터셋에 대해 효과있는 방법으로 보임.

5. Entity Swap Augmentation

- object entity 를 subject entity 로 subject entity 를 object entity 로 바꾸어도 30 개의 label 중 해당하는 label 이 있는 data 들에 대해서만 entity swap 을 진행하여 데이터를 늘리는 방법이다.
- 데이터가 상대적으로 많은 label 에 대해서 진행할 경우 데이터 불균형이 더 심해질 것이므로 데이터가 상대적으로 부족한 label 들에 대해서만 진행했다.

6. Random OverSampling

- 수가 적은 label 에 대해서 데이터를 중복으로 넣어 데이터 불균형을 극복하려 했다.
- 성능은 오히려 하락했다. 데이터가 적은 라벨이 중복이 많이 되며 해당 데이터에만 overfitting 됐던 것이 아닌가라는 생각을 했다.

7. train 데이터 구조 만들기

- BERT 는 Pretraining 할때 2 개의 문장에 대한 관계를 `[CLS] `토큰을 통해 추론하는 방식이므로, 이번 Relation Extraction task 에서는 `[CLS] original Sentence [SEP] 이 문장에서 {Subject Entity}와 {Object Entity}와의 관계는 무엇일까? [SEP]` 와 같은 구조로 Data 를 구성하여 실험을 진행하였다.

- BERT 는 Pretraining 할때 2 개의 문장에 대한 관계를 `[CLS]` 토큰을 통해 추론하는 방식이므로, 이번 Relation Extraction task 에서는 `[CLS] {Subject Entity} [SEP] {Object Entity} [SEP] 관계 [SEP] 정보 Sentence [SEP]` 와 같은 구조로 Data 를 구성해서 관계와 cls 토큰이 attention 을 학습하도록 실험을 진행하였다.

8. hyperparameter 실험적 결과 선정

4. 최종 프로젝트 결과

- 성능 향상에 도움이 되었던 적용
  - Typed Entity Marker (punct), TAPT, Entity Swap Augmentation, train 데이터 구조 만들기
  - soft voting ensemble (best model 에 대해 1.2 배 가중치 반영)
- 최종 모델 (이하 모델 앙상블)
  - 72.710 (TAPT+ added data + entity marker, tokenization modify)
  - 73.950 (TAPT+ entity marker, tokenization modify)
  - 74.034 (added data + entity marker + tokenization modify + k-fold)
  - 72.991 (entity marker + tokenization modify)
  - 70.724 (TAPT (epoch 30)+ added data + entity marker, tokenization modify)
- 최종 점수
  - Public micro_f1 score : 75.962
  - Private micro_f1 score : 73.794

5. 자체 평가 의견

- 잘한 점
  - 깃허브 사용이 만족스러웠다. -> 프로젝트 관리가 되는 느낌이였다.
  - task 관련 paper 를 기반으로 실험을 분할 진행한 것.

- 아쉬웠던 점들
  - 실험 관리 좀 불편했음 (이름 짓기, wandb 에서 내 모델 찾기)
    - arg 가 너무 많아서 오히려 힘들었음..
    - 다음에는 <김캠퍼> 1, 2, 3, 4, 5 <이부캠> 1,2,3,4 등으로 작성해보자.
  - 모델을 분할 하여 Task 를 나누는 방법을 고려하였으나 실제 적용하지 못한 점이 아쉽다.
    - 생각한 Task 분할은 데이터의 `subject_entity`가 2 가지이며 `no_relation`,'org','per' label 의 비율이 매우 비슷하여 3 가지로 분류 후 'org','per'은 각자 Task 를 한번 더 진행하는 것을 생각해봄
  - 모델 성능 향상에 기여할 수 있는 방법(논문)을 직접 찾지 못한 점.
  - tokenizing 부터 커스텀 모델 작성하기.
  - BERT 모델 위에 층을 더 쌓아서 실험해보지 못한것.
  - wandb 를 좀 더 체계적, 실용적으로 사용하지 못한 점.
  - hyperparameter 최적화 툴을 사용해보지 못한 점.