부스트캠프 랩업 리포트 가이드

- 리더보드 제출 내역 (tsv 파일)
  - 대회 종료 후 AI Stages 대회 리더보드 **- My Record** 페이지에서 다운 받으실 수 있습니다.
  - 파일명은 submit_record.tsv 로 통일해주세요.
- 최종 소스코드 (주피터 노트북 파일 혹은 스크립트 파일)
  - 소스코드 구성에 제한은 없습니다.
  - README.md 파일을 통해 소스코드에 대한 설명까지 첨부한다면 더욱 좋습니다.
- 랩업 리포트 글 (word 혹은 pdf 파일)
  - 파일 명은 wrap_up 으로 통일해 주세요.
  - 아래 가이드를 참고하시어 작성해 주세요.

<기술적인 도전>

본인의 점수 및 순위

LB 점수 80.2%, 32등 (반드시 포함해 주세요)

- 검증(Validation) 전략
  1. 제공된 데이터셋의 맨 앞부분 0.1 ratio를 validation set으로 사용했습니다.

- 사용한 모델 아키텍처 및 하이퍼 파라미터
  1. 아키텍처: xlm_roberta_large
     a. LB 점수 : 80.2
     b. training time augmentation
        i. 토론게시판(http://boostcamp.stages.ai/competitions/4/discussion/post/174)에 올라온 추가데이터를 사용
        ii. 토론게시판(http://boostcamp.stages.ai/competitions/4/discussion/post/177)에 올라온 추가데이터도 같이 사용.
     c. 추가 시도
        i. 초기에 시작할 때 autoML로 찾은 최적의 파라미터 batch_size 32, epoch 4로 잡았을 때 LB 75.3%. 후에 batch_size 64, epoch 9로 늘려서 LB 76.8%로 상승
        ii. loss를 cross entropy * 0.75 + focal * 0.25 로 섞어서 사용. LB 77.2%로 상승

iii. 토론게시판에 올라온 파라미터를 참고하여 batch_size 64, epoch10, warmup_step 500->300 으로 변경. LB 78.7%로 상승.
iv. 단어의 entity가 붙은 ner 데이터셋 사용. 그냥 데이터 사용했을 때의 점수 LB 78.7%에서 LB 77.5% 로 오히려 하락. 나빼고 다 잘된다는거 보면 뭔가 잘못된듯.
v. 단어 앞뒤로 특정한 기호(^, ` 등)만 붙여도 점수가 오른다고 해서 시도. LB 0.1% 하락.
vi. 토론게시판의 추가데이터를 조금만 조정해서 약 100000 (십만)개의 데이터 사용. 훈련데이터는 9000개이니 당연히 점수 LB 63.3%로 대폭 하락.
vii. 그래서 조금만 추가하기로 해서 원래 훈련데이터 비율에 맞춰서 추가. 0.7 비율만큼 추가했을 땐 LB 76.0%, 0.3 비율만큼 추가했을 땐 LB 78.8%. 0.4비율은 못내봤는데 val acc이 0.8이 넘었음. 근데 제출기회 때문에 따로 LB는 알수없음.
viii. 외부데이터 훈련 0.4비율로 추가한거 + 0.3비율로 추가한거 + 그냥 파라미터 조절 잘해서 나온거 + 특정기호 붙인거 앙상블로 LB 80.2%로 상승.

- 앙상블 방법
  1. (0.3비율모델) + (0.4비율모델 * 1.1) + (최적의 파라미터) + (앞뒤 특정기호). 그냥 덧셈 사용. 1.1를 곱한건 저 0.4비율모델의 val acc이 제일 잘 나왔기 때문.

- 시도했으나 잘 되지 않았던 것들
  1. 토론 게시판(http://boostcamp.stages.ai/competitions/4/discussion/post/216)에 pororo library를 이용하여 각 단어에 entity를 적은 데이터, 즉 ner 데이터를 제공해주었으나 난 성능이 하락함. 데이터를 보니까 띄어쓰기가 논문이랑 다르게 되어있는것 같아서 직접 만들려고 했으나 7시간 걸려서 안만듬.
  2. 외부 추가데이터를 그대로 넣어서 했더니 엄청난 성능 하락. 원래 훈련 데이터에 있는 label 비율 때문인 듯. 잘 조절하니 성능 향상.
  3. 단어 앞 뒤에 잘 안쓰이는 기호 붙여서 표시해도 약간 하락.
  4. 한번 잘 나온 환경 보고 그 val acc이 제일 잘나온 checkpoint를 저장하기 위해 똑같은 환경에서 다시 시도하면 절때 다시 안나와서 속이 터졌다.
  5. electra tokenizer가 UNK가 제일 적게 나온다고 해서 시도해봤는데 성능 하락. 그냥 해당 모델에 맞는 tokenizer 쓰는게 나은듯.

<학습과정에서의 교훈>

1. 실험할 때 무거운 모델로만 실험하지 말고 가볍고 적당히 성능있는 모델로 실험하되 데이터를 손보면서 성능 올리는 쪽으로 하는게 좋음. 그래야 빨리 결과 나와서 이것저것 많이 해볼 수 있어서 시간적으로 이득. 마지막 제출 전에만 무거운 모델로 바꾸면 되니까. 정말 좋은 방법인듯.

2. 외부데이터 쓸 때는 원본 훈련 데이터의 label 비율을 맞춰주자.

3. 마지막에는 val data도 모두 훈련데이터로 쑤셔 넣으라고 했는데 제출제한이 있어서 못함.

<마주한 한계와 도전숙제>

아쉬웠던 점들

1. 제출제한으로 제일 잘나온 모델(0.4비율)을 단일로 내놔서 앙상블의 효과를 직접 비교 못해봤다는게 아쉬움.
2. 나도 가벼운 모델로 이것저것 실험해보고 나중에 무거운걸 하는게 좋을듯.
3. 나도 scikit-learn에서 제공하는 strate 어쩌구 k-fold 하는걸 사용해보자. 성능이 정말 잘 나온다고 한다.

한계/교훈을 바탕으로 다음 스테이지에서 새롭게 시도해볼 것

1. 가벼운 모델로 실험하자.