Github : https://github.com/Aroma-Jewel/KERC-2022-4th

2022 제 4회 한국어 감정인식 경진대회 솔루션 발표

(Team 아로마쥬얼)

이재학(아주대학교)
정유석(티맥스 에이아이)

TABLE OF CONTENTS

1. Introduction

안녕하세요. 아로마쥬얼 팀 입니다.

아주대학교 수학과 졸업 후 NLP를 같이 공부하는 친구들이 모인 팀입니다.

이재학

github.com/wogkr810

아주대학교 수학과
네이버 부스트캠프 AI Tech 3기 NLP
취준생

정유석

github.com/j961224

아주대학교 수학과
네이버 부스트캠프 AI Tech 2기 NLP
티맥스에이아이

대회 개요

대회 기간

2022.09.01 ~ 2022.10.23

대회 설명

대화 텍스트 데이터를 분석하여 한국인의 감정을 예측하는 한국어 자연어처리 인공지능 모델 개발

평가 방법

Micro-F1 Score

Micro F1 점수는 정밀도(Precision)와 재현율(Recall)의 조화 평균으로 정의됩니다.

$$[\ \text{Micro F1-score} = 2 * \frac{\text{Micro-precision} * \text{Micro-recall}}{\text{Micro-precision} + \text{Micro-recall}}\ ]$$

데이터셋

데이터 예시

Sentence_id	person	sentence	scene	context
1	재학	커피가 맛이 이상한데?	S0001	행궁동 카페 안에서

데이터셋 설명

- 수상한 삼형제(한국 드라마)의 1513개 scene에서 추출한 총 12289개의 대화 텍스트 데이터
- 데이터 종류
  - train : 7339
  - public : 2566
  - private : 2384
- 데이터 column
  - sentence_id, person(speaker), sentence, scene_id, context(scene description), label(train_only)
- Labels(Train_only)
  - Euphoria : 신체적, 정서적으로 "행복한, 유쾌한, 의기양양한, 재미있는, 편안한"
  - Neutral : "잔잔한, 차분한, 보통인, 안정적인"
  - Dysphoria : "불행한, 불만족한, 안절부절하지 못한, 낙담한, 우울한, 좌절한, 불안한"

2. Motivation

Motivation

- 데이터셋의 다양한 column을 어떻게 활용할 수 있을까?
- 각 sentence마다 맞춤법이 맞지 않은 경우는 어떻게 처리할까?
- 모델이 어떻게 더 효율적으로 학습하도록 할까?
- 문장의 정보를 어떻게 최대한 활용할까?

3. Proposed Method

전처리(Data Input)

데이터 예시

Sentence_id	person	sentence	scene	context
1	재학	커피가 맛이 이상한데?	S0001	행궁동 카페 안에서

Input 구성

[context] context [context] past_sentence [SEP] target_sentence

- [context]
  - 스페셜 토큰이 아닌 텍스트(word)
  - NaN인 경우 추가하지 않음.
- past_sentence
  - get_scene_context 함수로 얻은 target_sentence 이전의 5개의 과거 문장 사용
- [SEP]
  - tokenizer의 sep_token으로 target_sentence 분리
- target_sentence
  - 예측 data의 sentence

전처리(Data Input)

각 문장 구성

- past_sentence :
  - "other_person" + " " + "other_person_sentence" + " "
- target_sentence :
  - [SEP] + " " + "target_person" + " " + "target_sentence"

Input 예시

context : O / past : O

[context]경찰들에 둘러싸여 축하받는 이상. [context] "순경" 뭐? "과자" 그냥 넘어가요. 혼자 지껄여 봤네요. "현찰" 아부지, 저 가봐야겠는데요. "순경" 중요한 일있다 그랬지? "과자" 잘 가. [SEP] "순경" 한방 박았으면 됐지 뭘 더박아?

context : NaN / past : O

"건강" 일하지 뭐해. "과자" 어지간하면 왔다가지. 아부지 한 소리 하시드라. [SEP] "건강" 죄송해요.

context : O / past : X

[context]과일 들고 안방으로 들어가려다 멈추는 우미 [context][SEP] "과자" 지가 어디 가서 우리 현찰이 같은 남편을 만나?

context : NaN / past : X

[SEP] "어영" 야! 전화 받아. 아무리 바빠도 내 전화는 받아야 되는 거 아냐? 약속 하나도 못 지키는 주제에 법을 지켜?

전처리(데이터 정제)

py-hanspell 한글 맞춤법 검사를 위한 파이썬 패키지

아부지가방에들어가신다. → 아버지가 방에 들어가신다.

PyKoSpacing 한글 자동 띄어쓰기를 위한 파이썬 패키지

아버지가방에들어가신다. → 아버지가 방에 들어가신다.

전처리 함수

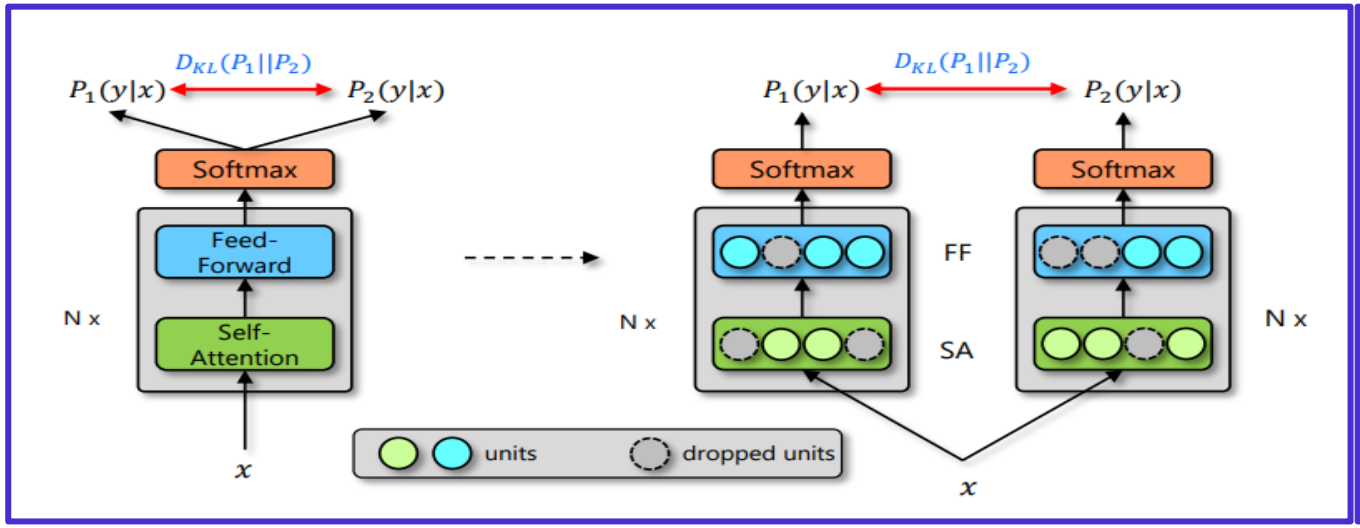
- py-hanspell & PyKoSpacing 라이브러리 통과 후 다른 것들
- 토크나이즈 이후 ['UNK'] 토큰 존재하는 경우
- 예시
  - '이여대' -> '이화여자대학교'
  - '느이' -> '너희'
  - '지껄여' -> '말해'

R-drop

같은 input x를 모델에 두 번 통과시켜주면, Dropout으로 각기 다른 모델에 통과하는 것과 비슷한 효과.

두 개의 output distribution의 KL-divergence를 줄이는 방향으로 학습

$$\mathcal{L}^i = \mathcal{L}^i_{NLL} + \alpha \cdot \mathcal{L}^i_{KL} = -\log \mathcal{P}^w_1(y_i|x_i) - \log \mathcal{P}^w_2(y_i|x_i) + \frac{\alpha}{2}[\mathcal{D}_{KL}(\mathcal{P}^w_1(y_i|x_i)||\mathcal{P}^w_2(y_i|x_i)) + \mathcal{D}_{KL}(\mathcal{P}^w_2(y_i|x_i)||\mathcal{P}^w_1(y_i|x_i))].$$



Rank	Model	ROUGE-1	ROUGE-2	ROUGE-L	Extra Training Data	Paper	Code	Result	Year	Tags
1	BRIO	47.78	23.55	44.57	X	BRIO: Bringing Order to Abstractive Summarization			2022	
2	GLM-XXLarge	44.7	21.4	41.4	✓	GLM: General Language Model Pretraining with Autoregressive Blank Infilling			2021	Transformer
3	BART + R-Drop	44.51	21.58	41.24	X	R-Drop: Regularized Dropout for Neural Networks			2021	Transformer

R-Drop: Regularized Dropout for Neural Networks (NeurIPS 2021)

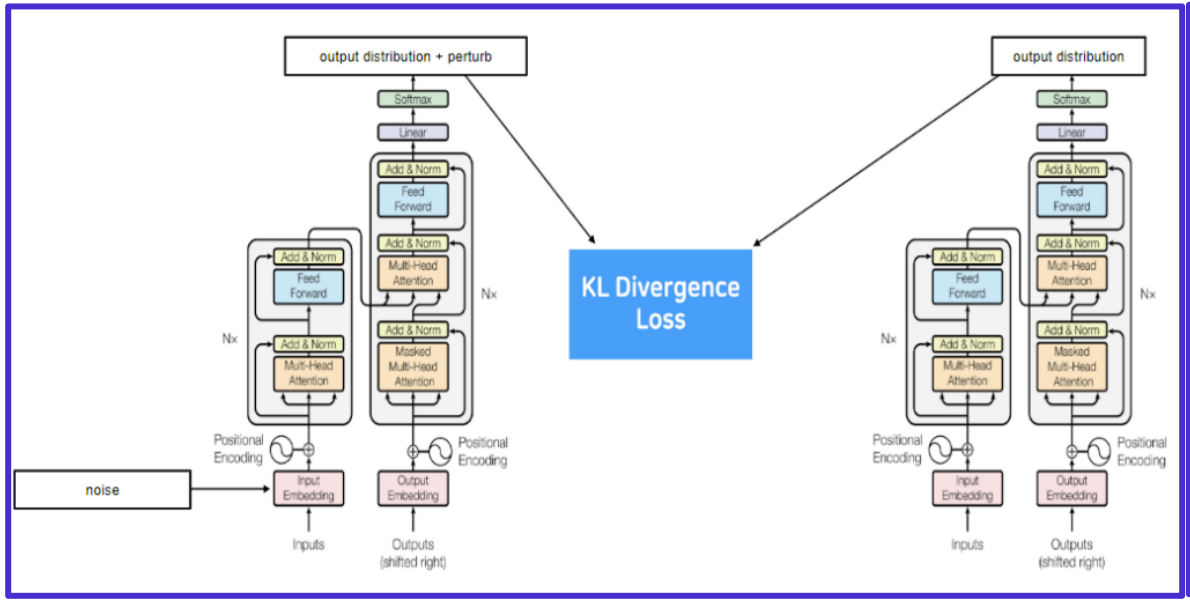
Smart loss

계산한 loss를 이용하여 noise에 대한 gradient를 구하여 noise update

1. 입력에 noise를 더해서 추론한 output_preturb
2. 평범한 입력으로 추론한 output

두개의 output에 대해서 kl-divergence로 loss 계산

각 output의 kl-divergence를 줄이는 방향으로 학습



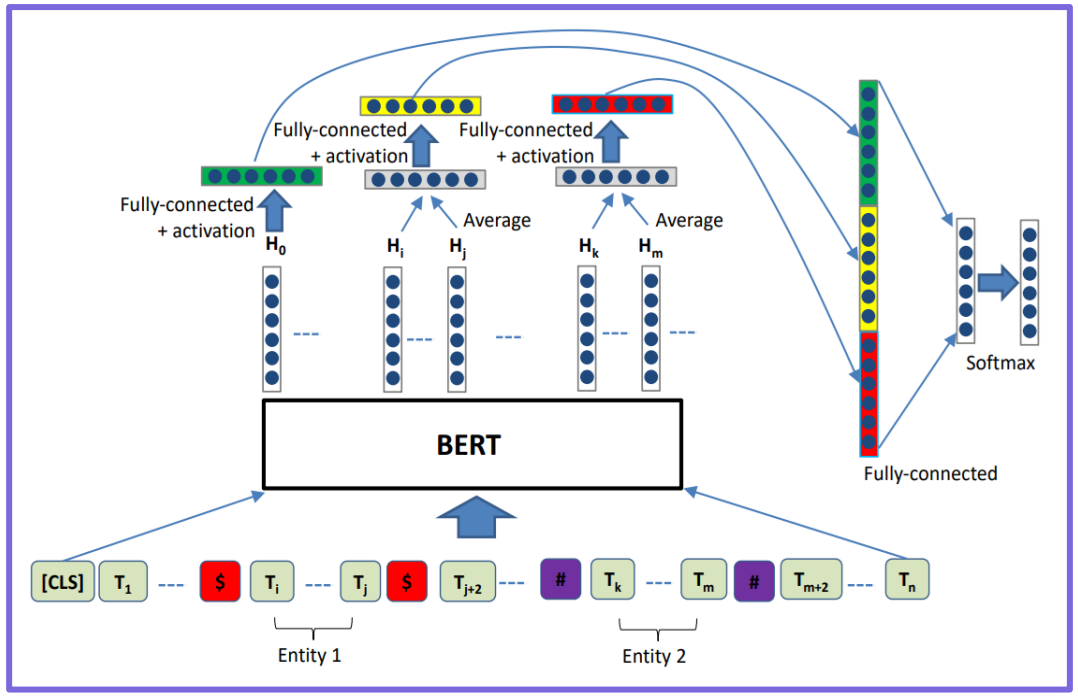
1	SMART-RoBERTa Large	97.5	SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization		2019	Transformer
2	T5-3B	97.4	Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer		2019	Transformer
3	MUPPET Roberta Large	97.4	Muppet: Massive Multi-task Representations with Pre-Finetuning		2021	
4	ALBERT	97.1	ALBERT: A Lite BERT for Self-supervised Learning of Language Representations		2019	Transformer

SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization (ACL 2020)

모델 설명

R-Roberta

- klue/roberta-large pretrained model을 이용
- Past sentence와 target sentence 사이에 relationship을 정의하기 위한 모델



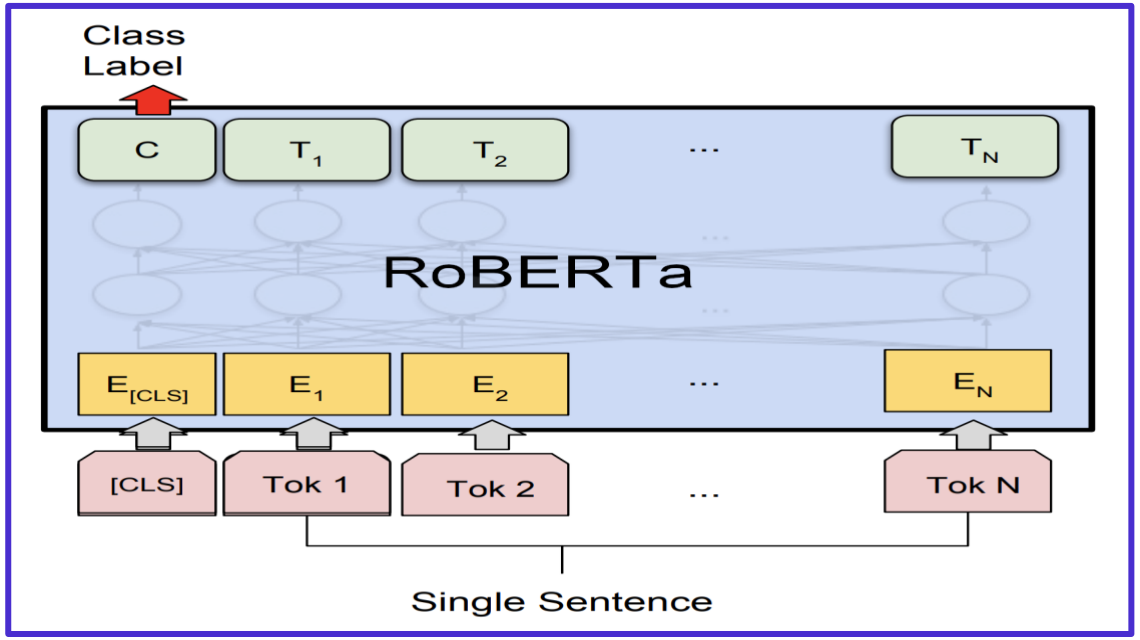
Method

- ROBERTA로부터 3개의 vector를 획득
  - [CLS] token vector
  - Past sentence에 대한 averaged vector
  - Target sentence에 대한 averaged vector
- 각 vector는 fully-connected layer들을 통과
  - dropout -> tanh -> fc-layer
- 3개의 vector를 concat합니다.
- 3개의 concat한 vector를 fully-connected layer를 통과하여 분류
  - dropout - > fc-layer

모델 설명

Roberta + classification layer(fc layer)

- Klue/roberta-large pretrained model을 이용
- Roberta 모델 끝 단에 감정 분류를 위해 classification layer를 붙여 Fine-tuning 진행

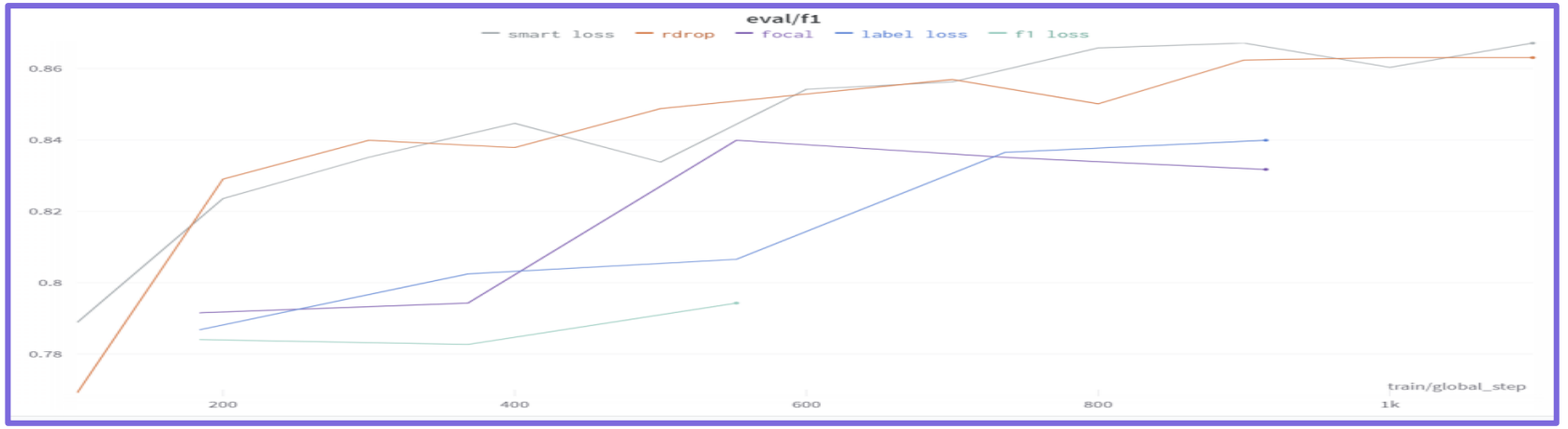


4. Experiments and Discussion

Experiments

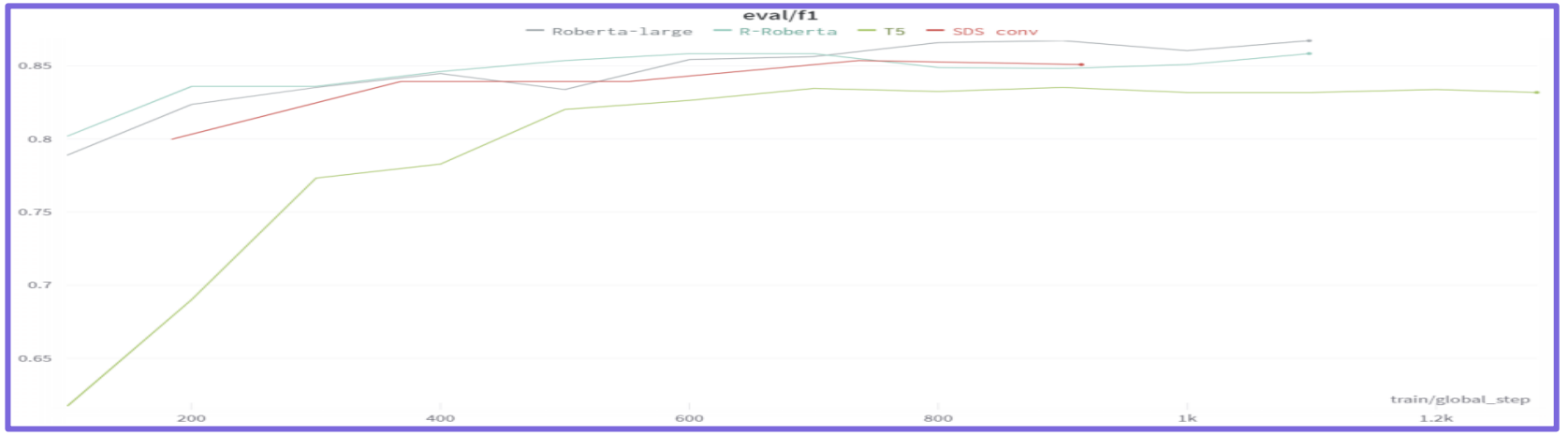
Loss 실험

F1 loss & **Smart loss & R-drop loss** & Focal loss & Label smoothing loss



Model 실험

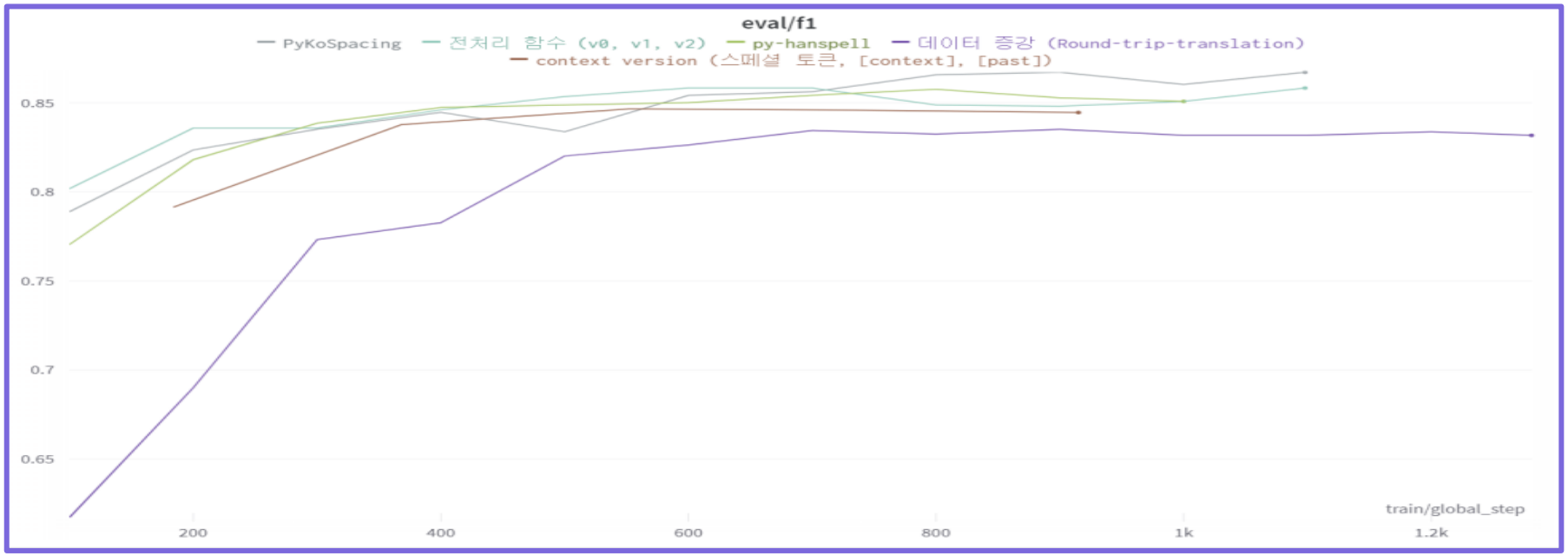
Roberta-large & R-Roberta & SDS conv & T5



Experiments

전처리 실험

- py-hanspell
- **PyKoSpacing**
- **전처리 함수 (v0, v1, v2)**
- 데이터 증강 (Round-trip-translation)
- **context version (스페셜 토큰, [context], [past])**



Experiments

최종 실험 결과

사용한 기법	Fold 유무	Private Score
Klue/roberta-large + R-drop + 전처리 함수 (v0, v1)	X	0.77436
Klue/roberta-large + Focal loss + PyKoSpacing	X	0.77241
R-Roberta + Smart loss + 전처리 함수 (v0, v1)	X	0.77358
Klue/roberta-large + Smart loss + 전처리 함수 (v0, v1)	O	0.78215
R-Roberta + focal loss + 전처리 함수 (v0, v1)	O	0.77397
Klue/roberta-large + Smart loss + PyKoSpacing	O	0.78943

최종제출(SOTA)

6개 제출 결과 Hard voting Private Score: 0.79027

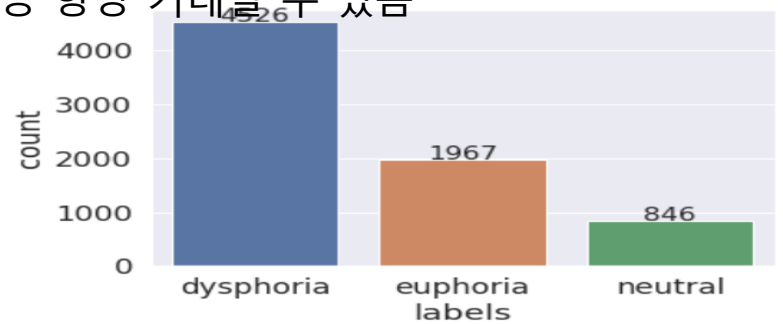
5. Conclusion and Future Works

Conclusion

- **대화체 텍스트 데이터를 활용한 모델 고도화**
  - Public LB : 0.79345
  - Private LB : 0.79027
- **단문이 아닌 대화의 흐름이 있는 데이터 활용**
  - Kakao Pororo & Naver API는 단문 감정분류
  - 우리의 데이터는 scene & context에 따라 대화가 연속적인 흐름으로 진행되므로 '네.' 같은 단문도 다른 감정으로 분류 할 수 있음
- **Text Sentiment Analysis의 SOTA 기법을 한국어 대화체 데이터셋에 적용**
  - Smart loss & klue/roberta-large 모델을 활용하여 성능 향상

Future Works

- **GPT기반 생성모델로 context 생성**
  - 총 12289개의 데이터의 unique한 context는 1513개
  - 총 12289(7339+2566+2384)개의 데이터 중 3655(2017+771+867)개의 'NaN' context 존재(약 30%)
  - context를 input에 활용하면서 모델의 성능이 향상되었기에, NaN인 context를 생성하면서 모델의 성능 향상을 기대할 수 있음
- **TAPT(Task Adaptive Pre-Training)**
  - 대회 데이터셋은 Klue/roberta-large가 기학습한 데이터셋과 다른 문장 형태의 데이터셋
  - 그에 따라 TAPT를 통해 모델이 대회 데이터셋을 사전학습하여 모델 성능 향상을 기대할 수 있음
- **외부데이터 수집**
  - EDA에 따르면, sentence의 길이가 길어질수록 'dysphoria'일 경우가 많음
  - 라벨별 균형도 'neutral' < 'euphoria' < 'dysphoria' 순으로 굉장히 불균형함
  - 드라마 대본을 정제하여 감정을 라벨링 한다면, 데이터 개수를 증가시킴으로써 성능 향상 기대할 수 있음



감사합니다!

Q & A