

딥 러닝 모델을 이용한 DKTC 분류

DKTC 분류 과제 개요

- DKTC는 협박, '갈취', '직장 내 괴롭힘', '기타 괴롭힘' 등 4개 클래스 각 1천 개 정도씩의 대화로 이루어진 데이터
- 이번 DLthon에서는 테스트 데이터에 대해서 위의 4개 클래스 중 하나로 분류하는 과제를 수행

데이터 개요

협박 대화	896
갈취 대화	981
직장 내 괴롭힘 대화	979
기타 괴롭힘 대화	1094
총 데이터	3950

딥러닝 모델

- 3종류의 모델을 사용
 - Bidirectional LSTM
 - Transformer Encoder
 - KoBERT-v1: SKT(2021)

데이터 전처리

- 영문자와 한글, 숫자, 문장부호(?!,.)를 제외한 문자 제거
- 토큰나이징
 - LSTM, Transformer: konlpy의 Mecab 형태소 분석기를 사용해 토큰화
 - KoBERT: 자체 토큰나이저 사용

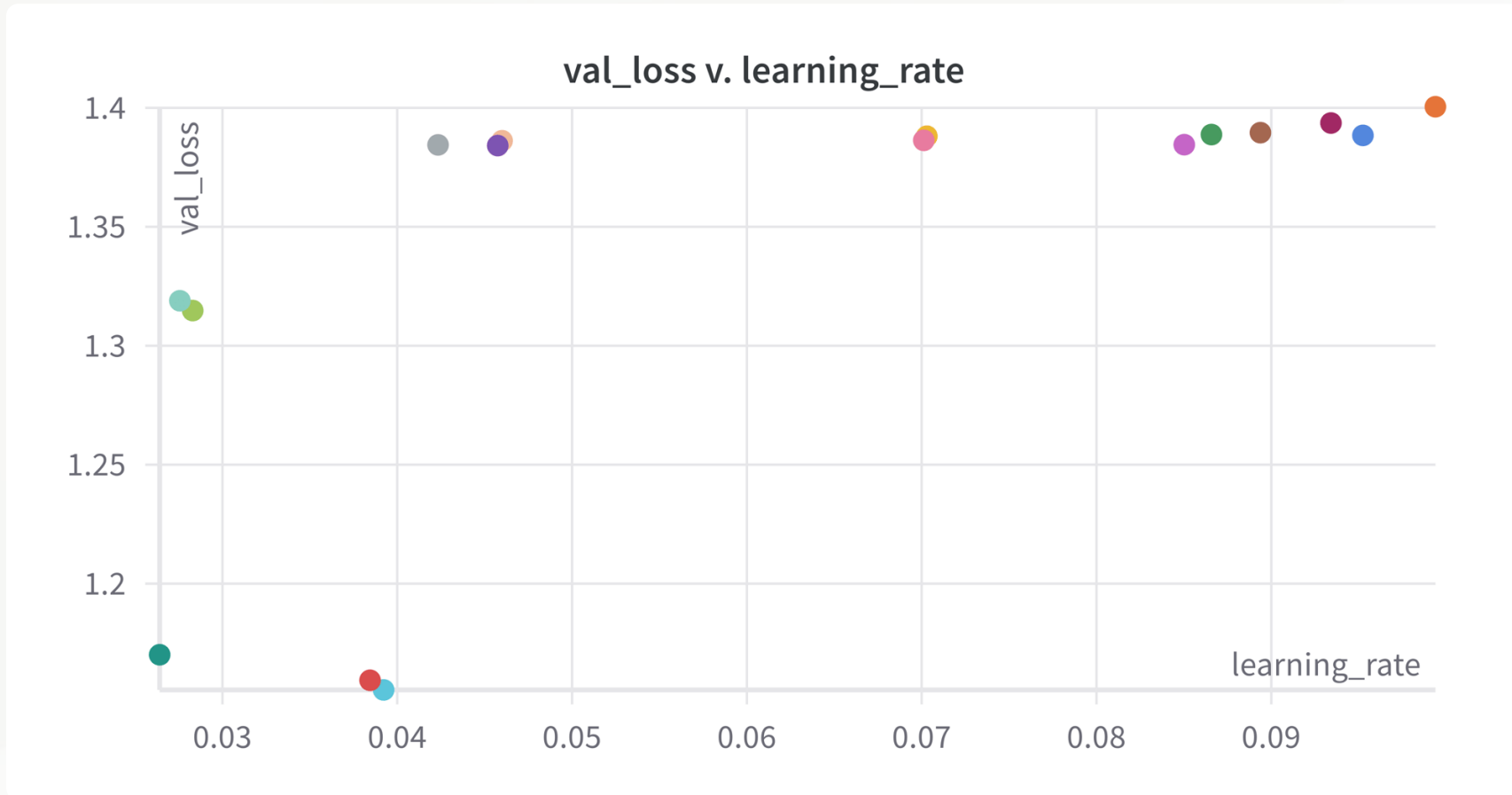
임베딩 레이어

- LSTM과 Transformer 모델에서 3종류의 임베딩 레이어를 사용
 - 초기화된 임베딩 레이어를 처음부터 학습
 - word2vec으로 학습시킨 임베딩 레이어를 고정하여 사용
 - word2vec으로 학습시킨 임베딩 레이어와 사전 학습된 임베딩 레이어를 병합하여 사용
- KoBERT: 8000개 단어로 구성된 자체 임베딩 레이어 사용

W&B를 이용한 학습 과정 시각화

- 최적 learning rate와 epoch를 찾기 위해 W&B 패키지를 사용
- Validation loss/accuracy와 Test accuracy를 보여준 learning rate를 최종 모델에서 채택
- KoBERT 모델의 경우 캡슐화 되어 있어 W&B를 적용하지는 못함

W&B를 이용한 학습 과정 시각화



W&B를 이용한 학습 과정 시각화

Test Accuracy Rate:



모델별 최종 성적

- 초기화된 임베딩 레이어 사용
 - bi-LSTM: 81.25%
 - Transformer: 80.5%
- word2vec으로 학습시킨 임베딩 레이어 사용
 - bi-LSTM: 80.5%
 - Transformer: 80.5%
- word2vec finetuning(박규병님 공개 w2v 모델에 train data 병합하여, 새로운 w2v 생성)
 - bi-LSTM: 74.5%
 - Transformer: 85.5%
- KoBERT: 85.75%

한계 및 개선점

- W&B로 learning rate, epoch 이외에 더 다양한 하이퍼파라미터를 변경해보지 못했다.
- 하이퍼파라미터만 조정하는 것 외에 구조를 다른 방법으로 설계한 모델을 시도해보지 못했다.
- 아직 학습하지 못한 pytorch 프레임워크로 캡슐화된 모델을 이용한 결과로, 내부 구조를 충분히 파악하지 못하여 W&B나 그리드 서치 등을 활용한 별도의 하이퍼파라미터 커스터마이징 전략을 시도해보지 못했다.