Github Repository 링크 : <https://github.com/bcaitech1/p3-dst-freshtomato>

Private(JGA) : 0.6791, 18등 (팀 4등)

1. EDA

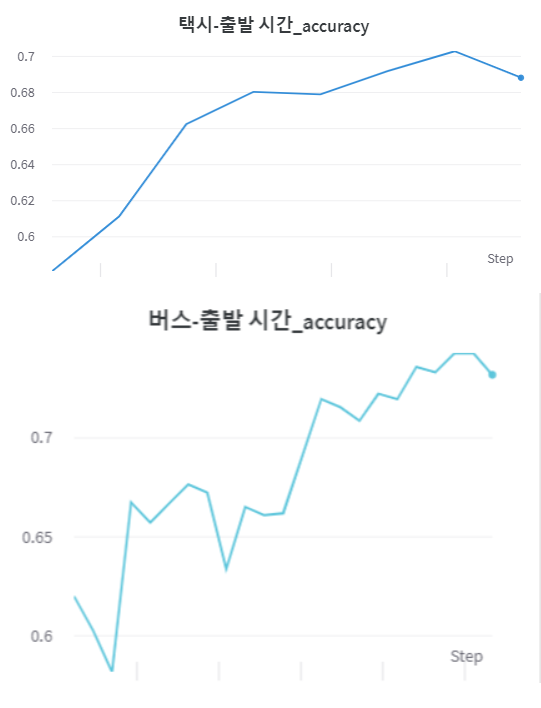
0.1 User Utterance, System Utterance, Label(Target Value)

WordCloud를 통한 키워드 출현 빈도수 시각화 시도 (아래의 예시로는 user utt에 관한 시각화 결과만 첨부)



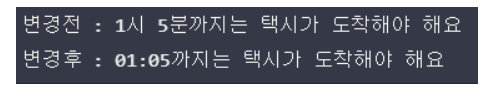
1. Data Preprocessing

‘택시’ 도메인 내의 ‘시간’과 관련된 slot에 대한 정확도가 좋지 않았는데, 혹시 택시라는 단어에 ‘시’가 포함되어 있어 모델이 헷갈리는 것이 아닐까라는 생각을 하였고, 택시라는 도메인의 dialog들을 버스라는 도메인으로 바꾸어주었습니다



성능의 향상이 조금 있었지만, 이는 어떤 가설검정의 결과가 아니다보니 세웠던 가설이 참이라고 생각할 수는 없을 듯 합니다.

또한, 시간의 dialog 내의 형태도 전처리를 해주기도 하였습니다.

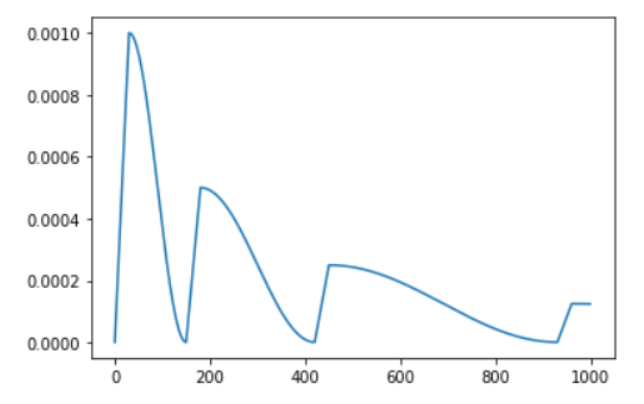


1. Training 보조 Technique

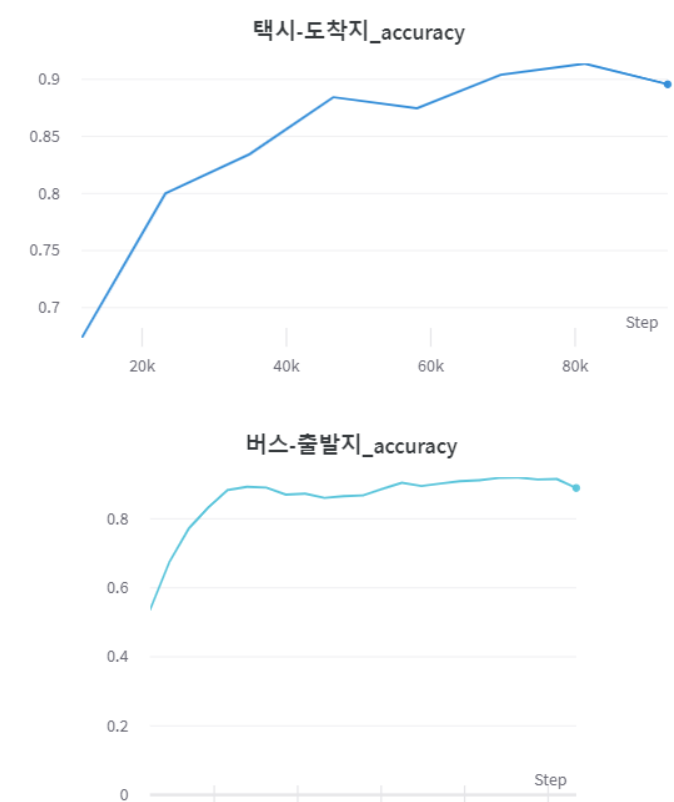
2.1 Customized LR Scheduler

설명:

<https://github.com/bcaitech1/p3-dst-freshtomato/blob/main/notebooks/Plot_Manual_Scheduler.ipynb>



2.2 domain-slot별 accuracy 확인



1. 모델 아키텍쳐

3.1 개인적으로 시도해본 모델 구조

- all data에 fine tuning

- CHAN-DST

* 1. 팀끼리 같이 시도해본 모델 구조

5-Fold Ensemble

TRADE 이용 PLM encoder + 5 gate or 3 gate 돌리기 – (PretrainedModel = koelectra)

TRADE 이용 PLM encoder + 5 gate or 3 gate 돌리기 – (PretrainedModel = dsksd/bert-ko-small-minimal)

all data + coco augmentation(10epochs) 결과 data에 fine tuning

SUMBT 모델과 TRADE 모델의 앙상블

STAR-DST

SOM-DST

LabelSmoothingLoss

1. 회고록

Ontology 기반이건 Open-Vocab 기반이건 DST Model들 간에 아키텍쳐가 확연히 달라서 새로운 모델을 공부할 때 유의하게 들여다볼 필요가 있다는 점, 보다 중요한 부분을 잘 들여다보고 현재 TASK에 적용해볼 수 있는 방법부터 파악할 수 있어야겠다는 점을 느낄 수 있었습니다. DST라는 Task 자체가 어렵다보니 혼자 해내려고 시도했다면 포기하게 될 수도 있었을 것 같지만, 좋은 팀원들 덕분에 매순간 즐겁게 대회에 참여할 수 있었고 팀의 소중함을 느낄 수 있었습니다

만약 이후 Stage나 Competition에 참여함에 있어서 NLP 관련 Task를 수행하게 된다면 (뿐만 아니라 다른 도메인이더라도) 사전에 Prediction 결과를 조금 더 세부적이고 가시적으로 표현하여 확인할 수 있는 특정 과정을 추가하고 적용할 수 있어야 하는 등의 어느 특정 부분에 우선순위를 두어야 할지 깨달을 수 있었습니다.