

Yelp 데이터셋을 이용한 영어 리뷰 감성 분류

1조

양다경, 구선헤, 김유나, 김현수,
김현지, 이용주, 최원서

목차

1. 프로젝트 개요
2. 프로젝트 프로세스
3. 프로젝트 팀 구성 및 역할
4. 프로젝트 결과
 - Pretrained LM 선정
 - Hyperparameter tuning
5. 자체 평가 및 보완점

1. 프로젝트 개요

목표

주어진 베이스라인 코드에서 Accuracy 향상

1. Hyperparameter Tuning 수행 (WandB를 이용한 데이터 기록)
2. 여러 Pretrained LM 탐색 및 적용

프로세스

1. 프로젝트 기획
2. hyperparameter 탐색 및 적용
3. Pretrained LM 탐색 및 적용
4. 보고서 작성 및 발표

기대효과

1. Yelp dataset에 다양한 Pretrained LM을 적용해 각 모델의 성능 비교
2. 99%의 정확도로 Yelp dataset 영어 리뷰 긍·부정 감성 분석

개발 환경



사용 패키지



Transformers



PyTorch

matplotlib

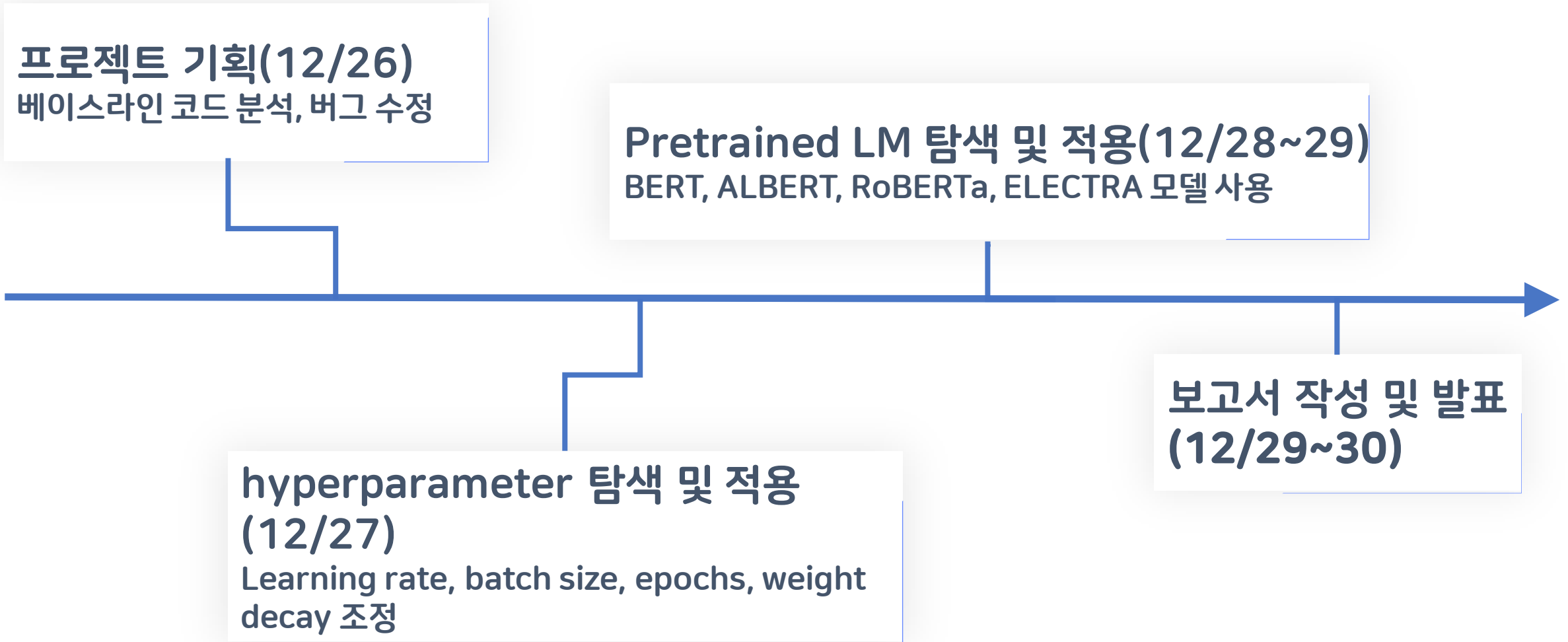


W&B



seaborn

2. 프로젝트 진행 프로세스



3. 프로젝트 팀 구성 및 역할

- 베이스라인 코드 분석 및 정리

- 코드 수정

- Hyperparameter tuning using WandB

- Visualization using matplotlib

- Ensemble

- 발표

양다경

- 베이스라인 코드 분석 및 정리

- 코드 수정

- Small sample of training data

- Hyperparameter 조정 및 WandB를 이용한 데이터 기록 (BERT)

구선혜

- 베이스라인 코드 분석 및 정리

- 코드 수정

- 버그 수정

- Label EDA

- Hyperparameter 조정 및 WandB 시도

- 보고서 작성

김유나

- 베이스라인 코드 분석 및 정리

- Hyperparameter 조정 및 WandB를 이용한 데이터 기록 (RoBERTa)

김현수

3. 프로젝트 팀 구성 및 역할

- 베이스라인 코드 분석 및 정리
- Hyperparameter 조정 및 WandB를 이용한 데이터 기록 (ELECTRA)

김현지

- 베이스라인 코드 분석 및 정리
- 코드 수정
 - seed_everything 함수
- Hyperparameter 조정 및 WandB를 이용한 데이터 기록 (ALBERT)

이용주

- 베이스라인 코드 분석 및 정리
- Hyperparameter 조정 및 WandB를 이용한 데이터 기록 시도 (ELECTRA)

최원서

4. 프로젝트 결과 - Yelp 데이터 소개

train_data	dev_data
sentiment.train.0 + sentiment.train.1	sentiment.dev.0 + sentiment.dev.1
443,259개의 문장 데이터 셋	4,000개의 문장 데이터 셋

	review
0	excellent food .
1	superb customer service .
2	they also have daily specials and ice cream wh...
3	it 's a good toasted hoagie .
4	the staff is friendly .

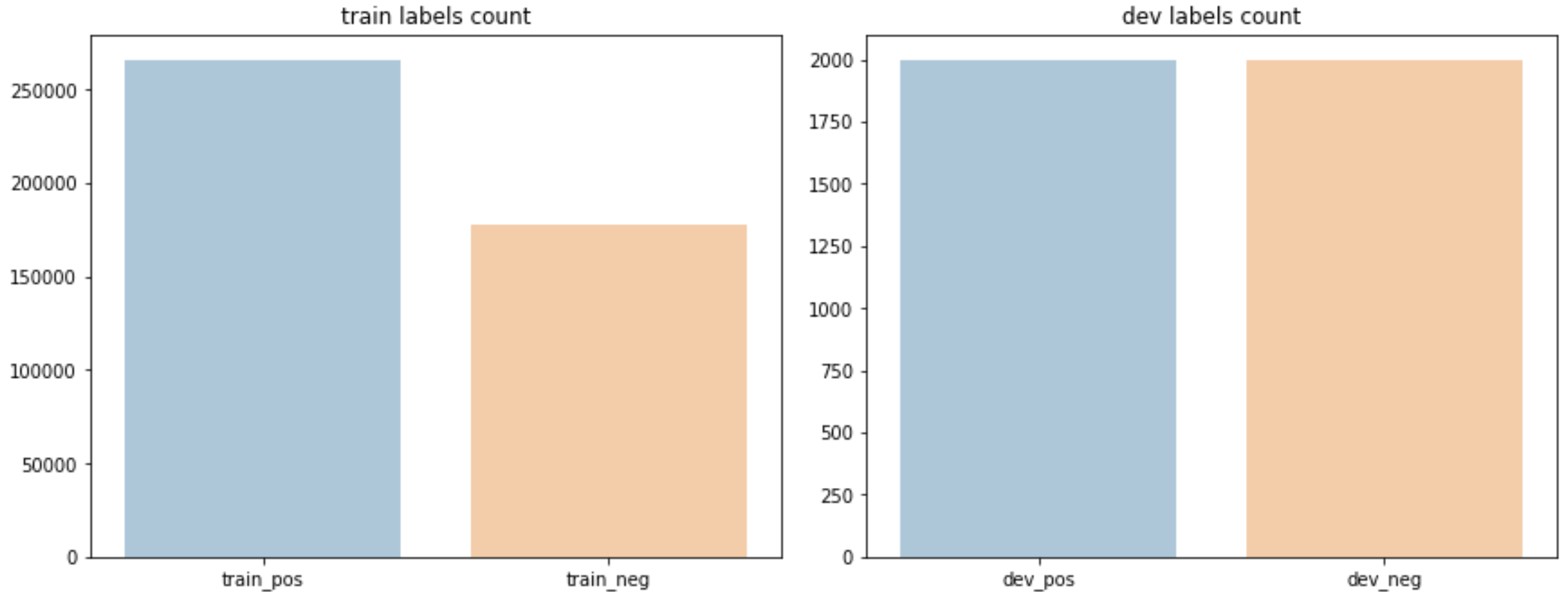
	review
0	these donuts have the perfect texture and taste .
1	good food for the price .
2	a little dirty on the inside , but wonderful p...
3	i always order it when i go there and it is al...
4	the rest of the food there is good also and no...

레이블: 파일명에 포함된 숫자

4. 프로젝트 결과 - Yelp 데이터 소개

Label EDA

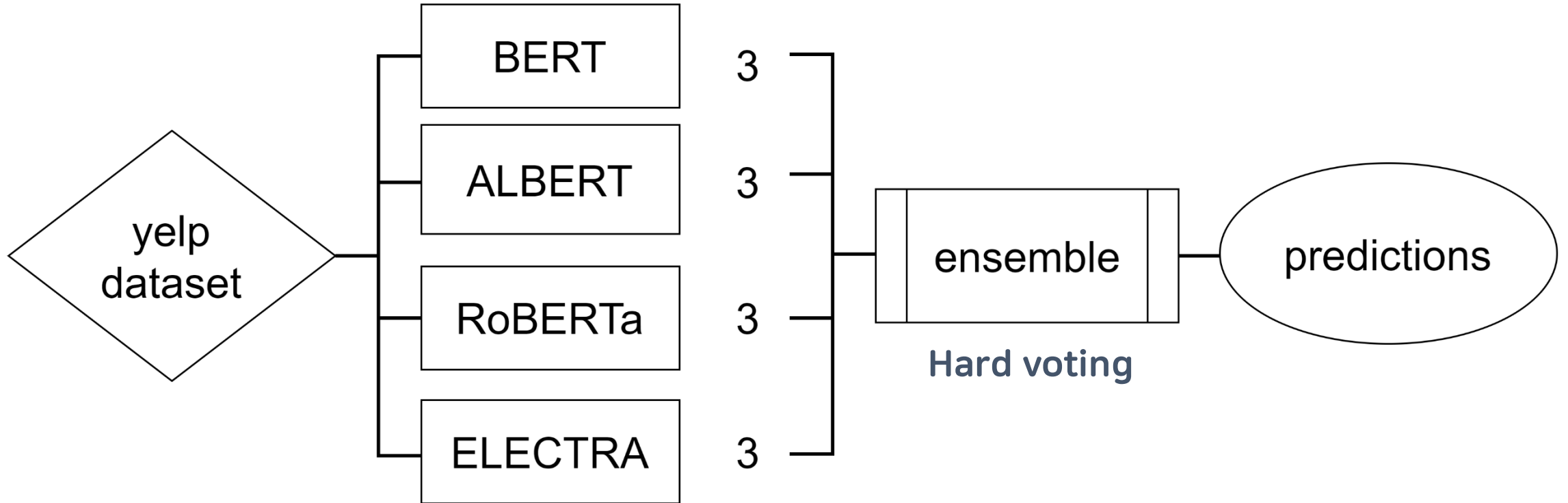
labels count



긍정 라벨과 부정 라벨 불균형 문제가 심각하지 않다고 판단 → 그대로 진행

4. 프로젝트 결과 - 전체 Process

모델별 hyperparameter 튜닝



Validation accuracy 상위 3개 모델 선정

4. 프로젝트 결과 – Pretrained LM 선정

Model	Published	Parameters	vs. BERT
BERT	2018	110M	-
RoBERTa	2019	356M	더 큰 batch와 data를 사용하여 training 하고 dynamic masking 사용
ALBERT	2020	12M	모델 크기를 줄여 memory limitation, training time, memory degradation 문제 개선
ELECTRA	2020	110M	모델이 작아도 BERT에 비해 빠르게 수렴하고 오히려 모델이 작을 수록 성능이 높아짐

선행 연구를 분석하여 Encoder 기반의 모델—BERT, RoBERTa, ALBERT, ELECTRA 선정
모든 pretrained LM은 base를 사용

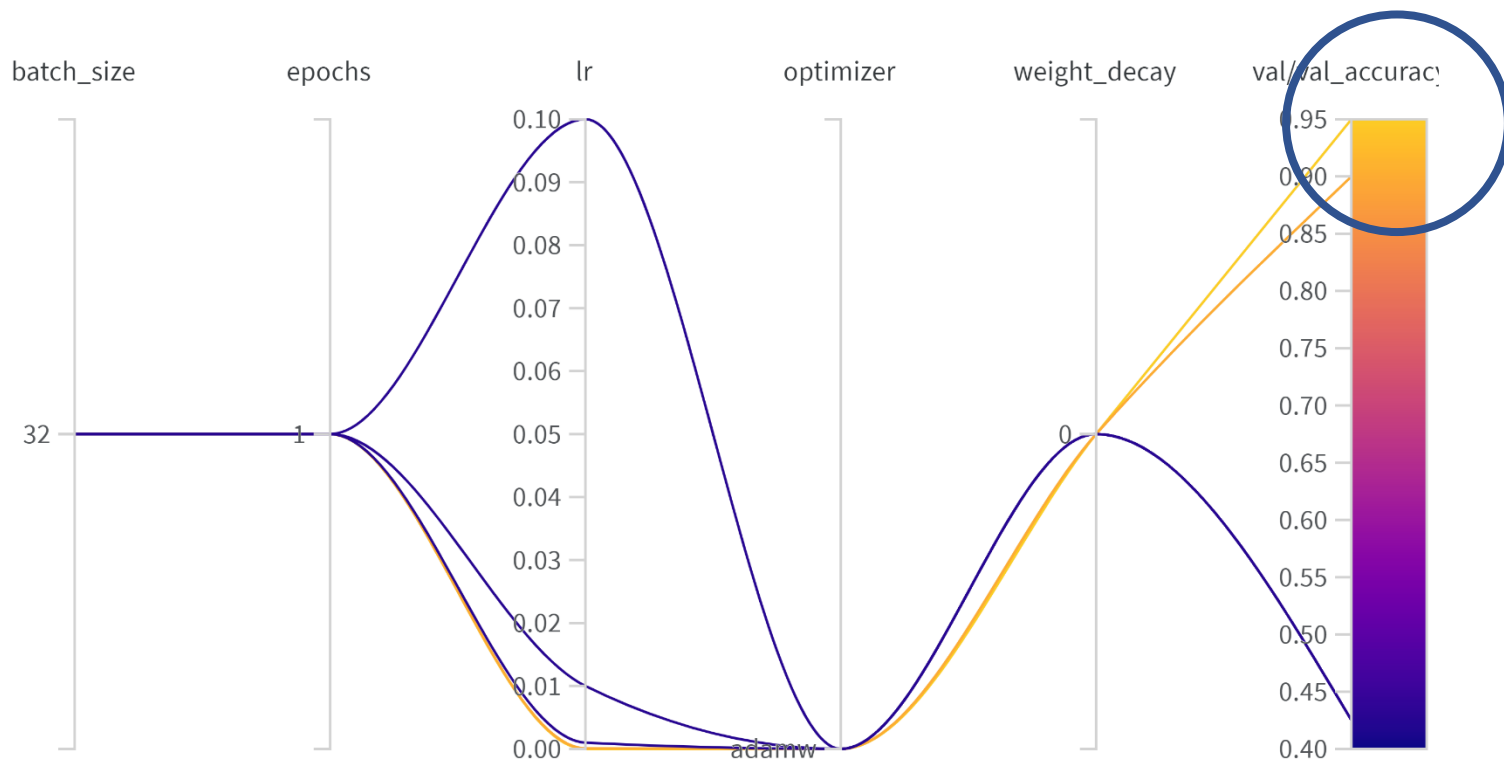
4. 프로젝트 결과 - Hard Voting

Model	Sample 1	Sample 2	Sample 3	Sample 4
BERT 1	1	1	1	0
BERT 2	1	0	0	1
⋮				
ELECTRA 3	1	0	0	1
Result	1	0	0	1

-> 가장 많이 나온 값을 최종으로 선정

4. 프로젝트 결과 - 1. hyperparameter tuning process

Step1. 데이터의 1%를 샘플링해 val_accuracy 기준 상위 2개의 learning rate 선정



Learning_rate 후보: [1e-05, 1e-04, 1e-03, 1e-02, 1e-01]

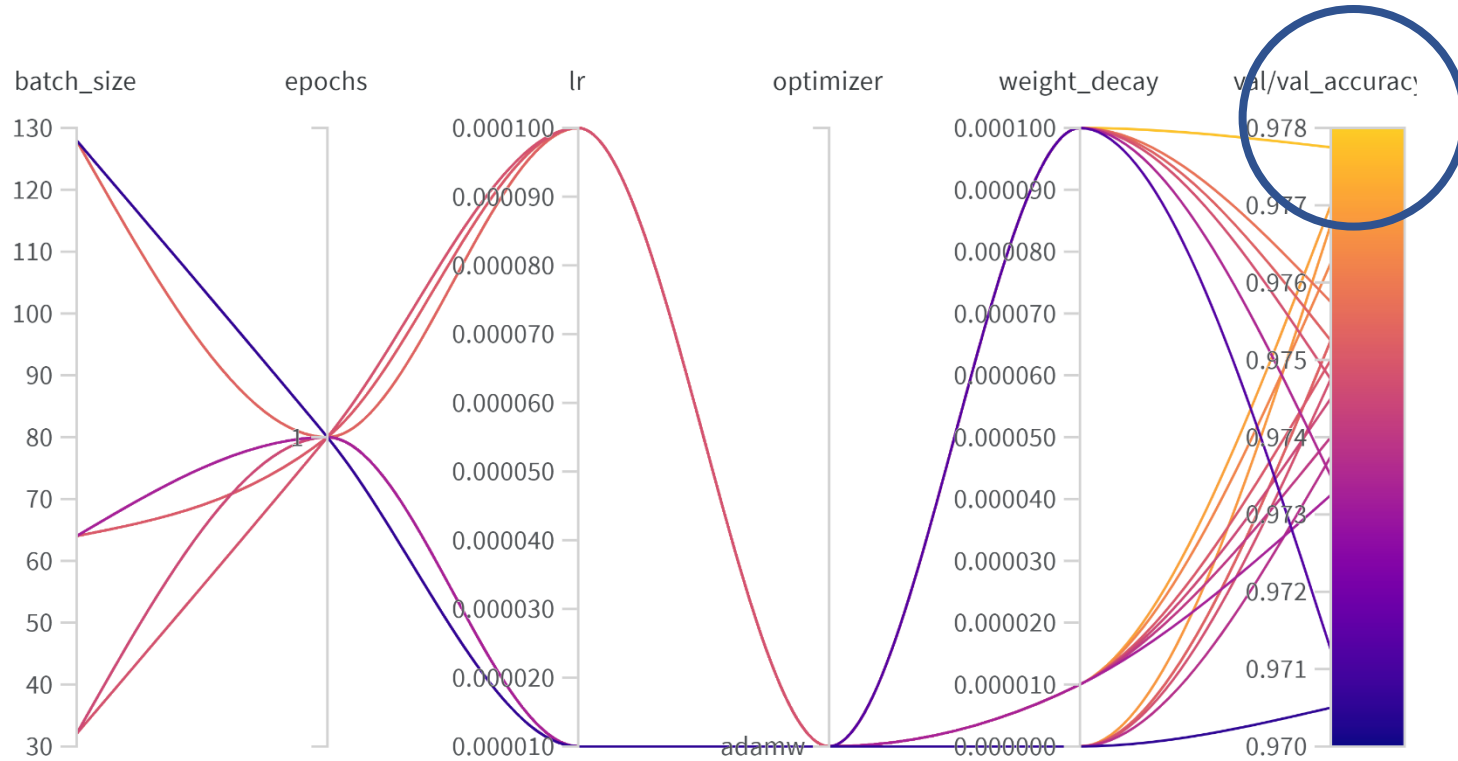
4. 프로젝트 결과 - 1. hyperparameter tuning process

Step1. 데이터의 1%를 샘플링해 val_accuracy 기준 상위 2개의 learning rate 선정

Pretrained Model	Learning Rate	Batch Size	Weight Decay	Epochs	Accuracy
Electra	1e-05	32	0	1	0.95
Electra	1e-04	ditto	ditto	ditto	0.9
Electra	1e-03	ditto	ditto	ditto	0.425
Electra	1e-02	ditto	ditto	ditto	0.425
Electra	1e-01	ditto	ditto	ditto	0.425

4. 프로젝트 결과 - 2. hyperparameter tuning process

Step2. 전체 데이터에서 step1에서 선정된 상위 2개 learning rate를 이용, epochs는 1로 고정하고 grid search를 진행해 최적 learning rate, batch_size, weight_decay를 구함



batch_size: [32, 64, 128]

epochs: [1]

weight_decay : [1e-04, 1e-05, 0]

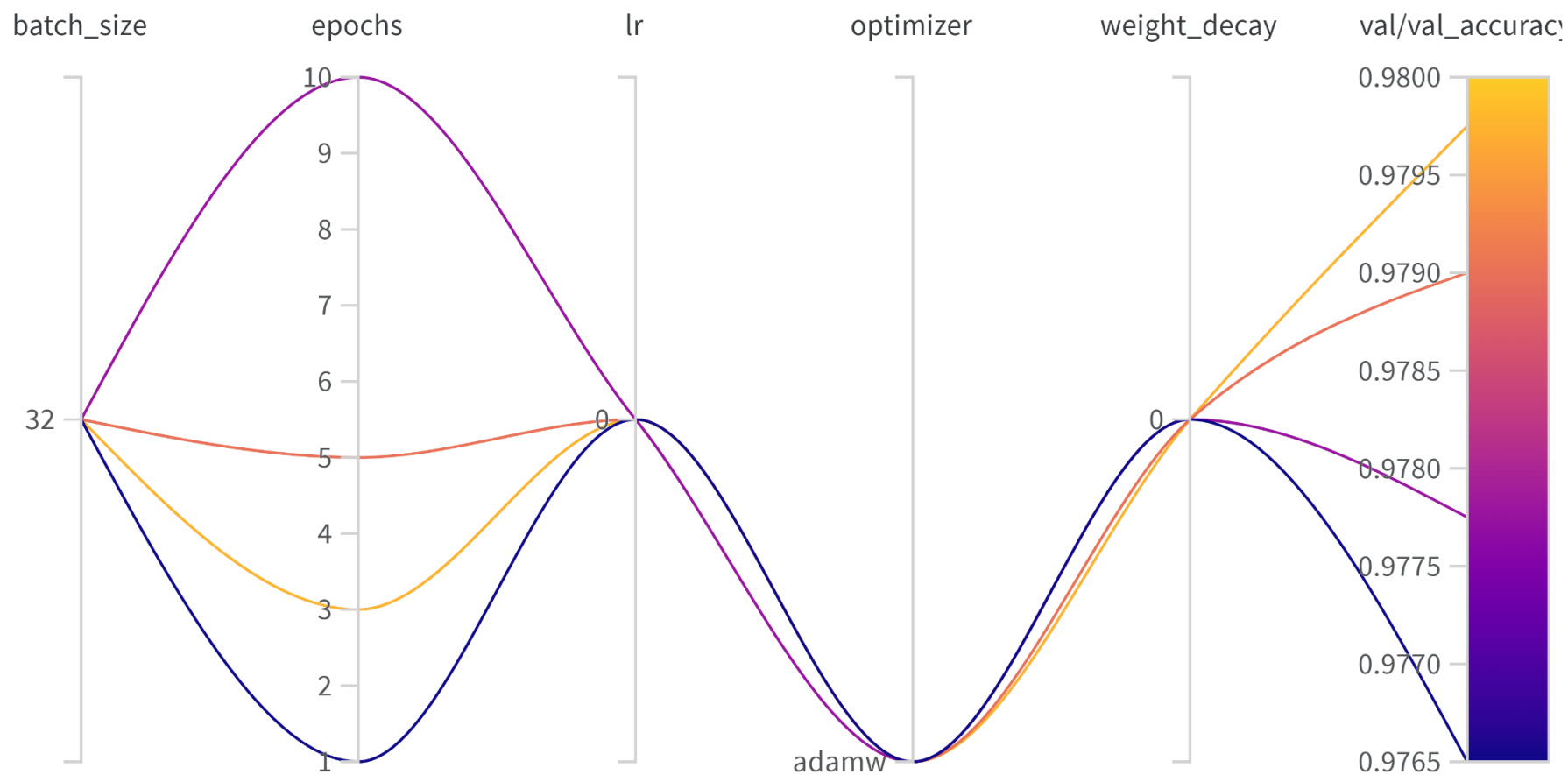
4. 프로젝트 결과 – 1. hyperparameter tuning process

Step2. 전체 데이터에서 step1에서 선정된 상위 2개 learning rate를 이용, epochs는 1로 고정하고 grid search를 진행해 최적 learning rate, batch_size, weight_decay를 구함

Pretrained Model	Learning Rate	Batch Size	Weight Decay	Epochs	Accuracy
Electra	1e-04	128	1e-04	1	0.9778
Electra	1e-04	32	1e-05	1	0.977
Electra	1e-04	64	0	1	0.9768
Electra	1e-04	128	1e-05	1	0.9762
Electra	1e-05	32	1e-04	1	0.9758

4. 프로젝트 결과 - 3. hyperparameter tuning process

Step3. 전체 데이터에서 위에서 선정된 hyperparameter에 epochs만 변경



4. 프로젝트 결과 – 1. hyperparameter tuning process

Step3. 전체 데이터에서 위에서 선정된 hyperparameter에 epochs만 변경

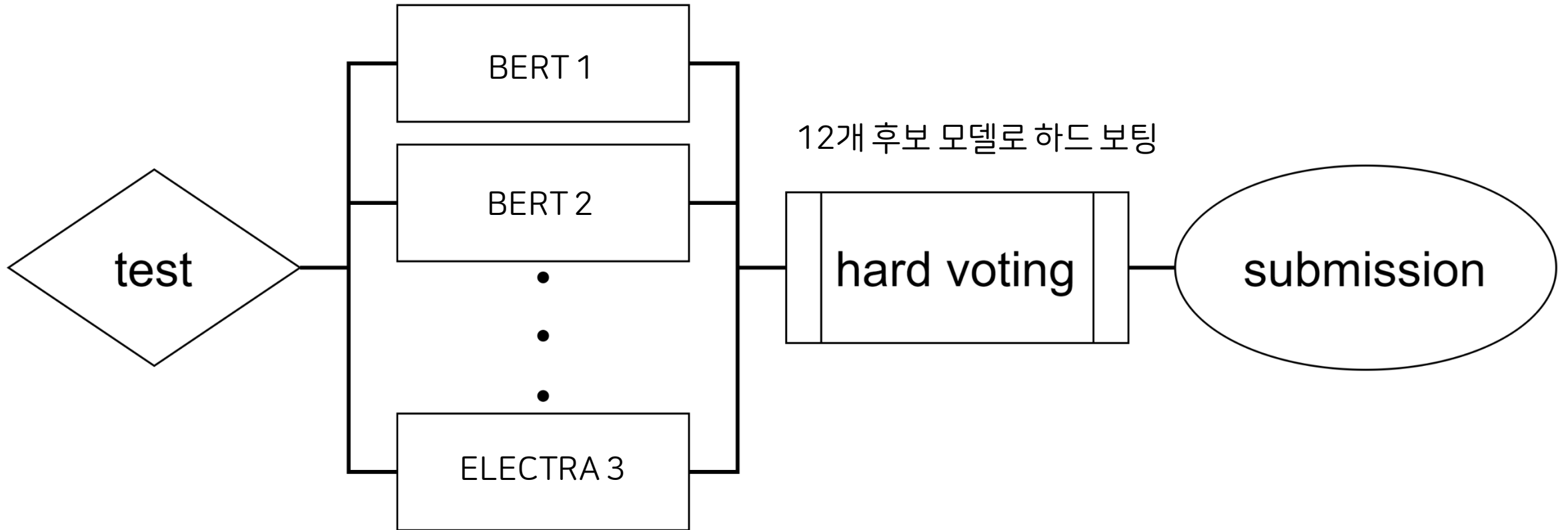
Pretrained Model	Learning Rate	Batch Size	Weight Decay	Epochs	Accuracy
Electra	1e-04	128	0	5	0.979
Electra	1e-04	128	0	1	0.9782
Electra	1e-04	128	0	3	0.9765
Electra	1e-04	128	0	10	0.9755

4. 프로젝트 결과 – hyperparameter tuning 결과

Pretrained Model	Learning Rate	Batch Size	Weight Decay	Epochs	Accuracy
BERT-base	1e-04	128	0	1	0.9805
	1e-05	32	0	1	0.979
	1e-05	64	0	1	0.9788
ALBERT-base	1e-05	64	1e-04	3	0.9825
	1e-05	64	0	3	0.9812
	1e-05	64	1e-04	10	0.981
RoBERTa-base	1e-05	32	1e-04	1	0.9782
	1e-05	32	0	3	0.9798
	1e-05	32	0	5	0.979
ELECTRA-base	1e-04	128	1e-04	1	0.9778
	1e-04	128	0	5	0.979
	1e-04	128	0	0	0.9782

ALBERT의 validation accuracy 가장 높게 나왔다

4. 프로젝트 결과 - 최종 스코어 결정



Kaggle 최종 정확도: 0.99

5. 자체 평가 및 보완

평가

1. Yelp dataset에 다양한 Pretrained LM을 적용해 각 모델의 성능 비교
2. Kaggle submission 99%의 정확도로 Yelp dataset 영어 리뷰 긍·부정 감성 분석

보완점

1. Hard voting 외의 다른 앙상블 방법 시도
2. grid search를 좀 더 세밀하게 진행
 - grid search 후보군을 다양하게 탐색
 - random seed, optimizer 등 다양한 hyperparameter 탐색
3. GPT2와 T5 모델을 시도했으나 결과가 예상처럼 나오지 않음
4. 테스트 하기 전에 훈련, 검증 데이터셋 합쳐서 모델을 훈련시키고 최종 아웃풋 산출

감사합니다!